Journal of Industrial and Systems Engineering
Vol. 9, No. 4, pp 9-25
Autumn (November) 2016

Intelligent approach for attracting churning customers in banking industry based on collaborative filtering

[1]Shafiei Gol, Elham, [1*]Ahmadi, abbas*, [2]Mohebi, Azadeh

[1]*Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran*

[2]*Iranian Research Institute for Information Science and Technology (IRANDOC), Tehran, Iran*

elham_shafieee@yahoo.com, abbas.ahmadi@aut.ac.ir, mohebi@irandoc.ac.ir

Abstract

During recent years, increased competition among banks has caused many developments in banking experiences and technology, while leading to even more churning customers due to their desire of having the best services. Therefore, it is an extremely significant issue for the banks to identify churning customers and attract them to the banking system again. In order to tackle this issue, this paper proposes a novel personalized collaborating filtering recommendation approach joint with the user clustering technology. In the proposed approach, first a hybrid algorithm based on Particle Swarm Optimization (PSO) and K-mean cluster the loyal customers. The clusters of loyal customers are used to identify the features of the churning customers. Finally, the list of appropriate banking services are recommended for the churning customers based on a collaborative filtering recommendation system. The recommendation system uses the information of loyal customers to offer appropriate services for the churning customers. We applied successfully the proposed intelligent approach to return the churning customers of an Iranian bank.

Keywords: Customer churn, data clustering, recommender system, collaborative filtering, particle swarm optimization.

1-Introduction

The subject of customer retention, loyalty, and churn is attracting a dramatic amount of attention in many industries particularly in the banking industry. Customer churn is one of the important issues for most banks. It costs considerably more to attract new customers than retaining existing ones, and, consequently the profit and revenue will increase by reducing customer churn rate. In fact, many experiential studies and models have proven that churn is one of the major challenges for banks and other consumer demanding companies (Lu et al., 2012). There exist many studies about the prediction of customer churn rate; however, they have not considered how we can retain the churning customers, how we can attract them to the company again, and finally, how we can prevent to lose our customers.

*Corresponding author.
ISSN: 1735-8272,

Our proposed approach for solving the above-mentioned problem is to design a recommender system in order to introduce suitable plans to the churning customers. A recommender system is one of the most beneficial and technological facilities for helping users to find most related and appropriate information for their requirements. These systems intend to propose items to users according to their interest and preferences (Wei et al., 2012).

The most famous type of recommender systems is based on collaborative filtering. Such systems gathering ratings or recommendations of items distinguish the commonalities among users based on their rating and a comparison between them and produce new recommendations. They try to suggest the items with similar tastes and favorites to user who liked them in the past. One common way to improve the results of collaborative filtering is to use unsupervised learning methods such as clustering, to reduce the scope of searching the nearest neighborhood (Shishehchi et al., 2011). In the collaborative filtering approach, the accuracy of the similarity measure among users plays a key role in the quality and performance of the recommender systems. Many studies have been done on using collaborative filtering along with the help of explicit feedback, but there is no profound study on implicit feedback. The lack of favorite information from users is the biggest challenge faced by implicit feedback recommenders particularly in the banking industry. Thus, having a method to recommend non-rated products is critical (Abdollahpouri & Abdollahpouri, 2013).

The proposed model in this paper can attract the customers who want to churn and try to return them to the bank. In other words, the proposed model endeavors to reduce customer churn and retain more customers in the bank. To achieve this goal, a new algorithm based on user category and interestingness for collaborative filtering is proposed, putting forward the novel calculation methodology of user similarity based on clustering via hybrid algorithm. Furthermore, in this paper, recommendations are implemented in the light of implicit feedback.

Specifically, in this paper we want to improve the performance of the collaborating filtering recommender system by means of hybrid algorithm, which is the combination of Particle Swarm Optimization (PSO) and K-means. PSO is a population-based search scheme tends to discover an optimal solution via a swarm of individuals known as particles. Due to the population-based nature of PSO, its procedure is less sensitive to the introductory conditions' effect and it leads to a near-optimal solution more likely. Besides, PSO can handle multiple objectives simultaneously. Accordingly, it is a suitable method for solving clustering difficulties where optimizing various objectives is of interest.

The paper is organized as follows. In the next section, the related studies in the area are described. Then, the methodologies used are expressed and the proposed model is presented. The experimental results and evaluation of the proposed system is described based on the real-data from Tourism bank (an Iranian Bank). Finally, the concluding notes and possible future extensions are described.

2- Literature review

Different clustering methods in collaborative filtering and various methods in churning customer's prediction are applied in recent studies. Ungar & Foster (1998) first introduced a formal statistical model of collaborative filtering and compared various algorithms for estimating the model parameters including variations of K-means clustering and Gibbs Sampling. Conner & Jon Herlocker (1999) presented an approach in collaborative filtering in which data clustering algorithms were applied to rating data. They used data partitioning and clustering algorithms to partition the set of products according to user rating data. Predictions are then calculated independently within each partition. Honda et al. (2001) proposed a new approach for the collaborative filtering using local principal components. Their proposed approach was based on principal component analysis and fuzzy clustering of incomplete data comprising missing values. Chee et al. (2001) developed an efficient collaborative filtering method called RecTree. This method first does an efficient *K*-means-like clustering to group the data and it creates neighborhood of similar users, and then performs subsequent clustering based on smaller, partitioned databases. Sarwar at al. (2002) addressed the performance issues by scaling up the neighborhood formation process via the use of clustering techniques.

Bridge and Kelleher (2002, 2003) generalized an existing clustering technique and applied it to a collaborative recommender's dataset. Later, they presented a collaborative recommender that uses a user-based model to predict user ratings for specified items. The model comprises summary rating information derived from a hierarchical clustering of the users. Mamunur et al. (2006) proposed ClustKnn, a simple and intuitive algorithm that first compresses data tremendously by building a straightforward but efficient clustering model. Recommendations are then generated quickly by using a simple nearest neighbor-based approach. They also demonstrated the feasibility of ClustKnn both analytically and empirically.

Huang et al. (2015) proposed a method using RFM and cross-correlation model. First, the value of customer is calculated via RFM. Then, by means of cross-correlation model, the usual losing curves of customer value are complemented. Eventually, by combining community detection and social network analysis (SNA), the possible losing customers are exposed.

Liao at al. (2015) introduced a hybrid churn prediction model for users of virtual world. The model is a combination of network influence approach and RFM model. It includes four parts: analysis of engagement, analysis of consumption, analysis of user behavior and analysis of social neighbor. The results show that the proposed model can detect the churners in the prediction time and effectively improves the churn prediction outcomes in the virtual worlds.

Bi et al. (2016) proposed a new clustering algorithm called Semantic Driven Subtractive Clustering Method (SDSCM) to prevent churning customers in big data era. Experimental results show that SDSCM is a better clustering method than Subtractive Clustering Method (SCM) and fuzzy c-means (FCM). Backiel at al. (2016) studied the combination of social network information into churn prediction models to increase precision, profitability and timeliness. They contributed a business-oriented approach, executed in Java and analyzed by means of SAS. It helps a company to use data produced in one month to generate a precise and useful predictions about churn in the next months and accurately plan incentives to interfere.

Fathian et al. (2016) presented data mining approaches for predicting churning customers. They introduced a combined model of clustering and ensemble classifiers. They used ensemble classifiers for comparisons. Particularly, self-organizing map (SOM) clustering technique, and four other classifiers, consisting of artificial neural networks, support vector machine, decision tree, and K nearest neighbors, were used. Furthermore, principal component analysis (PCA) method was used for decreasing the dimensions. The results show that performance of classification methods improves by combining two or more techniques .

All of the above studies related to churning customers have focused on finding the model to predict customer churn rate, but there is no model for retaining the churning customers and attracting them to the company again. Therefore, in this paper, we introduced the new collaborative filtering recommender system to attract churning customers. On the other hand, in all of these methods, an explicit feedback is used in the collaborative filtering approach, and there is no serious study on implicit feedbacks. Moreover, most of above-mentioned approaches have used clustering techniques such as *K*-means, *K*-harmonic means and fuzzy *c*-means. However, these techniques either are extremely dependent on the preliminary solutions or are more likely to converge to local optimal solutions. Furthermore, they do not provide promising results when coping with multiple objectives (Ahmadi et al., 2010). Therefore, in this paper we tend to increase accuracy of collaborating filtering recommender system by using the combination of PSO and *K*-means algorithms. PSO is a population-based search scheme, which tends to discover an optimal solution by using a swarm of individuals referred to as particles. It is less sensitive to the initial conditions because of its population-based nature. In addition, it does a global search of the solution space. Therefore, it is more probable to deliver a near-optimal solution (Ahmadi et al., 2010).

3- Clustering

In this paper, we focus on recommendation with implicit feedback, by using the collaborative filtering approach. The lack of favorite information from users is the biggest challenge faced by implicit feedback recommenders. One of the main problems with implicit data as compared to explicit data is the absence of feedback on the user’s favorites. Clustering techniques can help to find meaningful customer groups where each group has a specific behavior structure and the customers within such a group have similar product/service favorites and tastes. This behavior structure, i.e., the product/service favorites of a group of customers can be used to make recommendations to the customers (Renaud et al., 2013).

In our proposed approach, we used the hybrid of *K*-means and PSO algorithms. Although PSO is a fast clustering algorithm, when the dataset is large or complex it does not carry out well. PSO is proficient in global search but is poor in local search. While *K*-means is a good choice for local search, it didn't perform well in term of global search (Shen et al., 2010). This hybrid technique uses the benefits of both algorithms by sequentially applying PSO and *K*-means to the search area (Vora and Oza, 2013). At the initial stage, the PSO clustering algorithm is implemented to search globally for the location of clusters' centroid. These locations are used as initial centroids for *K*-means clustering algorithm for purifying and producing the optimal clustering solutions. This procedure not only locally avoids the limitations of these algorithms but increases the advantages of both algorithms as well.

3-1-K-means clustering

One of the most significant components of a clustering algorithm is the measure of similarity used to specify how close two patterns are to each another. *K*-means clustering groups' data vectors into a predefined number of clusters (*K*) based on Euclidean distance as similarity measure. For the purpose of this approach, we define the following notations:

- $\boldsymbol{x}_i$ indicates the data vector *i*.
- $\boldsymbol{m}_k$ shows the center of cluster *k*.
- n_k shows the number of data in cluster *k*.
- C_k shows the subset of data that form cluster *k*.

By using the above notations, the standard *K*-means procedure is given in Algorithm 1.

Algorithm 1 *K*-means clustering

(1) Randomly, initialize K cluster centers.

(2) Repeat

(a) For each data vector, assign the vector to the class with the closest centroid vector, where the distance to the centroid is defined using

$$\mathrm{d}(\boldsymbol{x}_i, \boldsymbol{m}_k) = \sqrt{\sum_{j=1}^{d}(x_{ij} - m_{kj})^2}, \tag{1}$$

where *d* indicates the dimension.

(b) Recalculate the cluster centroid vectors, using

$$m_k = \frac{1}{n_k}\sum_{\forall \boldsymbol{x}_i \in C_k} \boldsymbol{x}_i \ , \tag{2}$$

Until a stopping criterion is satisfied.

In this paper, the algorithm stops when a user-specified number of iterations has been exceeded or there is no change in clusters.

3-2-Particle swarm optimization (PSO) based clustering

PSO algorithm is based on bird flocking behavior and it is one of the population-based optimization algorithms, originally introduced by Kennedy and Eberhart (1995). The main idea is to initialize a group of particles randomly. Each particle is a candidate solution of the optimization problem. The performance of each particle is measured using a fitness function, f, which is defined properly according to the problem at hand. Each particle h moves in the search area, updating its velocity, $\boldsymbol{v}_h$, and position, $\boldsymbol{y}_h$, according to the formulas of velocity and position (Yuyan,et al., 2013; Xu, 2013). Each particle has a position in d-dimensional space and is "flown" among this multi-dimensional search space, changing its position toward both the particle's best position found so far, $\boldsymbol{y}_h^{bp}(t)$, and the best position in the neighborhood of that particle, $\boldsymbol{y}^*(t)$. Updating rules for position and velocity of particle h at iteration t+1 are given below:

$$\boldsymbol{v}_h(t+1) = w\boldsymbol{v}_h(t) + c_1 r_1(t)(\boldsymbol{y}_h^{bp}(t) - \boldsymbol{y}_h(\mathrm{t})) + c_2 r_2(t)(\boldsymbol{y}^*(t) - \boldsymbol{y}_h(\mathrm{t})), \tag{3}$$

$$\boldsymbol{y}_h(t+1) = \boldsymbol{y}_h(t) + \boldsymbol{v}_h(t+1)\ . \tag{4}$$

where w is the inertia weight which usually linearly decreases during the iteration; $c_1$and c_2 are regulatory factors which control global and local search and in the rang [0,2]; $r_1(t), (t)$ are two random numbers produced by uniform distribution in the range [0,1].

The personal best position of particle h is calculated as following according to the minimum objective function criterion:

$$\boldsymbol{y}_h^{bp}(t+1) = \begin{cases} \boldsymbol{y}_h^{bp}(t) & if\ \ f(\boldsymbol{y}_h(t+1)) \geq f((\boldsymbol{y}_h^{bp}(t)) \\ \boldsymbol{y}_h(t+1) & if\ f\big(\boldsymbol{y}_h(t+1)\big) < f((\boldsymbol{y}_h^{bp}(t)) \end{cases} \tag{5}$$

PSO algorithm is usually executed by repeated application of equations (3) and (4) until a specified number of iterations has been exceeded. Alternatively, the algorithm terminates when the velocity updates are close to zero over a number of iterations. The general procedure of the basic PSO procedure is given in Algorithm 2 (Ahmadi et al., 2010).

Algorithm 2 Particle swarm optimization (PSO)

1. Initialize the velocity and position of each particle randomly.
2. Update the velocity and position of particle according to the Eq. (3) & Eq. (4).
3. Calculate the fitness value of each particle according to a fitness function.
4. Compare the fitness value of each particle with the previous individual best fitness value of this particle. Personal best position of particle is modified if the position improves the solution.
5. Global best fitness value is updated if needed.
6. Repeat steps 2 to 5 until the termination condition is met (usually fixed number of iteration or no change in solution).

In order to use PSO procedure for clustering, the position of particle h is defined as $\boldsymbol{y}_h = (\boldsymbol{m}_h, \dots, \boldsymbol{m}_K)_h$. In other words, it has a candidate for the center of each cluster k. The same procedure as algorithm 2 can be used for PSO clustering. Moreover, the objective function is the same as that of K-means algorithm. That is, the cumulative distance of each data from their associated cluster center is minimized (Ahmadi et al., 2010).

3-3-Hybrid algorithm: combination of PSO and *K*-means

This hybrid technique comprises two clustering algorithms; first one is PSO and second one is K-means. The procedure is given in Algorithm 3.

Algorithm 3 Hybrid algorithm

1. PSO clustering
   a. Initialize the swarm randomly.
   b. Update the position and velocity of particles using Eq. (3) and Eq. (4).
   c. Update personal best of each particle, $\boldsymbol{y}_h^{bp}$, and global best of swarm, $\boldsymbol{y}^*$, if needed.
   d. Repeat steps (a) to (c) until the maximum number of iterations is exceeded.
2. *K*-means clustering
   a. Initialize cluster centroids using the global best position, $\boldsymbol{y}^*$.
   b. Assign each data to the closest cluster center.
   c. Recalculate the cluster centers using Eq. (2).
   d. Repeat steps (a) to (c) until the centers no longer change.

4 - Proposed approach to attract churning customers

The main idea for attracting churning customers is to learn from loyal customers' behavior. First, the underlying hidden groups in loyal customers are extracted using the hybrid of PSO and *K*-means clustering described in Algorithm 3. Then, the most similar loyal cluster for any given churning customer is determined. On one hand, the behavior of loyal customers are known. On the other hand, the similarity between churning customers and loyal clusters are obtained. Hence, a churning customer may reveal the same behavior as the similar loyal cluster's customers do. The proposed approach is organized in four phases and its schematic conceptual presentation is shown in Figure 1.

In the following, four phases of the proposed approach are explained in detail.

4-1- Data preprocessing

In this phase, personal and account information of customers, the list of customers who want to churn along with the loyal customers are gathered. First, two data bases including customers' transaction information and customers' personal information are merged together. Then, nominal variables are converted to numerical variables. Finally, all of the data are scaled to real numbers in the interval [0, 1].

4-2- Clustering loyal customers

The hybrid of *K*-means and PSO algorithm is employed to cluster the loyal customers. Then, the Euclidean distance using equation (1) is considered to find the similarity between churning customer and loyal clusters. At the end of this phase, the most similar loyal cluster for each of churning customers is determined.

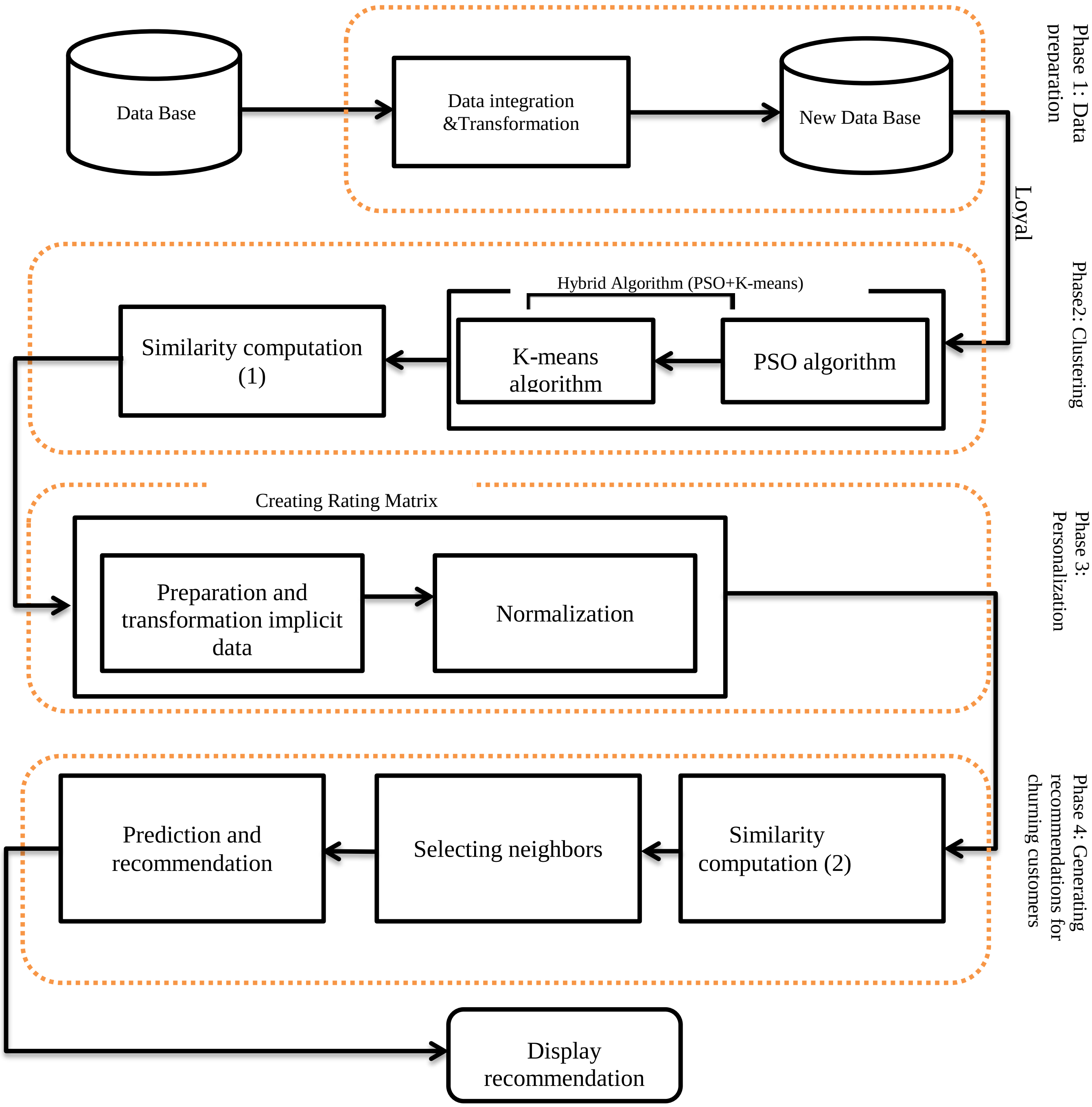

Figure 1. Schematic illustration of proposed approach

4-3- Data personalization

Now, we need customer's behavior history in the past service selection to appropriate him services. For instance, we need to know which services he has used most recently or we need to know which services he has used more than others have. Any customer displays his interest in particular service via some implicit behavior such as clicking on it, watching it or using it and does not give an explicit opinion about service. In banking industry, the implicit ratings are more useful since the explicit rating on services is not known. However, the number of times a customer uses a specific service, for instance, or fund transfer are known. One can use this information to personalize the recommendation.

We specify the ratings of a customer as a vector indicated by$\boldsymbol{R_x}$. For normalizing the values, the following equation is used:

$$\boldsymbol{R}_{normalized(x)} = \frac{\boldsymbol{R_x} - \bar{\boldsymbol{x}}}{s} \ , \tag{6}$$

where $\bar{\boldsymbol{x}}$ is the sample mean of ratings and s is the standard deviation of ratings calculated as follows.

$$s = \sqrt{1/d \sum_{i=1}^{d} (\boldsymbol{x}_i - \bar{\boldsymbol{x}})^2} \ , \tag{7}$$

where *d* is the number of elements in the vector, .i.e., the number of ratings a customer has given to different services (Abdollahpouri & Abdollahpouri, 2013).

4-4- Generating recommendations for churning customers

We use memory and user based collaborative filtering in the proposed model. We also use a database of preferences for services of bank by users to predict additional services a churn customer might like. Let's assume a list of *m* users $\{u_1, u_2, u_3, \dots u_m\}$ and a list of *n* services $\{i_1, i_2, i_3, \dots i_n\}$ are given and each user has a list of services. Moreover, $r_{u,i}$ is the service that user has rated, or about which his/her preferences have been inferred through his/her behavior (Sullivan, 2010, Miller et al., 2004, Mansour et al., 2005).

The ratings in our proposed approach are implicit indications, such as transaction of user or number of service usage, as described in phase 3. We use the user rating data to calculate the similarity, or weight, between users and make predictions or recommendations according to those calculated similarity values (Sullivan, 2010, Miller et al., 2004, Mansour et al., 2005).

Similarity computation between items or users is a very important step in memory-based collaborative filtering (CF) algorithms. For a user-based CF algorithm, we first compute the similarity, $w_{u,v}$, between the users *u* and *v* who have both rated the same items. Pearson correlation based similarity computation is considered. Pearson correlation measures the extent to which two variables linearly relate to each other [20], as given by:

$$w_{u,v} = \frac{\sum_{i \in I} (r_{u,i} - \overline{r_u})(r_{u,j} - \overline{r_v})}{\sqrt{\sum_{i \in I} (r_{u,i} - \overline{r_u})^2} \sqrt{\sum_{i \in I} (r_{u,j} - \overline{r_v})^2}} \ , \tag{8}$$

Where $i \in I$, and I denotes the items that both the users' *u* and *v* have rated and $\overline{r_u}$ is the average ratings of the co-rated items of the *u*th user.

Selection of the neighbors who will serve as recommenders is done next. Two techniques have been employed in the collaborative filtering recommender systems: First, threshold-based selection, according to which users whose similarity exceeds a certain threshold value are considered as neighbors of the target user. Second, the top-*n* technique, in which, *n*-best neighbors are selected and the *n* is given in advance (Songjie, 2010). We use both of techniques in our proposed approach.

To get *predictions or recommendations* is the most critical step in a collaborative filtering. In the neighborhood-based CF algorithm, a subset of nearest neighbors of the churning customer is chosen based on their similarity, and a weighted aggregation of their ratings is used to produce predictions for the churning customer (Sullivan, 2010, Abdollahpouri & Abdollahpouri, 2013). The most used technique to compute prediction is weighted sum of rating's average. In this technique to make a prediction for active user, *a*, on a certain item, *i*, we can take a weighted average of all the ratings on that item according to the following formula (Sullivan, 2010, Abdollahpouri & Abdollahpouri, 2013):

$$P_{a,i} = \overline{r_a} + \frac{\sum_{u \in U} (r_{u,i} - \overline{r_u}) . w_{a,u}}{\sum_{u \in U} |w_{a,u}|} \ , \tag{9}$$

Where $\overline{r_a}$ and $\overline{r_u}$ are the average ratings for users a and u on all services, and $w_{a,u}$ is the weight between the user a and user u. The summations are over all the users' u∈U who have rated item *i*.

5- Implementation and Results

In this section, we describe the used database and then, the performance of the hybrid (PSO& k-means) clustering algorithm is evaluated and is compared with other partitioned approaches such as K-means and PSO. Next, the proposed collaborative filtering recommender system is evaluated.

5-1- Data base

A real-world data set of a real Bank is used in this paper. The data is extracted from customer's account data, personal information of customers and the banking services, which customers used within a six-month period. The data set contains observations on 23 variables for 1000 customers. Among the variables, 12 variables are categorical, 10 variables are numerical and one variable is date. Moreover, all variables are independent (input variables). The description of database is given in Table 1.

Table 1. The summary of customers' dataset

Data Set Characteristics:	Multivariate	Number of Instance:	1000	Area:	Financial
Attribute Characteristics:	Categorical, Integer	Number of Attributes:	23	Associated Tasks:	Clustering
Number of Numerical Attribute:	11	Number of Qualitative Attribute:	12	Number of Output Variable:	0

5-2- Performance analysis measures

Evaluation of clustering results sometimes is mentioned as cluster validation. There have been several suggestions for a measure of similarity between two clusters. In this paper, we will use compactness measure, separation measure, Davies–Bouldin index (DBI), Dunn index and Silhouette index (Rui & Wunsch, 2005).

The compactness measure identifies that how much similar the samples of a cluster are to each other and are dissimilar from those in other clusters. A suitable example for this measure is within-cluster distance and the goal is to minimize this measure as much as possible. The separation measure is between-cluster distance, which is the cumulative distance between cluster centers. This clustering technique tends to maximize this criterion. The Davies-Bouldin favors clustering with low intra-cluster and high inter-cluster distances. The lower its value, the better it is. Dunn's index also prefers clustering with low intra-cluster and high inter-cluster distances, although the compactness of the clusters is assessed in a different way. The Dunn's index should be maximized. In recommender systems, people have utilized several kinds of measures for evaluating the performance. In this paper, we used *mean absolute error* (MAE) among ratings and predictions, which is a broadly used metric. MAE used as a measure of the deviation of recommendations from their real user-specified values. For each ratings-prediction pair $< p_i, q_i >$ this metric gives the absolute error among them, i.e., $|p_i - q_i|$ equally. The MAE is calculated by first summing these absolute errors of the *d* corresponding ratings-prediction pairs and then calculating the average, as given below:

$$\text{MAE} = \sum_{i=1}^{d} \frac{|p_i - q_i|}{d} \tag{10}$$

The lower the MAE, the more accurate the recommendations will be.

5-3- Evaluation of hybrid algorithm (PSO + K-means)

In this section the performance of the different clustering techniques including PSO, K-means and hybrid algorithm, is presented. Assume that the number of particles in PSO Algorithm is 30, inertia weight = 0.729 and $c_1 = c_2 = 1.49445$ according to (Ahmadi et al., 2010). Compactness measure is used as the objective function. The results are presented in Figure 2, which are the values obtained by running the associated clustering algorithm for 50 iterations independently.

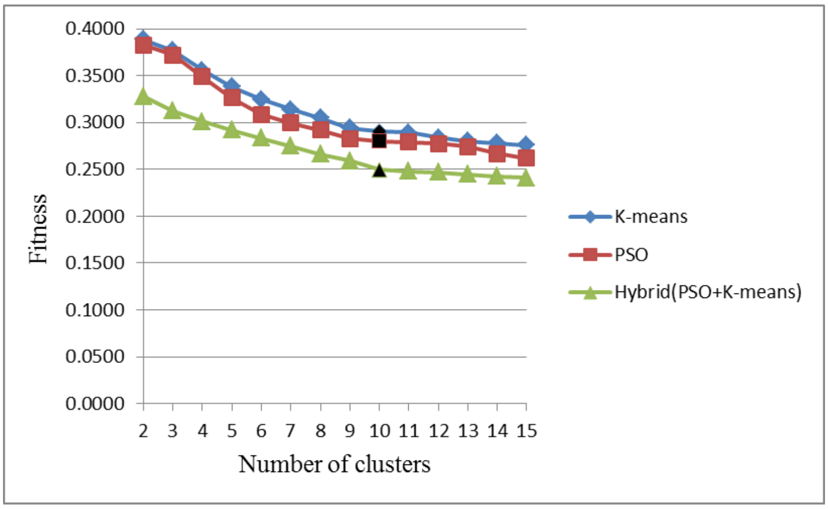

Figure 2. Investigating optimal number of clusters in terms of compactness measure: The bank data

As seen in Figure 2, the hybrid clustering algorithm has shown better result as compared to K-means and PSO algorithms. Moreover, the performance of these three clustering algorithms is evaluated in terms of three cluster validity measures, namely separation, Davies-Bouldin index and Dunn's index measures for the bank data set, as illustrated in Figure 3. The presented results in this table indicate the value obtained by running the associated clustering algorithm for 50 times independently.

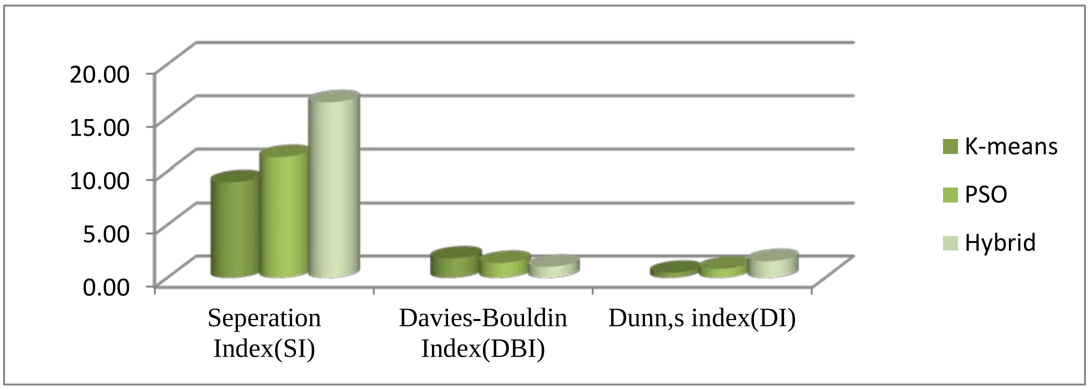

Figure 3. Average comparison of different measurements using the bank data

As can be seen in Figure 3, the hybrid algorithm provides better results than the others in all indexes do. Moreover, the convergence of *K*-means, single swarm and hybrid clustering algorithms using compactness measure for the bank data sets are shown in Figure 4.

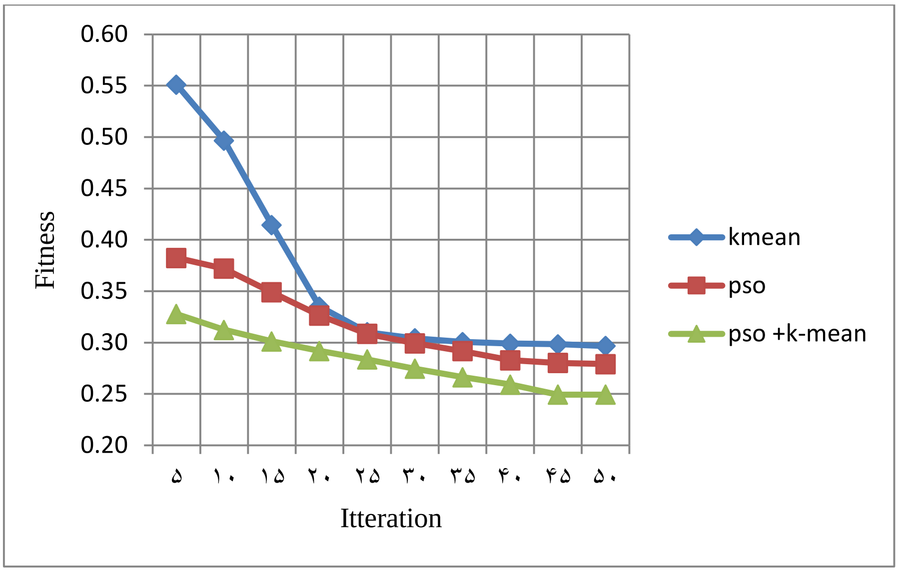

Figure 4. Comparing the performance of clustering methods in terms of Compactness measure: Bank data

The results presented in Figure 4 indicate that hybrid clustering algorithm produces better solutions in terms of the compactness measure for different iterations.

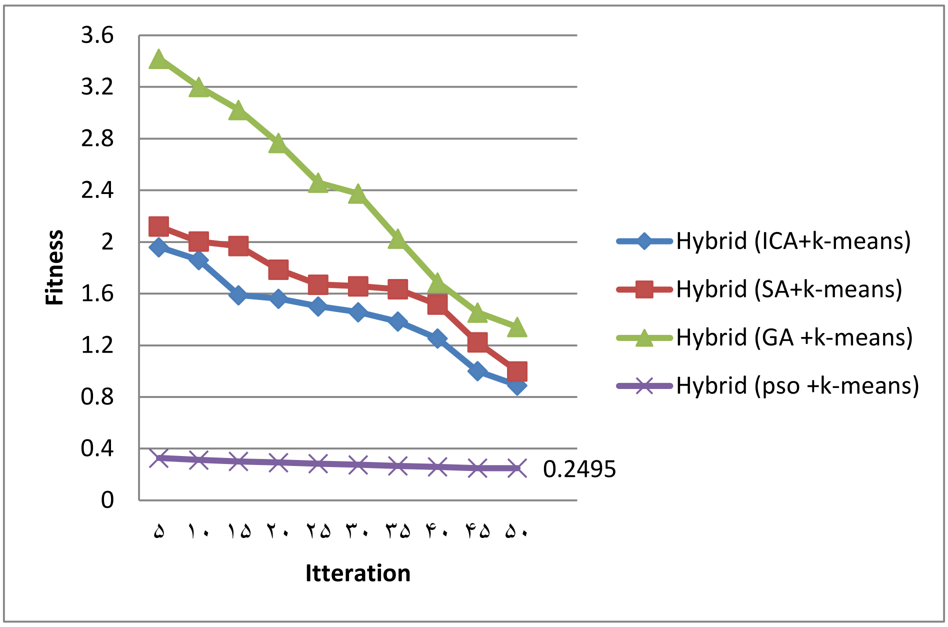

Figure 5. Comparing the performance of the different hybrid algorithms in terms of compactness measure: Bank data set

Moreover, the results displayed in Figure 5 indicate that hybrid (PSO + K-means) generates better solutions in terms of the compactness measure for different iterations.

Table 2. Results obtained by the algorithms for 50 different runs on the Bank data set

Method	Function value			Standard deviation
	F_{best}	$F_{average}$	F_{worst}	
K-means + PSO	0.2495	0.2816	0.3278	0.0015
K-means + ICA	0.8901	1.4566	1.95989	0.287
K-means + SA	0.9981	1.6578	2.12	0.3244
K-means + GA	1.3419	2.3755	3.4193	1.0789

Table 2 presents a comparison among the results of different hybrid approaches including (K-means + PSO), (K-means + ICA), (K-means + SA) and (K-means + GA) for 50 different runs on the Bank data set.

The results given in Table 2 demonstrate that hybrid of K-means and PSO is very precise and reliable. In other words, it provides the optimum value and small standard deviation in comparison to those of other methods. Besides, the standard deviation of the fitness function for this algorithm is significantly less than that of other methods.

Table 3. Values of parameters of each of five algorithms

(K-means + PSO)		(K-means + ICA)		(K-means + SA)		(K-means + GA)	
Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
# particles	30	N_{pop}	100	Probability threshold	0.98	Population	50
C_1	1.49445	N_{imp}	6	Initial temperature	5	Crossover	0.8
C_2	1.49445	β	5	Temperature multiplier	0.98	Mutation rate	0.001
Weight	0.729	ξ	0.05	Final temperature	0.01	# iterations	500
# iterations	50	γ	0.7	# Iterations detect steady state	50		
		# iterations	50	# iterations	1000		

5-4- Evaluation of the collaborative filtering recommender system

In this section, experimental results of applying the proposed user-based collaborative filtering based on hybrid algorithm are presented to produce recommendations for churning customers and attract them to the bank again. The results are mainly divided into two parts, quality results and performance results. In order to evaluate the quality of recommendations, first the sensitivity of some parameters is considered, before running the main experiment. These parameters include the neighborhood size and effects of various similarity measures. For specifying the sensitivity of different parameters, customers are divided in two groups, train and test. After evaluating the accuracy of proposed system, it will be evaluated with churning customers, to investigate how many of them will be attracted to the bank again, by calling and suggesting them the services that the system recommends.

5-4-1-Effect of similarity measures

Three different similarity algorithms are applied which are basic cosine, adjusted cosine and Pearson correlation. Then, they are implemented on testing group data, which are three groups with 100 memberships. For each similarity algorithms, its effect on the accuracy of proposed recommender system is measured through computing the MSE. Figure 6 shows the obtained results.

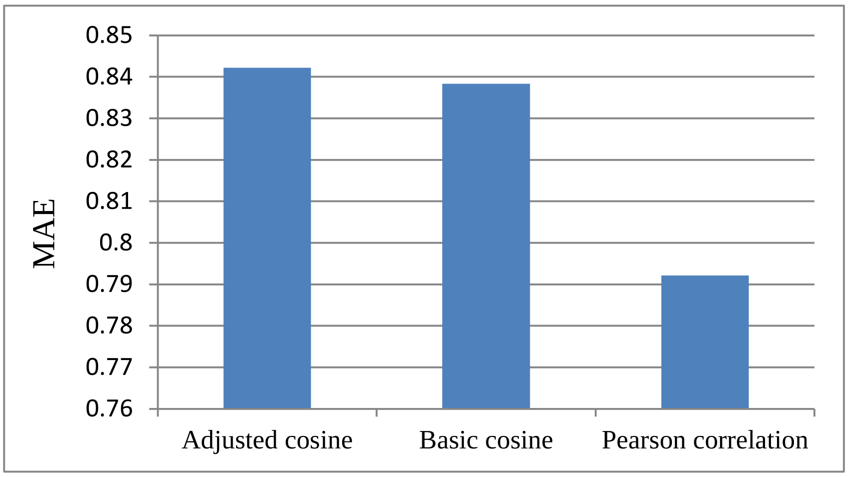

Figure 6. Relative performance of different similarity measures

It can be observed from the results that the Pearson correlation similarity computation has a clear advantage, as its associated MAE is meaningfully low. Hence, the Pearson correlation similarity is chosen for our proposed approach.

5-4-2-Effect of different thresholds for selecting neighbors

Different thresholds is considered for selecting the number of neighbors who are similar to the target customer and weighted sum algorithm is used to generate the prediction. The test set is used in order to compute MAE. Figure 7 illustrates the effect of various threshold values in terms of MAE.

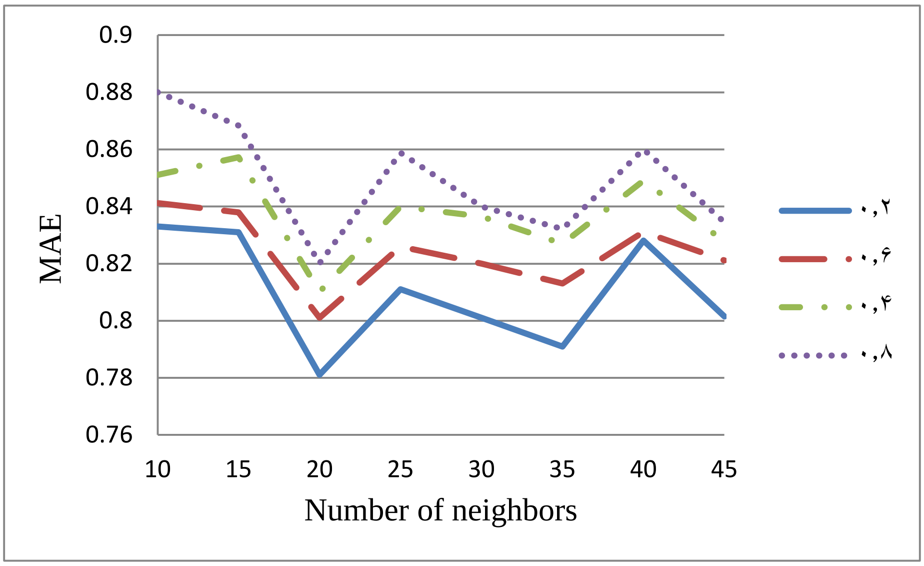

Figure 7. The influence of different thresholds for selecting neighbors in terms of MAE

Figure 7 shows that when the threshold is 0.2 and number of neighbors is 20 the MAE is the lowest. Thus, we choose 0.2 as the threshold for the proposed recommender system.

5-4-3- Comparison of the proposed approach with the others

Here, the proposed approach is compared with other existing approaches including: traditional collaborative filtering, collaborative filtering based on *K*-means and collaborative filtering based on PSO. Four groups of customers that each of them has 100 members are considered as test data. Then, the different approaches are run and for each of them the MAE is computed. The results are shown in Figure 8.

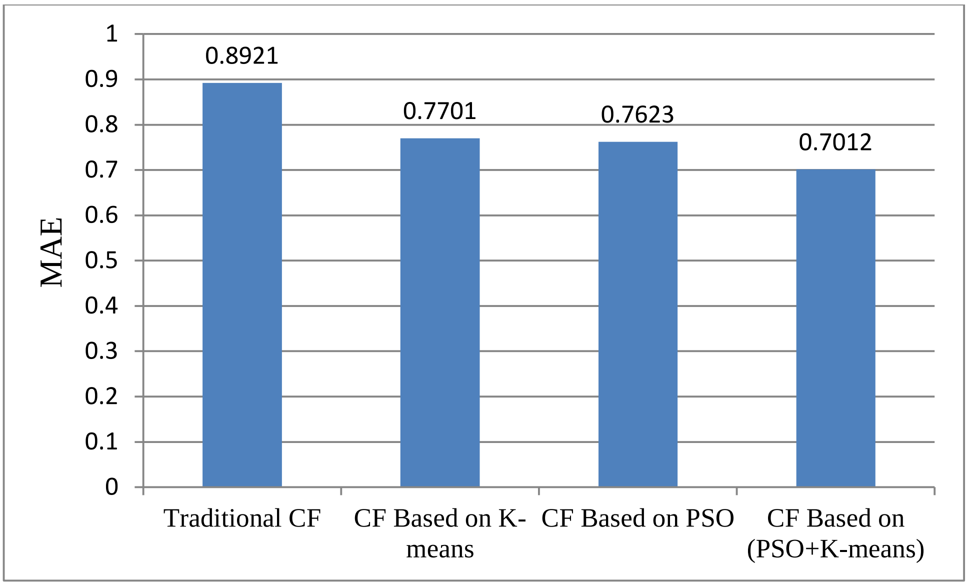

Figure 8. Comparison of the proposed approach and the other ones

As shown in Figure 8, the MAE for the proposed approach, which is collaborating filtering, based on hybrid algorithm is the lowest one. Therefore, it has the best result and higher accuracy than others do.

6- Conclusion

Customer churn is a serious problem in almost every company especially at the banks. Therefore, it is very important issue for every industry to find churning customer and attract them to company again. In this paper, the novel model is introduced for attracting customers who want to churn in the banking industry. It was the personalized collaborative filtering recommender system based on hybrid clustering algorithm which was the combination borrowed from PSO and *K*-means algorithms. The proposed model included four phases. In the first phase, related personal information, account information, the list of customers who want to churn and customers who are loyal were gathered and then, they were preprocessed. Next, in phase two, the clustering was done on loyal customers by means of hybrid algorithm to distinguish which cluster of loyal customers were more similar to our churning customer based on customers account data and their personal information. Then in phase three, the personalization is done. We worked on historical implicit data to deduce the history of customer's treatment and activities in the past service selection to be able to propose him/her some services that will be interesting to him/her. At last, the predictions or recommendations are done.

In terms of future research, we can get the advantage of all kinds of technologies, such as multiple PSO clustering, multi-classifier approach and Genetic algorithm to improve the proposed approach.

References

Lu, N., Lin, H., Lu,, & Zhang,G. (2012). A Customer Churn Prediction Model in Telecom Industry Using Boosting. *IEEE Transactions on Industrial Informatics*, 10(2), 1659-1665.

Wei, S., Ye, N., Zhang, S., Huang, X., & Zhu, J. (2012). Item-based collaborative filtering recommendation algorithm combining item category with interestingness measure. *Proceeding of International Conference on Computer Science & Service System (CSSS)*, pp. 2038-2041.

Shishehchi, S., Banihashem, S.Y., Zin, N.A.M., & Noah, S.M. (2011). Review of personalized recommendation techniques for learners in e-learning systems. *Proceeding of International Conference on Semantic Technology and Information Retrieval (STAIR*), pp. 277-281.

Ungar, L., & Foster, D.P. (1998). Clustering Methods for Collaborative Filtering. *Processing Workshop on Recommendation Systems at the 15th National Conference on Artificial Intelligence*, vol. 1.

Ungar, L., & Foster, D.P. (1998). A Formal Statistical Approach to Collaborative Filtering. *Proceedings of Conference on Automated Leading and Discovery*.

Conner, M.O., & Herlocker, J. (1999). Clustering Items for Collaborative Filtering. *Proceedings of the ACM SIGIR Workshop on Recommender Systems*, UC Berkeley, Vol. 128.

Honda, K., Sugiura, N., Ichihashi, H., & Araki, S. (2001). Collaborative Filtering Using Principal Component Analysis and Fuzzy Clustering. *Web Intelligence: Research and Development*, Springer Berlin Heidelberg, pp. 394-402.

Chee, S.H.S. , Han, J., & Wang, K. (2001). Rectree: An efficient collaborative filtering method. *Data Warehousing and Knowledge Discovery*, Springer Berlin Heidelberg, pp. 141-151.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2002). Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. *Proceedings of the Fifth International Conference on Computer and Information Technology*, Vol. 1.

Bridge, D., & Kelleher, J. (2002). Experiments in sparsity reduction: Using clustering in collaborative recommenders. *Artificial Intelligence and Cognitive Science*, Springer Berlin Heidelberg, pp.144-149.

Kelleher, J., & Bridge, D. (2003). Rectree centroid: An accurate, scalable collaborative recommender". *Proceeding of AICS*.

Mamunur, R.A., Lam, S.K., Karypis, G., & Riedl, J. (2006). A Highly Scalable Hybrid Model- & Memory-Based CF Algorithm. *Proceeding of WebKDD*.

Wanqiu, H., Jia, X., Tian, F., Zhang, Y., & Zhou, Z. (2015). The Method of Finding Potentially Churning Customers Based on Social Networks. *International Journal of Multimedia and Ubiquitous Engineering*, 10(11), 95-10, 2015.

Liao, H., Chen, K., Liu, D., & Chiu, Y. (2015). Customer Churn Prediction in Virtual Worlds. *In Advanced Applied Informatics (IIAI-AAI), 4th International Congress on,* pp. 115-120.

Bi, W., Cai, M., Liu, M., & Li, G. (2016). A big data clustering algorithm for mitigating the risk of customer churn. *IEEE Transactions on Industrial Informatics,* 12(3), 1270-1281, 2016.

Aimée, B., Baesens, B., & Claeskens, G. (2016). Predicting time-to-churn of prepaid mobile telephone customers using social network analysis. *Journal of the Operational Research Society*. DOI 10.1057/jors.2016.8.

Fathian, M., Hoseinpoor, Y., & Minaei-Bidgoli, B. (2016). Offering a hybrid approach of data mining to predict the customer churn based on bagging and boosting methods. *Kybernetes*, 45(5), 732-743.

Ahmadi, A., Karray, F., & Kamel, M.S. (2010). Flocking based approach for data clustering. *Natural Computing*, 9(3), 297-321.

Renaud-Deputter, S., Xiong, T., & Wang, S. (2013). Combining collaborative filtering and clustering for implicit recommender system", *Proceeding of IEEE 27th International Conference on Advanced Information Networking and Applications (AINA*), pp. 748-755.

Shen, H., Jin, L., Zhu, Y., & Zhu, Z. (2010). Hybridization of particle swarm optimization with the K-Means algorithm for clustering analysis. *Proceeding of IEEE Fifth International Conference*, pp. 531-535.

Vora, P., & Oza, B. (2013). A Survey on K-mean Clustering and Particle Swarm Optimization.

Proceeding of International Journal of Science and Modern Engineering (IJISME), pp. 24-26.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceeding of IEEE international conference on Neural networks,* New Jersey, USA.

Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. *Proceedings of the sixth international symposium on micro machine and human science,* vol 1, pp. 39-43.

Sullivan, M. (2010). Fundamental of Statistics. 3rd Edition, *Loose Leaf*.

Abdollahpouri, H., & Abdollahpouri, A. (2013). An approach for personalization of banking services in multi-channel environment using memory-based collaborative filtering. *Proceeding of 5th Conference on Information and Knowledge Technology (IKT),* pp. 208-213.

Miller, B., Konstan, J., & Riedl, J. (2004). PocketLens: Toward a Personal Recommender System. *ACM Transactions on Information Systems (TOIS),* 22(3), 437-476.

Mamunur, R.A., George, R. , & Ried, K.J. (2005). Influence in Ratings Based Recommender Systems: an Algorithm-independent Approach. *Processing of SIAM International Conference on Data Mining*.

Songjie, G. (2010). A collaborative filtering recommendation algorithm based on user clustering and item clustering. *Journal of Software,* 5(7), 745-752.

Rui, X. , & Wunsch, D. (2005). Survey of Clustering Algorithms. *IEEE Transaction on Neural Networks*, 16(3), 645-678.