International Journal of Advanced Science and Technology
Vol.68 (2014), pp.11-22
http://dx.doi.org/10.14257/ijast.2014.68.02

Neighborhood Cleaning Rules and Particle Swarm Optimization for Predicting Customer Churn Behavior in Telecom Industry

Hossam Faris

Business Information Technology Department, the University of Jordan, Amman, Jordan
hossam.faris@ju.edu.jo

Abstract

Churn prediction is an important task for Customer Relationship Management (CRM) in telecommunication companies. Accurate churn prediction helps CRM in planning effective strategies to retain their valuable customers. However, churn prediction is a complex and challenging task. In this paper, a hybrid churn prediction model is proposed based on combining two approaches; Neighborhood Cleaning Rules (NCL) and Particle Swarm Optimization (PSO). NCL is applied in the preprocessing stage for handling the imbalanced churn data; and eliminating outliers and unrepresentative data. In the next stage, a Constricted PSO is applied for developing the final prediction model. The developed model is evaluated and compared with a baseline PSO model. The proposed hybrid model is compared also with Artificial Neural Networks (ANN) and Decision trees (DT) models which are traditional and common approaches used in the literature for churn prediction. The experimental results show that the proposed hybrid model outperforms the baseline PSO model, ANN and DT in terms of accuracy and actual churn rate.

Keywords: *Particle Swarm Optimization, Neighborhood Cleaning Rules, Churn prediction, Modeling, Classification, Telecommunication*

1. Introduction

In most telecommunication business markets where the competition is fierce, it is becoming easier for any customer to churn and switch to another service provider. The reasons behind customers' decision to leave their service provider can vary. Some reasons for example are related to the quality of customer service and coverage while others could be economic and related to the cost of service.

On the other hand, companies realize that the customer is the most valuable asset for them and losing customers means serious profit loss. Moreover, it is agreed that attracting new customers costs much more than retaining the existing ones [1, 2]. For all these reasons, companies and service providers are investing more in developing reliable models that are able to predict customers who are likely to leave. Accurate models can help customer relationship management (CRM) in designing more effective retention strategies targeting those customers.

In literature, researchers and practitioners have developed wide range of statistical and data mining based models in order to accurately predict customer churn. Data mining approaches applied for churn prediction include traditional classification methods like Decision trees algorithms, Naive Bayes and Logistic Regression [3–5]. It also includes artificial intelligence based approaches like Artificial Neural Networks, Genetic Programming and Support Vector Machines [6–8].

However, there are some challenges encounter researchers and practitioners while tackling the problem of churn behavior identification. One of the most important challenges is the imbalanced class distribution and the rarity of the class of interest. Like in many other marketing problems, churn data are imbalanced; where the number of churn customers (class of interest) is much less than the active ones. In such case, the instances from the class of interest are lost among the majority class which makes the learning process more difficult. This problem can lead learning algorithms to create classifiers that are unable to classify new rare classes. Therefore, applying common statistical and machine learning approaches could be inadequate [9].

For this reason, researchers used different techniques for handling imbalanced churn data; such techniques include adopting more appropriate evaluation metrics, using cost sensitive learning methods and sampling [10].

Most of the previous works considered applying one data mining technique for churn prediction [11]. However, few recent studies in the field showed that applying one more technique as a preprocessing stage to filter the data and eliminate representative samples can outperform the single method approach without any preprocessing [12, 13].

In this paper, a new hybrid approach for churn prediction is proposed. The approach is

based on combining Neighborhood Cleaning Rules (NCL) and Particle Swarm Optimization (PSO). NCL is applied in order to tackle the problem of the imbalanced distribution of the classes in the dataset. NCL performs data reduction by removing unrepresentative instances from the dataset. Then, PSO is used for the task of automatic classification rules discovery based on the reduced dataset.

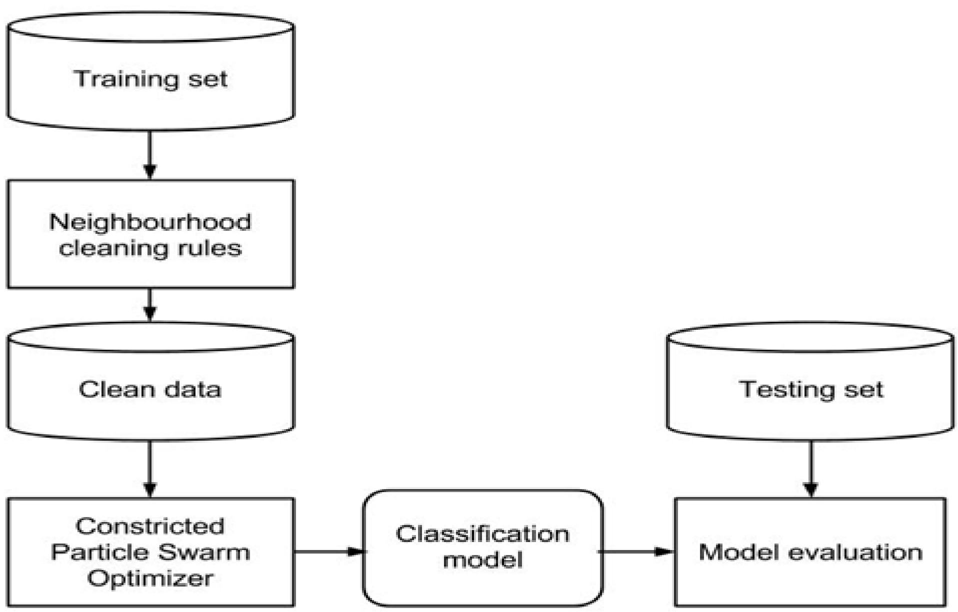

Figure 1. Neighborhood Cleaning Rules combined with Constricted Particle Swarm Optimization for Churn Prediction

2. The Proposed Hybrid Approach

The proposed approach for churn prediction in this work is illustrated in Figure 1. The main objective of this approach is to improve the capability of identifying customer churn behavior compared to traditional classification techniques. The proposed hybrid approach includes two main phases namely the data reduction phase and the classification phase. In the data reduction phase, Neighborhood Cleaning Rules (NCL) is applied to relieve the problem of the imbalance data distribution by eliminating outliers and unrepresentative data. In the classification phase, a Particle Swarm Optimizer (PSO) is used for the task of automatic classification rule discovery.

For NCL, different reduction pressures are tested in order to determine the best reduction level for the churn data. While for PSO, a Constricted PSO (CPSO) is applied using different sizes of swarms in order to obtain the maximum prediction power of the algorithm.

The applied hybrid approach is assessed using different evaluation criteria with five folds cross validation. Evaluation results are compared with basic CPSO algorithm and other traditional techniques commonly used in the literature which are the Multilayer Perceptron (MLP) neural network and the C4.5 Decision Tree algorithm.

In the following two sections, a brief description of the NCL and CPSO algorithms is given along with some key definitions and explanation of some customizations made in order to fit the investigated churn prediction problem.

3. Neighborhood Cleaning Rules (NCL)

NCL is an under sampling technique introduced by J. Laurikkala for the task of balancing imbalanced class distribution by performing data reduction [9]. NCL proved its efficiency in identifying difficult small classes [9, 14]. The main advantage of NCL is that it considers the quality of the removed data by focusing on data cleaning more than data reduction.

NCL is based on the concept of one-sided selection (OSS) which is a data reduction technique utilized when class distribution is imbalanced [15]. OSS applies instance-based methods to reduce the larger classes while the class of interest (the smaller class) is intact.

Suppose we have a dataset where C is a small class of interest and O is the rest if the data (majority class). NCL uses Wilson's edited nearest neighbor rule (ENN) [16] to reduce O by removing noisy data A_1 in O. ENN removes examples whose class differs from the majority class of at least two of its three nearest neighbors. Moreover, NCL increases cleaning by removing the three nearest neighbors that misclassify examples of C and belong to O. The NCL algorithm can be described as shown in Figure 2.

In the original version of the NCL algorithm, only samples from classes larger or equal to half size of class of interest are considered for A_2. This idea was proposed in order to avoid reduction of very small classes. Since the churn prediction problem is a binary classification problem and the non-churners class is much larger than churners' class, the latter condition is relaxed when NCL is applied in the churn prediction problem.

Neighborhood Cleaning Rule
1. Split data T into the class of interest C and the rest of data O.
2. Identify noisy data A_1 in O with edited nearest neighbor rule.
3. For each class C_i in O
if ($x \in C_i$ in 3-nearest neighbors of misclassified $y \in C$) and ($\lvert C_i \rvert > 0.5 \lvert C \rvert$) then $A_2 = \{x\} \cup A_2$
4. Reduced data $S = T - (A_1 \cup A_2)$

Figure 2. Neighborhood Cleaning Rule algorithm [9]

4. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a metaheuristic and evolutionary algorithm inspired by the intelligent social behavior of the coordinated motion in flocks of birds or schools of fish. In PSO, individual birds or fish are called individual while the group is called swarm [17-19].

The solution for a given problem is developed from the interactions between particles moving in an n-dimensional space to search for solutions. A solution in this case is considered as particle's location in the n-dimensional problem space.

In PSO algorithms, each particle in the swarm moves around by considering its own experience and the experience of its most successful neighbor. The experience here is the particle's memory of the best past position [17, 18, 23].

As described in Figure 3, PSO algorithm starts with a number of randomly initialized particles then it starts searching for the optimal solution through a predefined number of iterations. In every iteration, particles' fitness value is assessed using some evaluation criteria. Particles memorize their best past solution which is called personal best (*pBest*). Moreover, particles also keep tracking of their neighborhood associated with them and memorize the best fitness value achieved by any particle in this neighborhood. This best value is stored as global best (*gBest*).

Pseudocode for Particle Swarm Optimization
Initialize population (Swarm)
Do
For each particle
Evaluate fitness value
If the fitness value is better than its personal best ($pBest$)
Set current value as the new $pBest$
End
Choose the particle with the best fitness value of all as $gBest$
For each particle
Calculate particle velocity according Equation 1
Update particle position according Equation 2
Terminate when termination criterion is met or minimum error criteria is attained

Figure 3. Pseudo Code Algorithm for Particle Swarm Optimization

Based on the values of *pBest* and *gBest*, each particle moves around in the search space with a certain velocity calculated by Equations 1 and 2. Where c_1 and c_2 are positive constants. r_1 and r_2 are random positive weights drawn from a uniform distribution defined by an upper limit. V_i is the current velocity, V_{i-1} is the previous velocity, pL is the new location of the particle, pvL is the previous location of the particle.

$$V_i = \omega V_{i-1} + c_1 r_1 * (pBest - pL) + c_2 r_2 * (gBest - pL) \tag{1}$$

$$pL = pvL + V_i \tag{2}$$

PSO proved to be suitable algorithm for the rules discovery problem and classification tasks [20–22]. In this paper, three tiers PSO based application developed by Sousa and others in [23] are used for automatic classification rule discovery. The three tiers application is illustrated in Figure 4 and it can be summarized in following points:

- **The inner tier**, this tier implements the classification rule discovery algorithm which is responsible for searching the best set of rules (conditional clauses) that classifies correctly a number of given instances. In the experiments of this work, a Constricted PSO (CPSO) is applied. CPSO is a variation of the PSO algorithm whose velocity function type is constricted [24]. This constriction is added to force the algorithm to converge and avoid the explosion of the particle swarm where the particles velocities and positional coordinates run to infinity [24].

- **The middle tier**, in this tier a covering divide-and-conquer algorithm is implemented. This algorithm is responsible of calling the classification rule discovery algorithm in the inner tier in order to reduce the training set by removing the correctly classified instances by the rule returned by the inner algorithm. This process is repeated and a sequential set of rules is created consequently until a predefined number of instances are left to be correctly classified.
- **The outer tier**, here a validation algorithm is implemented to assess the accuracy of a rule set returned by the middle tier algorithm. Moreover, this tier gauges the liability of the previous two tiers by performing ten folds cross validation and computing their average accuracy, time spent, rule number per set and attribute tests number per rule.

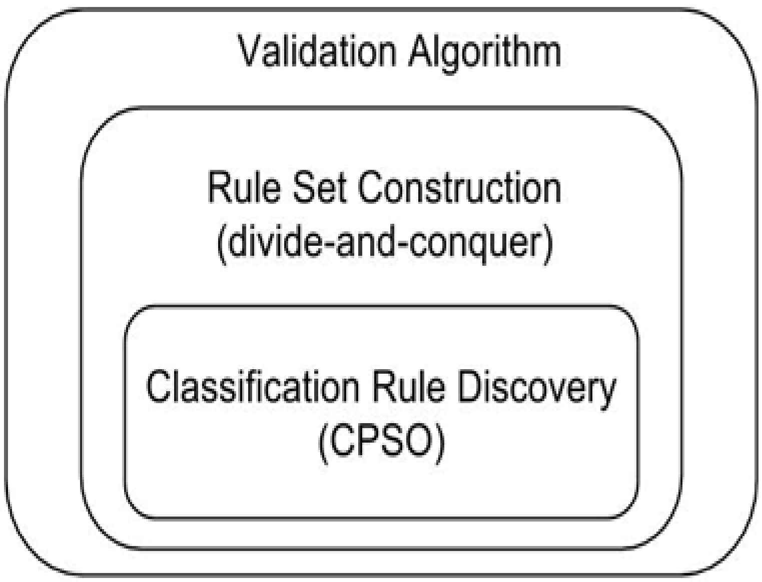

Figure 4. Three Tiers PSO based Application for Classification Tasks [22]

5. Dataset Description

The data set used for this work was provided by a major Jordanian cellular telecommunication network. The dataset contains 11 attributes of randomly selected 5000 customers for a time interval of three months. The last attribute indicates whether the customer churned (left the company) or not. The total number of churners is 381 (7.6% of total customers). The attributes along with their description are listed in Table 1.

The data is normalized by dividing each variable by its standard deviation. Normalization is recommended when data variables follow different dynamic ranges. Therefore, to eliminate the influence of larger values, normalization is applied to make all variables lie in the same scale.

Table 1. Dataset Attributes

Attribute name	Description
3G	The subscriber is provided with 3G service (Yes, No)
Total Consumption	Total monthly fees (calling +SMS) (JD)
Calling fees	Total monthly calling fees (JD)
Local SMS fees	Monthly local SMS fees (JD)
Int'l SMS fees	Monthly fees for international SMS (JD)
Int'l calling fees	Monthly fees for international calling (JD)
Local SMS count	Number of monthly local SMS
Int'l SMS count	Number of monthly international SMS
Int'l MOU	Total of international outgoing calls in minutes
Total MOU	Total minutes of use for all outgoing calls
On net MOU	Minutes of use for on-net-outgoing calls
Churn	Churning customer status (Yes, No)

6. Model Evaluation Criteria

In order to evaluate the developed model, the confusion matrix shown in Table 2 is used since it is considered as the primary source for accuracy estimation in classification problems. Based on this confusion matrix, the following two criteria are used for evaluation:

Table 2. Confusion Matrix

		Actual	
		Non churn	Churn
Predicted	Non churner	A	B
	Churner	C	D

1. Accuracy: Identifies the percentage of the total number of predictions that were correctly classified.

$$Accuracy = \frac{A + D}{A + B + C + D}$$

2. Actual churners rate (Coverage rate): The percentage of predicted churn in actual churn. It can be given by the following equation:

$$Actual\ churn\ rate = \frac{D}{B + D}$$

Since the churn data is imbalanced, using only the Accuracy ratio for evaluating the developed classification model is inadequate. Therefore, Actual churn rate is considered for evaluating to give more attention to the rare class which is the churn class.

7. Experiments and Results

Experiments are conducted in main three stages as follows:

7.1. Basic CPSO Model

In general, PSO is a parametric approach and the values of these parameters have crucial impact on its performance and accuracy. One of these important parameters is the number of particles. In order to find the best number for particles for PSO that achieves best accuracy and best churn rate, eight different experiments were conducted by applying CPSO with different number of particles each time (*i.e.*, 25,50,75,...,200).

Table 3. CPSO Parameters Settings

Parameter	Value
Convergence radius	0.1
Weights upper limit	2.05
Maximum uncovered instances	0.1
Indifference threshold	0.1
Constriction coefficient	0.73
Convergence platform width	30

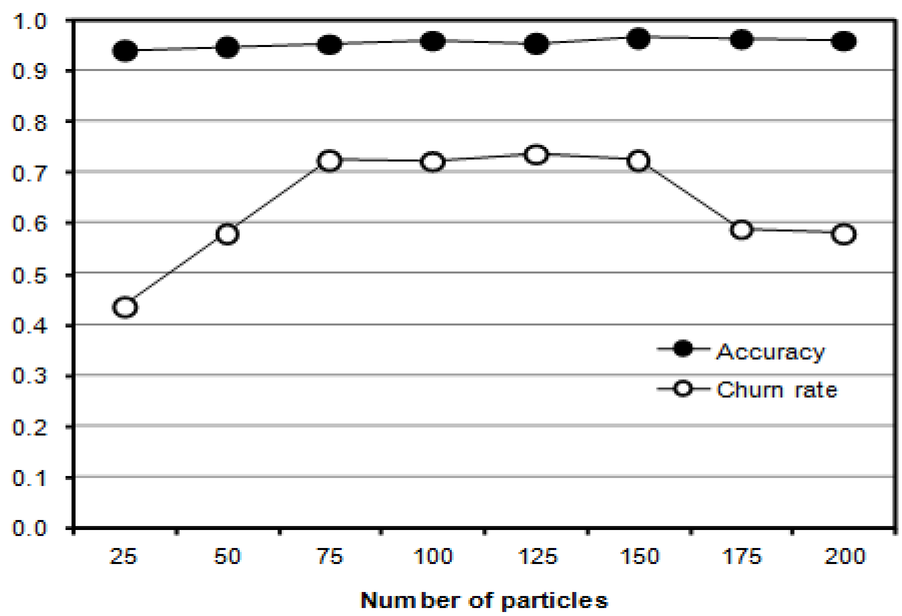

Figure 5. Accuracy and Churn Rate Values for CPSO Models using Different Number of Particles in each Experiment

The experiments were performed using the settings listed in Table 3. Five folds cross validation was applied to obtain a better indication of how well the developed classification model will perform when new data is presented. Results of the experiments are shown in Figure 5. It can be noticed that number of particles has almost no effect on the accuracy ratio while churn rate significantly increases when the number of particles is 75 then it goes steady until 150 then it starts to decrease again. CPSO with 75 particles developed a model with accuracy and churn rate of 95.2% and 72.4%, respectively. Therefore, number of particles is set to 75 for the next experiments since it achieved best accuracy and churn rate, and it needs less execution time than larger numbers of particles.

7.2. NCL+CPSO Model

In this stage of the experiments, a new model is developed by combining NCL with CPSO, NCL is applied in order to perform the data reduction process then CPSO is applied on the reduced data set to develop the final model. However, NCL performance is affected by the number of neighbors *K* which are the nearest instances considered to generate a new positive example. Therefore, NCL was applied on the dataset different times with different number of *K* each time (*i.e.*, *K*=3, 5, 7...,19). The amounts of reduction made by NCL with different *K* sizes are shown in Table 4. Then, CPSO is applied on the reduced data to find the best *K* that leads to best classification results. In Figure 6, it can be seen that increasing *K* decreases the accuracy rate slightly from 95% reaching around 90%; while larger *K* values significantly improves the actual churn rate which reaches the maximum at 82.7% for *K* =15.

Table 4. Amount of Reduction Performed by NCL for Different K Sizes

K size	3	5	7	9	11	13	15	17	19
Reduction	6.14%	8.68%	12.86%	15.72%	18.26%	20.63%	22.65%	24.50%	26.06%

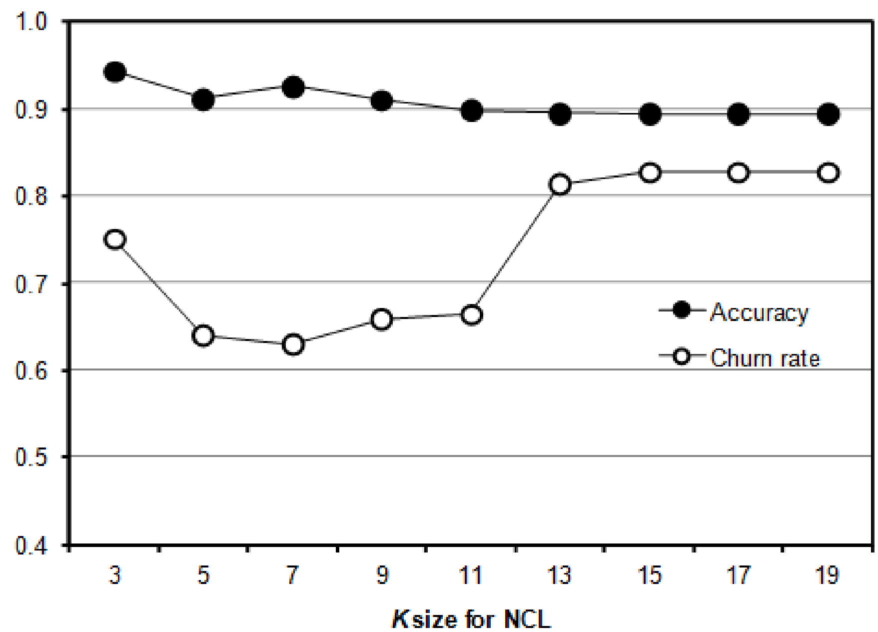

Figure 6. Accuracy and Churn Rate Values for CPSO Models after Applying NCL with Different *K* Values in each Experiment

7.3. Comparison

The best results obtained for basic CPSO and NCL+CPSO are compared with those for Decision tree algorithm C4.5 and MLP neural network. For MLP, the back propagation algorithm was used for training the neural network. Empirically, it was found that the best MLP performance can be obtained by setting four neurons in the hidden layer with 5000 epochs and learning rate of 0.2 to train through. All developed models are evaluated using the accuracy and churn rate criteria. Results of evaluation are shown in Figures 7 and 8 respectively.

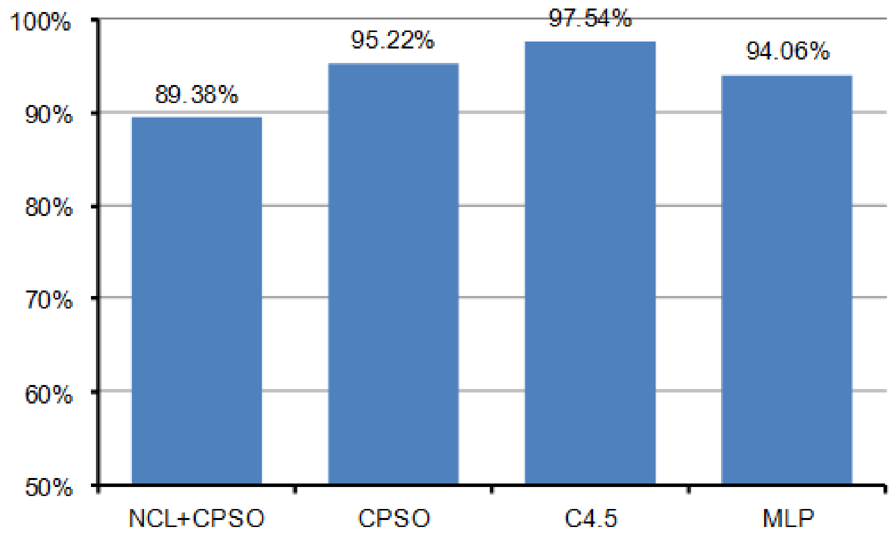

Figure 7. Accuracy Values for Alternative Prediction Approaches

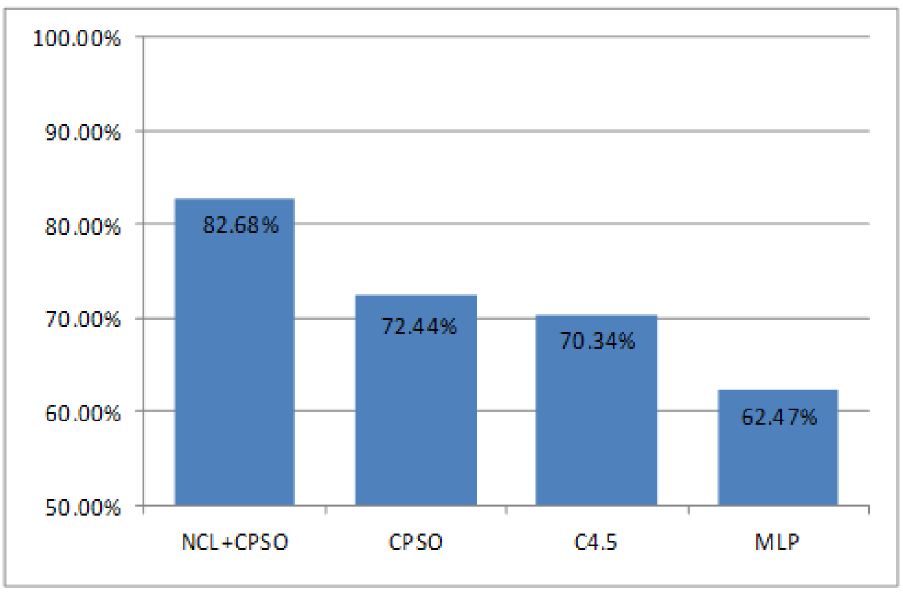

Figure 8. Churn Rate Values for Alternative Prediction Approaches

According to Figure 7 it can be noticed that CPSO achieved the second best accuracy rate compared to the other techniques with 2.3% less than C4.5. On the other hand, although NCL+CPSO decreased the accuracy rate of CPSO with around 5% it achieved excellent results for the actual churn rate as shown in Figure 8 compared to CPSO, C4.5 and MLP with improvement of 12% and 20%, respectively. It can be concluded from these results that CPSO has promising prediction power compared to traditional techniques used in the literature for predicting customer churn. Moreover, applying NCL as a preprocessing technique for cleaning the data from unrepresentative data has promising potential for improving actual churn rates.

8. Conclusion

In this paper, we propose a new hybrid approach for predicting customer churn behavior in telecommunication companies. The hybrid approach is based on combining Neighborhood Cleaning Rules and Constricted Particle Swarm Optimization. Neighborhood Cleaning Rules are used as a preprocessing stage for data reduction and cleaning the data from unrepresentative data. While the Constricted Particle Swarm Optimization is used for building the final predictive model. The hybrid model is evaluated using different evaluation criteria then compared to traditional techniques used for churn prediction in the literature. The experimental results show that the new hybrid model has promising prediction results compared to other techniques.

.

References

[1] J. Hadden, A. Tiwari, R. Roy and D. Ruta, "Computer assisted customer churn management: State-of-the-art and future trends", Computers and Operations Research, vol. 34, no. 10, **(2007)**, pp. 2902-2917.
[2] H.-S. Kim and C.-H. Yoon, "Determinants of subscriber churn and customer loyalty in the Korean mobile telephony market", Telecommunications Policy, vol. 28, no. 9, **(2004)**, pp. 751-765.
[3] B. Huang, M. T. Kechadi and B. Buckley, "Customer churn prediction in telecommunications", Expert Syst. Appl., vol. 39, **(2012)** January, pp. 1414-1425.
[4] M. Owczarczuk, "Churn models for prepaid customers in the cellular telecommunication industry using large data marts", Expert Systems with Applications, vol. 37, no. 6, **(2010)**, pp. 4710-4712.
[5] G. Li and X. Deng, "Customer churn prediction of china telecom based on cluster analysis and decision tree algorithm", in Emerging Research in Artificial Intelligence and Computational Intelligence (J. Lei, F. Wang, H. Deng, and D. Miao, eds.), Communications in Computer and Information Science, Springer Berlin Heidelberg, **(2012)**, pp. 319–327.
[6] O. Adwan, H. Faris, K. Jaradat, O. Harfoushi and N. Ghatasheh, "Predicting customer churn in telecom industry using multilayer preceptron neural networks: Modeling and analysis", Life Science Journal, vol. 11, no. 3, **(2014)**, pp. 75-81.
[7] R. Obiedat, M. Alkasassbeh, H. Faris and O. Harfoushi, "Customer churn prediction using a hybrid genetic programming approach", Scientific Research and Essays, vol. 8, no. 27, **(2013)**, pp. 1289–1295.
[8] X. Yu, S. Guo, J. Guo and X. Huang, "An extended support vector machine forecasting framework for customer churn in e-commerce", Expert Systems with Applications, vol. 38, no. 3, **(2011)**, pp. 1425-1430.
[9] J. Laurikkala, "Improving identification of difficult small classes by balancing class distribution", in Artificial Intelligence in Medicine (S. Quaglini, P. Barahona and S. Andreassen, eds.), Lecture Notes in Computer Science, Springer Berlin Heidelberg, vol. 2101, **(2001)**, pp. 63-66.
[10] J. Burez and D. Van den Poel, "Handling class imbalance in customer churn prediction", Expert Syst. Appl., vol. 36, **(2009)** April, pp. 4626-4636.
[11] C.-F. Tsai and Y.-H. Lu, "Customer churn prediction by hybrid neural networks", Expert Syst. Appl., vol. 36, **(2009)** December, pp. 12547-12553.
[12] A. Idris, M. Rizwan and A. Khan. "Churn prediction in telecom using Random Forest and PSO based data balancing in combination with various feature selection strategies", Comput. Electr. Eng, vol. 38, no. 6, **(2012)**, pp. 1808-1819.
[13] I. Bose and X. Chen, "Hybrid models using unsupervised clustering for prediction of customer churn", Journal of Organizational Computing and Electronic Commerce, vol. 19, no. 2, **(2009)**, pp. 133-151.
[14] J. Laurikkala, "Instance-based data reduction for improved identification of difficult small classes", Intell. Data Anal., vol. 6, **(2002)** September, pp. 311-322.
[15] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: one-sided selection", in Proc. 14th International Conference on Machine Learning, **(1997),** pp. 179–186, Morgan Kaufmann.
[16] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data", Systems, Man and Cybernetics, IEEE Transactions on, vol. SMC-2, **(1972)** July, pp. 408-421.
[17] J. Kennedy and R. Eberhart, "Particle swarm optimization", in Neural Networks, 1995. Proceedings, IEEE International Conference on, vol. 4, **(1995)** November, pp. 1942-1948.
[18] J. Kennedy, "The particle swarm: Social adaptation of knowledge", in Proceedings of the 1997 International Conference on Evolutionary Computation, IEEE Service Center, Piscataway, NJ, **(1997)**, pp. 303-308.
[19] S. Alian, N. Ghatasheh and M. Abu-Faraj, "Multi-Agent Swarm Spreading Approach in Unknown Environments", International Journal of Computer Science Issues, vol. 11, no. 2, **(2014)**, pp. 160–168.

[20] Y. Liu, Z. Qin, Z. Shi and J. Chen, “Rule discovery with particle swarm optimization”, in Content Computing (C.-H. Chi and K.-Y. Lam, eds.), of Lecture Notes in Computer Science, Springer Berlin Heidelberg, vol. 3309, **(2004)**, pp. 291-296.
[21] Z.Wang, X. Sun and D. Zhang, “Classification rule mining based on particle swarm optimization”, in Rough Sets and Knowledge Technology (G.-Y. Wang, J. Peters, A. Skowron, and Y. Yao, eds.), of Lecture Notes in Computer Science, Springer Berlin Heidelberg, vol. 4062, **(2006)**, pp. 436-441.
[22] K. Gandhi, M. Karnan and S. Kannan, “Classification rule construction using particle swarm optimization algorithm for breast cancer data sets”, in Signal Acquisition and Processing, ICSAP ’10. International Conference on, pp, **(2010)** February, pp. 233-237.
[23] T. Sousa, A. Silva, and A. Neves, “Particle swarm based data mining algorithms for classification tasks,” Parallel Comput., vol. 30, May **(2004)**, pp. 767–783.
[24] M. Clerc and J. Kennedy, “The particle swarm - explosion, stability, and convergence in a multidimensional complex space”, Evolutionary Computation, IEEE Transactions on, vol. 6, **(2002)** February, pp. 58-73.