Egyptian Informatics Journal 18 (2017) 215–220

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Churn prediction on huge telecom data using hybrid firefly based classification

Ammar A. Q. Ahmed *, Maheswari D.

Rathnavel Subramainam College of Arts & Science, Coimbatore, Tamil Nadu, India

ARTICLE INFO

Article history:
Received 25 September 2016
Accepted 10 February 2017
Available online 1 March 2017

Keywords:
Firefly algorithm
Simulated annealing
Telecom churn prediction
Data imbalance
Data sparsity
Huge data

ABSTRACT

Churn prediction in telecom has become a major requirement due to the increase in the number of telecom providers. However due to the hugeness, sparsity and imbalanced nature of the data, churn prediction in telecom has always been a complex task. This paper presents a metaheuristic based churn prediction technique that performs churn prediction on huge telecom data. A hybridized form of Firefly algorithm is used as the classifier. It has been identified that the compute intensive component of the Firefly algorithm is the comparison block, where every firefly is compared with every other firefly to identify the one with the highest light intensity. This component is replaced by Simulated Annealing and the classification process is carried out. Experiments were conducted on the Orange dataset. It was observed that Firefly algorithm works best on churn data and the hybridized Firefly algorithm provides effective and faster results.

1. Introduction

Increase in the number of telecom providers has led to a huge rise in competition and hence customer churn. Currently organizations have their major focus on reducing the churn by focusing on customers independently. Churn [1] can be defined as the propensity of a customer to cease business transactions with an organization. The major requirement now is identification of customers who have high probabilities of moving out. The ability of an organization to intervene at the right time could effectively reduce churn.

Churn occurs mainly due to customer dissatisfaction. Identifying customer dissatisfaction requires several parameters. A customer usually does not churn due to a single dissatisfaction scenario [2]. There usually exist several dissatisfaction cases before a customer completely ceases to do transactions with an organization. Several properties associated with the customer and their mode of operations with the organization are recorded by the organizations. This represents the customer's behavior data. Analyzing this data would present a clear view of the customer's current status [3]. Hence this can be used as the base data for churn prediction. The major difficulty arising from this mode of operation is that the data under discussion tends to be very huge. The hugeness can be attributed to the behavioral nature of the data, depicting all the product lines dealt with by the organization. Further, due to the requirement of structural representation of the data, all the instances are bound to contain all the properties corresponding to a generic customer in the organization [4,5]. This leads to data sparseness, since customers will be associated with only a few properties and not all the properties pertaining to the organization. The hugeness of data and sparsity acts as the major difficulties in the process of churn prediction.

Large companies interact with their customers to provide a variety of services to them [6]. Customer service is one of the key differentiators for companies. The ability to predict if a customer will leave in order to intervene at the right time can be essential for pre-empting problems and providing high level of customer service. The problem becomes more complex as customer behavior data is sequential and can be very diverse.

Churn is an unavoidable process in any industry. However, though difficult, it is possible to identify the causes of churn using several approaches.

* Corresponding author.
E-mail addresses: ammar.aqahmed@gmail.com (A.A.Q. Ahmed), mahelenin@gmail.com (D. Maheswari).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.eij.2017.02.002
1110-8665/

2. Related work

This section discusses the recent approaches for churn prediction. A risk prediction technique that identifies probable customers for churn was presented by Coussement et al. in [7]. This technique utilizes Generalized Additive Models (GAM). These models relaxe the linearity constraints, hence allowing complex non-linear fits to the data. This technique is exhibited to improve marketing decisions by identifying the risky customers and also providing visualizations of non-linear relationships.

A neural network based customer profiling technique that can be used for churn prediction was presented by Tiwari et al. in [8]. This technique differs from the other proposed techniques by the fact that most of the techniques are only able to identify the customers who will instantaneously churn. However the neural network based churn prediction model proposes to predict customer's future churn behavior, providing the much required buffer for the organizations to perform prevention activities. A similar neural network based model includes [22,24]. The approach in [22] is based on the 80-20 rule to identify the key attributes affecting churn, while that of [24] involves identifying the major features of the data to determine churn.

A regression based churn prediction model was presented by Awnag et al. in [9]. This method identifies churn by using multiple regressions analysis. This technique utilizes the customer's feature data for analysis and proposes to provide good performance.

Class imbalance plays a major role in affecting the reliability of a classifier. The major issue existing due to class imbalance is that the minority class is not well represented and hence the classifier is undertrained on the minority classes. The technique proposed by Zhu et al. in [10] proposes to eliminate this issue by using transfer learning techniques. The approach presented in [10] operates by training the classifier using customer related behavioral data obtained from related domains. This approach has its major focus on the banking industry and the results are proposed to exhibit enhanced performance. Another technique that considers the imbalance nature of data to perform churn prediction was presented by Xiao et al. in [15]. A comparison of sampling techniques for effectively operating on churn data was presented by Amin et al. in [16]. Game theory based churn prediction techniques [17] are also on the raise.

The complex nature of churn behavior has also enabled several publications on churn prediction using multiple models. A churn prediction model based on cluster analysis and decision tree algorithm was presented by Li et al. in [11]. This technique operates on China's Telecom data. Another technique utilizing multiple prediction techniques was proposed by Le et al. in [12]. This technique utilized a combination of k-Nearest Neighbor algorithm and sequence alignment. This technique has its major focus on the temporal categorical features of the data to predict churn.

Utilizing heuristics for predictions are on the raise due to the complex nature of data. A rule generation techniques that employs heuristics for customer churn prediction in telecom services was presented by Huang et al. in [13]. A combination of Self Organizing Maps (SOM) and Genetic Programming (GP) to identify and predict churn was presented by Faris et al. in [14]. SOM is utilized to cluster the customers and then outliers are eliminated to obtain clusters depicting customer behaviors. An enhanced classification tree is built using GP.

A boosting algorithm that proposes to improve the prediction accuracy of classifier models was proposed by Lu et al. in [18]. This method boosts the learning process by using a combination of clustering and logistic regression. A similar prediction boosting technique using Genetic Algorithm was proposed by Idris et al. in [19]. This is also an ensemble model utilizing multiple techniques for the prediction process. Other ensemble based prediction techniques include [20,21,1,23].

3. Churn prediction on huge data using hybrid firefly based classification

Churn prediction on huge data utilizes Hybrid Firefly algorithm to effectively identify churn. This technique modifies the comparison component of the actual firefly algorithm with Simulated Annealing to provide faster and effective results.

A. Firefly algorithm: WorkingFirefly algorithm [25] is a nature inspired metaheuristic algorithm that was inspired by the behavior of fireflies attracting other fireflies by flashing lights. The intensity of the light plays a major role in determining the attractiveness of a firefly. It works on the following assumptions:

- All fireflies are unisexual, hence any firefly can be attracted to any other firefly.
- Attractiveness is proportional to the brightness of a firefly.
- For any two fireflies, the brighter one will attract the other.
- Brightness decreases as the distance between the fireflies increase.
- If no firefly is brighter than a given firefly, then it moves randomly.

For an optimization problem, the brightness of a firefly is associated with the objective function. The objective function contains all the parameters dependent on applications, hence expresses the degree of importance that the current solution holds.

B. Firefly algorithm: pros and cons

Firefly algorithm, due to its metaheuristic nature, can effectively identify optimal solutions when compared to other statistics based classification algorithms. Movement of the fireflies are directed by the intensity of the fireflies, provided by the firefly intensity parameter. The usage of a single dependent parameter leads to lesser memory requirements, hence this algorithm is capable of operating on huge data.

The major drawbacks of this algorithm is that for every iteration, a firefly is compared with every other firefly in the system [26], hence increasing the number of computations. Hence as the number of fireflies in the search space increases, the level of computations also increases to a large extent.

C. Hybrid Firefly: architecture

The hybrid firefly architecture is proposed to eliminate the problem of huge computational requirements due to comparisons. The working of hybrid firefly algorithm is presented in Fig. 1.

Building the search space marks the beginning of the classification process. The initial population of fireflies is generated and are distributed across the search space. The distribution of fireflies is carried out in random. Position of each firefly is recorded and the initial intensity of the fireflies (Intensity) are identified on the basis of their distance from the test data.

$$Intensity_i = 1/\sqrt{\sum_{j=1}^{attr}(X_{test,j} - X_{i,j})^2} \tag{1}$$

where $X_{test,j}$ refers to the j^{th} attribute of the test data and $X_{i,j}$ refers to the j^{th} attribute of the firefly i.

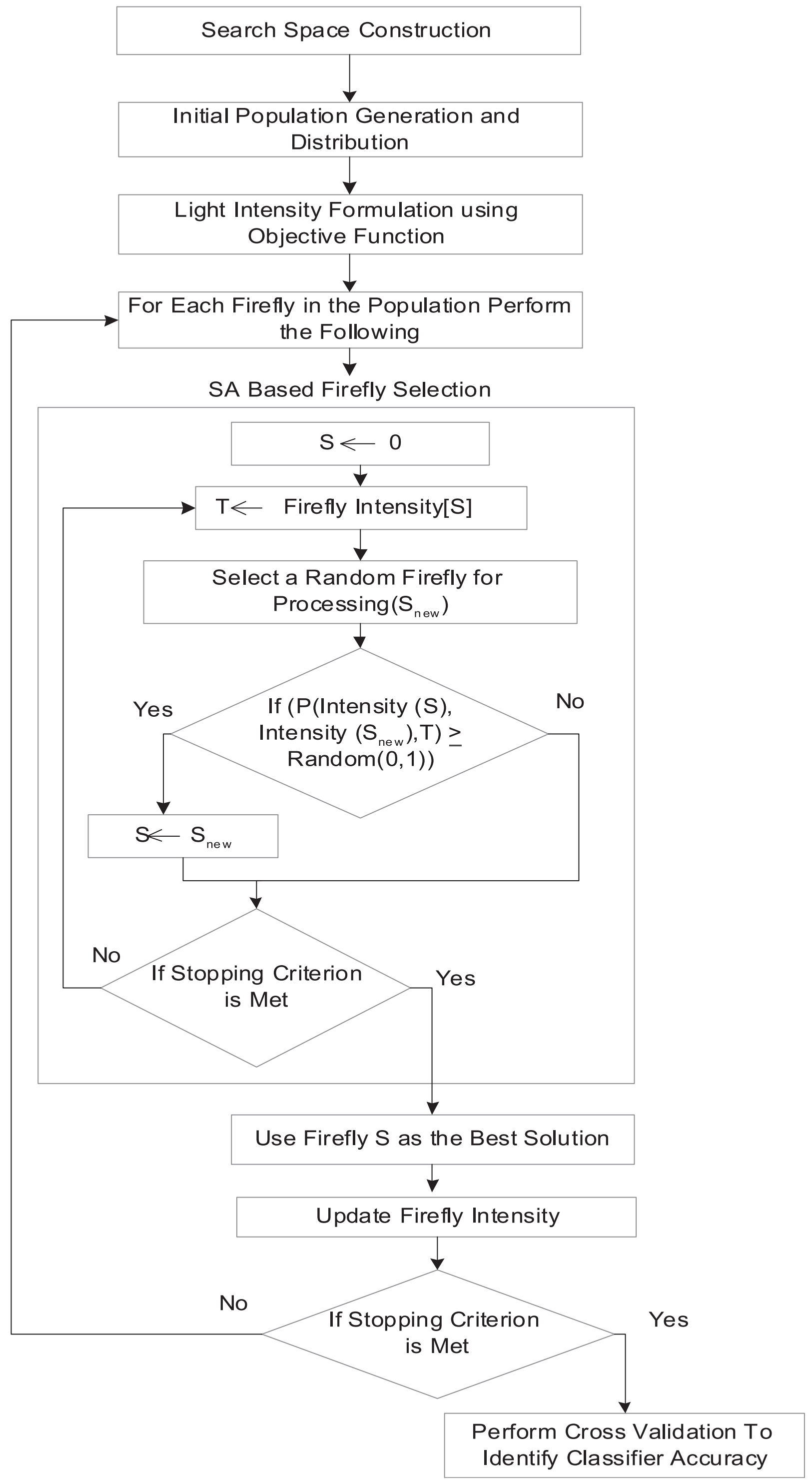

Fig. 1. Hybrid firefly ARCHITECTURE.

Firefly intensities along with the test data are passed to the Simulated Annealing module to identify the optimal solution for the test data. Firefly 0 is placed on the test data and the remaining fireflies are distributed on the training set.

Algorithm (Hybrid Firefly with Simulated Annealing):

1. *Search space boundary identification using base data*
2. *Firefly population generation (ffCount)*
3. *For each firefly i = 1...ffCount*
   a. *Firefly initialization*
   b. *Firefly distribution using uniform distribution function*
   c. *FireflyPosition[0] ← Test data*
4. *Until the termination criterion is met perform the following*
   d. *Index ← simulatedAnnealing(fireflyIntensity, ffCount)*
   e. *If the intensity of firefly in index is greater than the intensity of the firefly in the test data, move firefly[0] to index.*
   f. *Calculate new intensity using eq. (2)*
5. *Perform steps 3 and 4 for all the test data*

Simulated Annealing(fireflyIntensity, ffCount)

1. *Let s = 0*
2. *For k = 0 through ffCount:*
   a. *T ← fireflyIntensity[s]*
   b. *s_{new} ← Pick a random firefly*
   c. *If P(fireflyIntensity (s), fireflyIntensity (s_{new}), T) ≥ random(0, 1), move to the new state:*
      - *s ← s_{new}*
3. *Output: the final state s*

P(e,e′,T) was defined as 1 if e′ < e and exp(-(e′-e)/T) otherwise.

Simulated Annealing [27,28] is a probabilistic technique used to identify a global optimum of a given objective function. In this approach, the firefly intensities are considered as the objective function and the requirement for the algorithm is to identify the firefly with maximum intensity for the firefly containing the test data. Identifying the firefly with maximum intensity will directly correspond to the optimal solution and hence the best classification. Simulated Annealing is assumed to perform best on a discrete search space with large number of solutions. Since the process of identification of fireflies correspond to a similar scenario, using Simulated Annealing is applied here to identify the best firefly corresponding to the test data.

The intensity values of all the fireflies is passed to the Simulated Annealing module and the intensity of the resultant best firefly is compared with the test data to identify the firefly with maximum intensity ($Intensity_{max}$). If the resultant firefly has higher light intensity when compared to the firefly containing test data, the firefly containing test data is moved towards the firefly with the best solution. The light intensity of the firefly containing the test data ($Intensity_{test}$) is updated as

$$Intensity_{test} = Intensity_{test} + (\beta * Exp(-\gamma * Intensity_{max}) * (Intensity_{max} - Intensity_{test})) + (\alpha + \varepsilon) \tag{2}$$

where β is of order 1 (ideally), α is the parameter controlling the step size, γ is the absorption coefficient and ε is a vector drawn from a Gaussian distribution.

This process is continued until the specified stopping criterion is met. Stopping criterion is usualy set with two conditions. The operations are terminated when a specified maximum generations (*maxgen*) have been reached, or if the system does not move to a better solution for a specified number of iterations. Criteria of the first type is usually set in a production environment, while the second type is set during development to identify the time complexity. This process is carried out for each of the test data. Cross validation is finally performed to identify the accuracy of the classifier.

4. Results and discussion

Firefly algorithm and the Hybrid Firefly algorithm were implemented using C#.Net on Visual Studio 2012. Experiments were conducted with the Orange Dataset on both Firefly and the Hybrid Firefly algorithms. Orange is a benchmark dataset that corresponds to a French Telecom company [29]. It was used as a part of KDD 2009 challenge [30]. An analysis of the Orange dataset is presented in Table 1.

The dataset was segregated with 90% data for training and 10% of the data for testing. The search space was populated with 20 fireflies and classification was carried out with a *maxgen* of 1000.

The ROC plot obtained by classifying Orange data using the Hybrid Firefly algorithm is presented in Fig. 2. It could be observed from the figure that the plots are concentrated in two areas, top left and the top right. It could be interpreted that the algorithm exhibits very high True Positive Rates (TPR), i.e. it performs excellent classification of the positive cases. The False Positive Rates (FPR) are found to be low initially, however, finally the false positives show a huge increase.

Plot depicting Precision and Recall are presened in Fig. 3. Precision refers to the fraction of retrieved instances that are relevant and recall refers to the fraction of relevant instances that are retrieved. High values for precision and recall exhibits high performance levels of the algorithm. It could be observed from the figure

Table 1
Dataset analysis.

Property	Orange small
Attribute density	230
No of records	50,000
Missing values	60%
No of numerical Attributes	190
No of categorical attributes	40

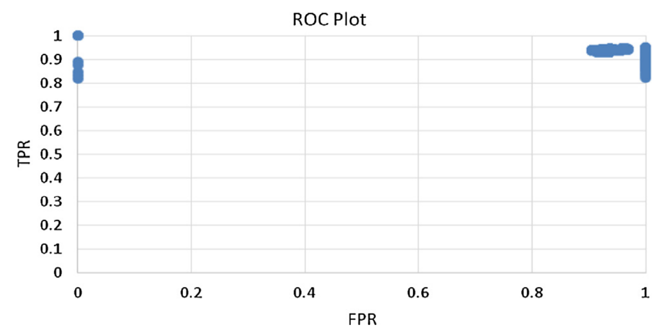

Fig. 2. ROC plot.

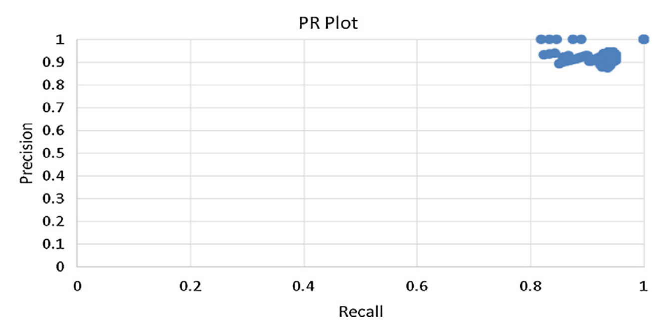

Fig. 3. PR plot.

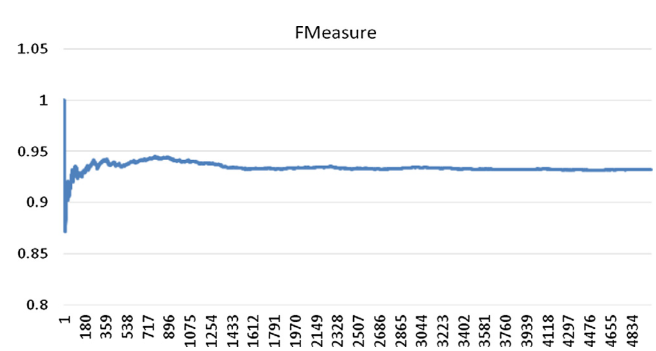

Fig. 4. F-Measure.

that the precision levels range from 0.85 to 1 and the recall levels range from 0.8 to 1 exhibiting very high performance levels.

F-Measure or F1 score is a measure of the accuracy exhibited by the classifier. It considers both precision and recall and can be computed as

$$F_1 = 2.\frac{precision.recall}{precision + recall}$$

It could be observed from the figure that the F-Measure ranges from 0.855 to 1, depicting high accuracy levels (see Fig. 4).

5. Comparative study

Comparison was carried out by applying the Orange data on Firefly algorithm, with 90% data for training and 10% for testing. The search space was populated with 20 fireflies and classification was carried out with a *maxgen* of 1000. Figures represent the ROC Plot, PR plot and the F-Measure obtained by using the normal Firefly algorithm.

It could be observed from the ROC plot (Fig. 5) that the FPR levels of Firefly algorithm are much higher when compared to the hybrid firefly algorithm. The PR plots (Fig. 6) exhibit very similar performance levels compared to hybrid firefly. Though the algorithm exhibits slightly low F-Measure levels (Fig. 7), it is still comparable to the hybrid firefly algorithm.

A comparison of the accuracy obtained from firefly and the hybrid firefly algorithm is presented in Fig. 8. It could be observed that the hybrid firefly algorithm exhibits slightly higher accuracy of 86.38% when compared to the firefly algorithm (86.36%).

A time comparison between normal firefly and the hybrid firefly is presented in Fig. 9. It could be observed that the time taken for a normal firefly algorithm is approximately 349.4 min, while that of the hybrid firefly algorithm is 2.5 min. This exhibits the efficiency of the hybridization process.

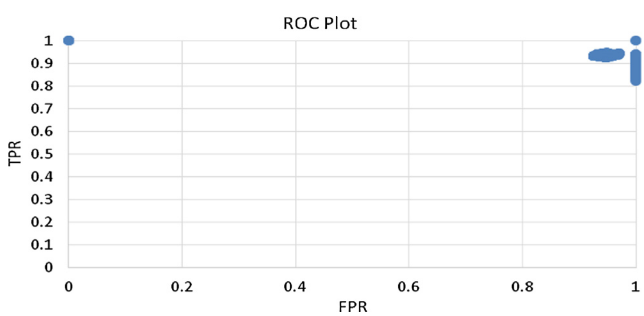

Fig. 5. ROC plot.

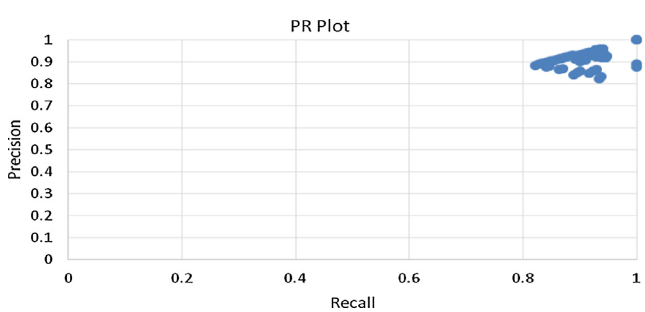

Fig. 6. PR plot.

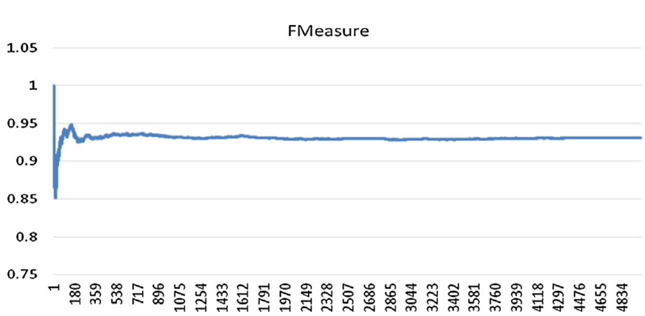

Fig. 7. F-Measure.

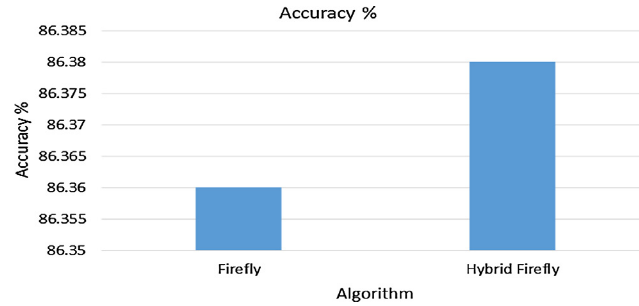

Fig. 8. Accuracy%.

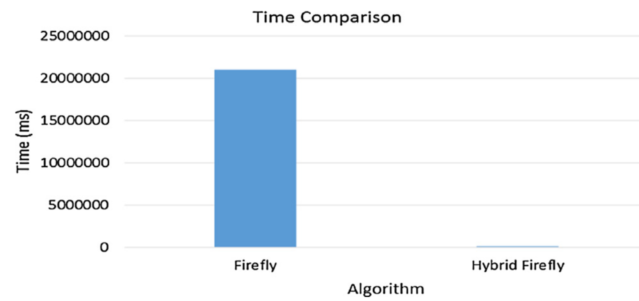

Fig. 9. Time comparison.

6. Conclusion

Churn prediction is one of the major requirements of the current competitive environment. This paper deals with identifying and predicting churn in the telecom data. This paper presents an efficient hybridized firefly algorithm for churn prediction. A comparison was carried out between the normal firefly algorithm and the proposed algorithm and it was identified that even though the accuracy exhibited by them are similar, hybrid firefly outperforms the normal firefly algorithm by exhibiting very low time latency. Analysis of the algorithms was carried out on the basis of ROC, PR, F-Measure, Accuracy and Time. Future directions will include incorporation of schemes or modifications to reduce the False Positive rates. Further, analysis in terms of imbalance levels and data sparsity will also be carried out. Incorporation of Game theory in the decision making process will also help improve the accuracy levels and in the identification of churn.

References

[1] Effendy V, Baizal ZA. Handling imbalanced data in customer churn prediction using combined sampling and weighted random forest. In: 2014 2nd International Conference on Information and Communication Technology (ICoICT), IEEE; 2014. p. 325–30.

[2] Seo D, Ranganathan C, Babad Y. Two-level model of customer retention in the US mobile telecommunications service market. Telecommun Policy 2008;32 (3):182–96.

[3] Hung SY, Yen DC, Wang HY. Applying data mining to telecom churn management. Expert Syst Appl 2006;31(3):515–24.
[4] Canning G. Do a value analysis of your customer base. Ind Mark Manage 1982;11(2):89–93.
[5] Eriksson K, Vaghult AL. Customer retention, purchasing behavior and relationship substance in professional services. Ind Mark Manage 2000;29 (4):363–72.
[6] Bhattacharya CB. When customers are members: customer retention in paid membership contexts. J Acad Mark Sci 1998;26(1):31–44.
[7] Coussement K, Benoit DF, Van den Poel D. Preventing customers from running away! Exploring generalized additive models for customer churn prediction. In: The Sustainable Global Marketplace. Springer International Publishing; 2015. p. 238-238.
[8] Tiwari A, Hadden J, Turner C. A new neural network based customer profiling methodology for churn prediction. In: Computational Science and Its Applications–ICCSA 2010. Berlin Heidelberg: Springer; 2010. p. 358–69.
[9] Awang MK, Rahman MNA, Ismail MR. Data mining for churn prediction: multiple regressions approach. In: Computer applications for database, education, and ubiquitous computing. Berlin Heidelberg: Springer; 2012. p. 318–24.
[10] Zhu B, Xiao J, He C. A balanced transfer learning model for customer churn prediction. In: Proceedings of the eighth international conference on management science and engineering management. Berlin Heidelberg: Springer; 2014. p. 97–104.
[11] Li G, Deng X. Customer churn prediction of China telecom based on cluster analysis and decision tree algorithm. In: Emerging Research in Artificial Intelligence and Computational Intelligence. Berlin Heidelberg: Springer; 2012. p. 319–27.
[12] Le M, Nauck D, Gabrys B, Martin T. KNNs and sequence alignment for churn prediction. In: Research and development in intelligent systems XXX. Springer International Publishing; 2013. p. 279–85.
[13] Huang Y, Huang B, Kechadi MT. A rule-based method for customer churn prediction in telecommunication services. In: Advances in knowledge discovery and data mining. Berlin Heidelberg: Springer; 2011. p. 411–22.
[14] Faris H, Al-Shboul B, Ghatasheh N. A genetic programming based framework for churn prediction in telecommunication industry. In: Computational collective intelligence, technologies and applications. Springer International Publishing; 2014. p. 353–62.
[15] Xiao J, Teng G, He C, Zhu B. One-step classifier ensemble model for customer churn prediction with imbalanced class. In: Proceedings of the eighth international conference on management science and engineering management. Berlin Heidelberg: Springer; 2014. p. 843–54.
[16] Amin A, Rahim F, Ali I, Khan C, Anwar S. A comparison of two oversampling techniques (SMOTE vs MTDF) for handling class imbalance problem: a case study of customer churn prediction. In: New contributions in information systems and technologies. Springer International Publishing; 2015. p. 215–25.
[17] Kawale J, Pal A, Srivastava J. Churn prediction in MMORPGs: A social influence based approach, vol. 4. In: International conference on computational science and engineering. CSE'09, IEEE; 2009. p. 423–8.
[18] Lu N, Lin H, Lu J, Zhang G. A customer churn prediction model in telecom industry using boosting. IEEE Trans Indust Inform 2014;10(2):1659–65.
[19] Idris A, Khan A, Lee YS. Genetic programming and adaboosting based churn prediction for telecom. In: 2012 IEEE international conference on Systems, Man, and Cybernetics (SMC), IEEE; 2012. p. 1328–32.
[20] Xie L, Li D, Xia J. Feature selection based transfer ensemble model for customer churn prediction, vol. 2. In: 2011 International conference on system science, engineering design and manufacturing informatization (ICSEM), IEEE; 2011. p. 134–7.
[21] Idris A, Khan A. Ensemble based efficient churn prediction model for telecom. In: 2014 12th International conference on Frontiers of Information Technology (FIT), IEEE; 2014. p. 238–44.
[22] Liu J, Yang G. Research on customer churn prediction model based on IG_NN double attribute selection. In: 2010 2nd International conference on Information Science and Engineering (ICISE), IEEE; 2010. p. 5306–9.
[23] Huang Y, Huang BQ, Kechadi MT. A new filter feature selection approach for customer churn prediction in telecommunications. In: 2010 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), IEEE; 2010. p. 338–42.
[24] Shen Q, Li H, Liao Q, Zhang W, Kalilou K. Improving churn prediction in telecommunications using complementary fusion of multilayer features based on factorization and construction. In: The 26th Chinese Control and Decision Conference (2014 CCDC), IEEE; 2014. p. 2250–55
[25] Yang XS. Firefly algorithm. Nature-inspired metaheuristic algorithms 2008; 20: 79–90.
[26] Prakasam A, Savarimuthu N. Metaheuristic algorithms and probabilistic behaviour: a comprehensive analysis of Ant Colony Optimization and its variants. Artific Intell Rev 2016;45(1):97–130.
[27] Kirkpatrick S, Vecchi MP. Optimization by simmulated annealing. Science 1983;220(4598):671–80.
[28] Černý V. Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. J Optimiz Theory Applications 1985;45 (1):41–51.
[29] Morik K, Köpcke H. Analysing customer churn in insurance data–a case study. In: Knowledge discovery in databases: PKDD 2004. Berlin Heidelberg: Springer; 2004. p. 325–36.
[30] http://www.kdd.org/kdd-cup/view/kdd-cup-2009/Data.