Evolutionary Churn Prediction in Mobile Networks Using Hybrid Learning

V. Yeshwanth
Ericsson India Global Services Pvt. Ltd
Chennai, India
yeshwanth.v@ericsson.com

V. Vimal Raj
Sri. Venkateswara College of Engg,
Chennai, India
vimal1805@yahoo.co.in

M. Saravanan
Ericsson India Global Services Pvt. Ltd
Chennai, India
saravanan.r.mohan@ericsson.com

Abstract

Churn is the movement of customers from one mobile network operator to another. It is always better to retain a customer than having to find a new customer in the present competitive environment and the importance of this fact can't be stressed enough. Being more of a social phenomenon than a mathematical one the existing models fail in prediction of such a behavioral quantity. Churn prediction is valuable to the mobile operator depending on the level of accuracy of predictions. This paper presents predictive modeling of customer behavior based on the application of hybrid learning approaches for churn prediction in the mobile network. Our proposed framework deals with a better and more accurate churner prediction technique compared to the existing ones as it incorporates hybrid learning method which is a combination of tree induction system and genetic programming to derive the rules for classification based on the customer behavior. Finally using the game theory techniques we understand the community effect of churn. We calculated the predicted score which is a churn value of a mobile customer. The proposed model is used for prediction of various user defined groupings based on usage time, location and their underlying social network, thus making it a pragmatic approach which models churn on human level than a mathematical level. The post evaluation results on a real world dataset from a leading operator validate our findings.

Keywords

Churn Prediction, Data Mining, Genetic Algorithm, Game Theory, Community Effect of Churn

Introduction

In the telecom service industry customer churn prediction is a very important problem [1, 2]. With the continuous addition of new operators the market has become more competitive than before. Thus, it is important for the operator to retain an existing customer than to get a new one. But churn, being a social influence cannot be modeled completely on a person's activity alone. Every person is affected by his/her social group movements. Hence his/her social influence needs to be accounted for the analysis.

More specifically, today's researches on telecom churn are mostly concentrating on mobile user's behavior [1]. The mobile network operators are struggling to comprehend their customer problems, due to which they are facing difficulty in understanding them accurately and that timely knowledge translates into improved business performances. The aim of this study is to show that the data mining methods based on evolutionary algorithms could be successfully applied to understand the mobile user's behavior in the problem of churn prediction. Existing models of churn prediction follow regression models or normal classification/prediction techniques [3]. The effectiveness of such results remains under question for practical implementation. Considering the limited human resources at hand with the operator for such activities, it is important to get the proper subset of *potential churners* to target. By identifying the core set of possible churners we can deduce the factors common to their usage pattern. This turns out to be a more than a useful spin off. Due to the complexity in prediction and difficulty in building a good single model that shows high performances, many researchers have employed a hybrid model approach in solving the problem [4]. We have used two different techniques namely, *tree induction* and *genetic algorithm* in early stages of predictive modeling of this study.

In artificial intelligence, Genetic Programming (GP) is an evolutionary algorithm-based methodology inspired by biological evolution to find computer programs that perform a user-defined task [5]. It is a machine learning technique used to optimize a population of computer programs according to a fitness landscape determined by a program's ability to perform a given computational task. For the non genetic standard approach we choose the Tree Induction algorithm [6]. The tree induction algorithm is used specifically because of its output format which is in a human readable form of if-then statements and its proven effectiveness on the kind of domain data we are going to use. Test results corroborate the above statement. We have seen that depending upon the dataset the usefulness of an algorithm can vary and that too in a domain like telecom we cannot underestimate the variety of data coming in. Hence with the use of two different techniques for rule discovery, we can improve the overall efficiency. In order to exhibit the usage of this model we apply it to the mobile telecom domain related to churn prediction problem, as the abstraction of churners at different levels is predictable and it is helpful for the analysis.

With regard to network analysis, existing models which take into account the neighboring nodes effects are bound by traditional centrality metrics which consider only the relationship between individual nodes and the rest of the network [7]. It is a limitation in some applications because the dynamics of combinations of nodes are completely ignored. Game theoretic centrality [8] has been proposed as a part of the framework in this study that would address the above limitation. That is, one might want to define centrality of a node n as of "the average extent by which the addition of n to an arbitrary set of nodes S reduces the distance between S and other nodes in the network". This is done by calculating the Shapley Value [9] of each individual node in an operator network. The added advantage is that it is computationally efficient.

Community effects on any specific task are difficult to calculate considering the size of the data [7]. Trivially we see that one person is affected by his neighbor's behavior. But when we take into the overall effect on a single node, we have to consider which combination of nodes affect its behavior. So when we take into effect every such combination it becomes an exponential time problem. To solve this problem we used a game theoretic approach similar to the problem of calculating Shapley Values [9], to find the predicted values of "community effect" of churn. To our best knowledge this is the first work considering the application of game theoretic techniques in understanding the community effect of churn in the mobile social networks by considering time effective solution. Now that we have pragmatic real world mapped values for churn, we can set various thresholds according to the operators needs to filter out the required core set of potential churners. Thus our framework captures the best of both the learning methods and network centrality measures in predicting *potential churners* in the mobile network.

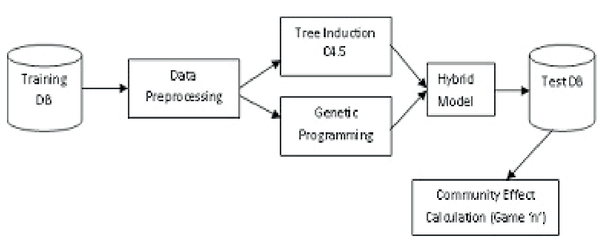

Figure 1. The diagram represents the flow of events in hybrid model.

In this paper, we have proposed a framework to effectively solve the problem of churn prediction in a mobile network. The flows of events that happen in our framework are shown in Fig 1.

Data Pre-processing

In a typical billing system of a mobile operator, for every operation performed by the customer, which varies from Voice, SMS usage to GPRS usage, individual records are generated and stored. These records are referred to as Call Detail Records (CDRs). We have considered around 1.2 million subscribers CDRs from the largest telecom operator in a developing country for our analysis. The time duration is between 1st July and 1st September 2009. From the available users dataset 0.7 million is taken for training and the remaining as a test set. The required fields from CDRs are aggregated over time intervals and possible churners are identified. In earlier churn prediction models the fields were aggregated over entire time period. The disadvantages were that, a new user joined during the last week of the month is classified as churner due to his low usage and also it is difficult to generate any patterns from those data, as differences in usage pattern cannot be derived.

These issues are analyzed thoroughly in our approach. For example, if we have data for a month, we can split into three intervals. Now having split calls into three equal intervals, if the user hasn't made any calls in the last interval (or) the last two intervals he can be classified as a possible churner. This would imply that he has ceased to use the services offered by the operator in the last recorded days. In other case, if he is a new user (who has joined in the end of month) the number of calls made by him in the first two intervals might be zero. In other models he will be classified as a churner but in our model that user will not be considered as a churner. Moreover, we are classifying possible churners on different types as it is not viable just to say if a person is a "possible churner "or not.

"If a mobile subscriber does not have any transaction records in the following month of analysis, he/she is considered to be a churner".

Hybrid Learning Method

When we use more than one technique to classify/predict values over a data set, in certain cases, there is a possibility that we will get a better result in one of the methods and not the others. In those cases we have a choice as to which one we should consider for the result. Hence we need to design a mechanism to determine which of the two results must be considered and to what extent the result is applicable to the mobile telecom scenario. This stage comprises of two techniques, the tree induction C4.5 technique as well as the genetic programming [6]. In our model, Genetic Algorithm acts as a reinforcement technique to C4.5. For an initial framework building, we have chosen two simple classification techniques like tree induction and genetic programming which have already been applied to many mobile network related studies.

Other techniques can be easily adapted to this framework for further improvement.

A. Classification using Tree Induction

We start the process by applying the C4.5 algorithm on the pre processed dataset. We emphasize on the pre-processing because of the split-aggregation done, which will aid in better patterns or paths to process for the tree formation. Post evaluation we have seen that this does indeed give very good results as the classification label (possible churner label) seems to have a correlation with the periodic usage of customers.

Decision trees are commonly used for gaining information for the purpose of decision-making. A decision tree starts with a root node on which further actions are taken. From this node, it is split to each node recursively according to the decision tree learning algorithm. The final result is a decision tree in which each branch represents a possible scenario of decisions and its outcome. Decision tree learning algorithm is suitable when the target function has discrete output values. It can easily deal with instance which is assigned to a boolean decision, such as 'true' and 'false' or 'p(positive)' and 'n(negative)'. Moreover, it is possible to extend targets to real valued outputs. The training data on which the tree is constructed may contain errors. This can be dealt with pruning techniques [6].

One of the most common techniques used in information theory is the Information Gain which is used to find the difference between the original information requirement i.e. the expected information to classify an instance in dataset D (collection of instances required for classification) and the new information requirement i.e. the information needed after partitioning in order to arrive at an exact classification (Impure partitions) based on the set of formulas given below:

$$\mathrm{Info}(D) = {}_{i=1..m}\Sigma p_i \log_2(p_i) \quad (1)$$

$$\mathrm{Info}_A(D) = {}_{j=1..v}\Sigma (|D_i| / |D|) \, \mathrm{Info}(D) \quad (2)$$

$$\mathrm{Gain}(A) = \mathrm{Info}(D) - \mathrm{Info}_A(D) \quad (3)$$

where pi is the probability that an arbitrary tuple in D belongs to class Ci. D denotes the partition. Let m be the number of distinct classes possible for the specific class label. Attributes with higher information gain are selected as relevant attributes. C4.5 works well on the real training data which is revealed to have structural properties compared to the random data generated for early testing (using J48 implementation in WEKA Data mining Toolkit [10]).

B. Classification using Genetic Programming

Next we apply the genetic programming technique to the pre-processed dataset to generate a new set of classification rules and class labels. GP is evolved as set of computer programs traditionally represented in memory as in the form of tree structures [5]. In general, trees can easily be evaluated in a recursive manner. Every tree node has an operator function and every terminal node has an operand, making mathematical expressions easy to evolve and evaluate. The main operators used in evolutionary algorithms such as GP are, crossover and mutation. Crossover is applied to an individual by simply switching one of its nodes with another node from another individual in the population. With a tree-based representation, replacing a node means replacing the whole branch. This adds greater effectiveness to the crossover operator [5]. The expressions resulting from crossover are very much different from their initial parents. Mutation affects an individual in the population. It can replace a whole node in the selected individual, or it can replace just the node's information. The reason for using this genetic algorithm is because some algorithms might get stuck in a local maxima or minima, but this algorithm gives a constant accuracy rate on a set of similar data as seen from initial set of experiments conducted. In lieu of the number of attributes taken and practical time needed for completion, the population size was taken to be 100, the maximum generations was set to 20 and the max depth of the program trees were set to 5. The fitness function used is dependent upon the number of instances correctly classified under a class label for the rule in iteration.

C. Hybrid Learning Approach

Now we describe a mechanism to determine which of the 2 results (C4.5 or GP) must be taken and to what extent the result is applicable to the scenario. Both C4.5 and genetic programming will generate an output, describing whether a person is a *possible churner* or not. But the model used for classification is dependent on the model used for prediction. Hence we represent the *%Churn* value of every customer as follows

$$\%\mathrm{Churn} = (OP_{C4.5} + OP_{GA}) / 2 \quad (4)$$

$$OP_{C4.5} = \mathrm{output(C4.5)*Accuracy\ of\ C4.5} \quad (5)$$

$$OP_{GA} = \mathrm{output(GA)*Accuracy\ of\ GA} \quad (6)$$

%Churn can be defined as *the accuracy of the prediction that a person will churn out with* and OP denotes the numerical measure derived from the outputs of the individual classifiers. We shall see the evaluation results in later part of this paper, as it makes for better understanding of the hybrid learning approach.

Community Churn Effect

Churn, as mentioned before is a social phenomenon; hence a mathematical treatment alone is not enough to understand it. The churn value of a person is influenced not only by his own usage patterns but also his neighbors' usage activities. The reason is that as humans we tend to be highly influenced by the actions of people surrounding us. Hence it is important to include the neighbors' effect in the method. This can be done on the basis of energy dissipation model.

In order to calculate the effects of propagation we follow a game theoretic model to calculate the same. Using a Monte Carlo Approach for the calculation of community effects of Churn would leave us with an execution time constraint [9]. While it is easy to model the dissipation of energy from one node to its neighbors, it is difficult to follow the same when it is going to be applied to every node. Hence we devise various games to achieve our result in polynomial time. This problem is similar to the calculation of Shapley Values [9] which can be calculated in polynomial time in a co-operative network game. We also calculated the Shapley Values of the nodes in our study. The calculation is done for filtering out the highly connected nodes so as to see the relation between their Shapley Values and the Community Churn propagation results. The calculation of Shapley value in our study is a prediction that one would ideally like to assign a centrality to a node *n* based on the consequences of failure of every arbitrary combination of nodes containing *n*, rather than just failure of the single node.

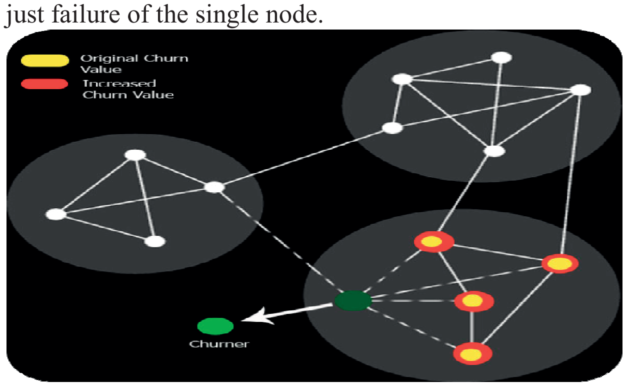

Figure 2 Churn value of neighbors increasing due to Churner

Fig 2 shows the effect of increase in the individual churn values when a person connected to them churns out of the operator network. If the person churned out is connected strongly to his neighbors then the churn propagation to those around him would also be high. First we use an unweighted network to simulate the community based approach. This by far, has the least computation involved and does not take into consideration how much effect the connectivity between users plays a part in community propagation. Before we get into the algorithms, an important calculated quantity needs to be discussed in our model. It is the *CommChurn* variable associated with every node in the network. This is the variable attached to every node, *which describes how "influenced" or "liable to churn" the node is due to the effect of his/her neighbor's behavioral patterns*.

For the proposed game we group all the data by the <*caller,callee*> pairs to construct an adjacency list matrix in memory for the network. We apply this weighted graph method as input to our game to understand the churn propagation. In the weighted network, the weights we have taken are frequency of voice calls, SMS and cost, duration of calls between two users. Every Node's %Churn value or the Hybrid[i] is taken as input to use as initial energy. We don't go into analyzing the un-weighted network, which are an indicator of the strength of connectivity as it makes it difficult to study the effects of the churn propagation in the network. This is also loaded into memory with the community-based approach designed for the proposed game.

Game:
Algorithm : Computing Community Churn Values for game

Input: Undirected weighted network
Output: Community Churn Value of all nodes in Network w.r.to Game 2
foreach *node v in Network* do
    CommChurn[v]=Hybrid[v]
    foreach *neighbor u of v* do
        CommChurn[v]+=f(u,v)*Hybrid[u]
    end
end

f(u,v)=f(no. of calls, SMS and duration, cost between u and v)

Thus a realistic community churn value is obtained for the users of the network. Now let us take a trivial example to explain the community effort of churn.

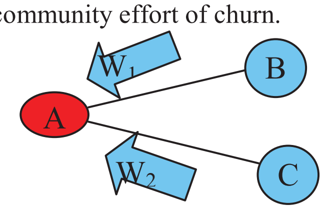

Figure 3. Effects of neighboring nodes churn values on CommChurn of a node

In Fig 3, let A, B, C be three nodes of the network. For simplicity, we see the propagation of churn to a node from its neighboring nodes. The churn values associated with the nodes are *CommChurn*(A), *CommChurn*(B) and *CommChurn*(C) respectively from the hybrid method stages. So the weights of the connection are indicated by W_1 and W_2 in the above figure. So for the calculation of *CommChurn* (A), we will get through our algorithm

$$\text{CommChurn(A)+=CommChurn(B)*}W_1\text{+CommChurn(C)*}W_2 \qquad (7)$$

Experimental Evaluation

A. Evaluation of Hybrid learning Approach

We observe from experimental results that when one method identifies a person as a possible Churner and another does not, the hybrid method output i.e., *%Churn* is pointed towards the method with higher accuracy. This is particularly useful when one model is giving a lesser accuracy compared to other one. But with GA, constant accuracy can be guaranteed. Hence we get a continuous value as output instead of discrete values. This is more useful than it appears as operators have limited resources to handle churners, now by setting a threshold according to the needs; we can drill up or down the number of *possible churners* according to the *%Churn* value. This value can be the initial energy for the energy dissipation model we want to apply for the community churn propagation.

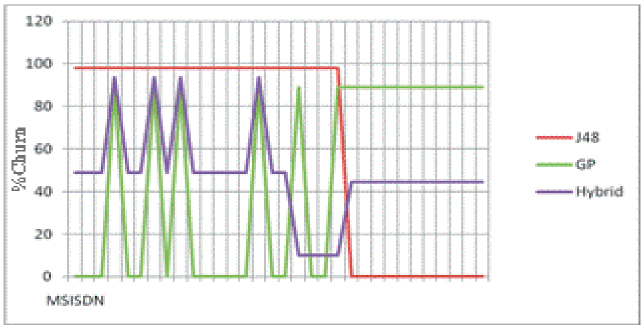

Figure 4. Comparison of the 3 methods (MSISDN numbers blanked for privacy)

Fig 4 shows the comparison between the three methods used. The X axis represents the MSISDN numbers of the subscribers while the y axis represents the %Churn value. Hybrid Model demonstrates better accuracy than the J48 and GP models. The above graph represents the difference in results between the 3 models for a randomly choosen 200 churners having highest Shapley values [9]. The shapley value of a node is a function of the degrees of its neighbors taken in an iterative manner. Moreover, we observed that the number of churners in reality exceeds 50,000 with a single operator, so we have taken a subset of the churners to show the comparison. We see that the hybrid model generates a value between the two values of the proposed models. If one model picks up a churner and the other doesn't, the relevance of the result is dependant on the accuracy of the models.

B. Evaluation of Community Churn Effect

When run on the call usage records of all the subscribers we get an average *CommChurn* value of 1.10533 and a maximum of 679 (Table 1)). Note that these are absolute values representing the extent to which an individual subscriber is connected and how likely he is to churn, over another subscriber in the same group. These values are not normalized to show the absolute difference in behavior of subscribers. Table 1 shows the statistics regarding the values of *CommChurn*, whereas Table 2 shows the range distribution of the *CommChurn* values.

Table 1. *CommChurn* Statistics

MAX *CommChurn* value	679
MIN *CommChurn* value	0.4497
AVG *CommChurn* value	1.10533

Table 2. *CommChurn* Distribution

MIN - 1.5 MIN	18017
1.5 MIN - 0.5AVG	36559
0.5 AVG - AVG	10059
AVG - MAX	8138

An interesting point we have to notice is that most of the population in Table 2 lies between the Average (AVG) *CommChurn* value and the Minimum (MIN) *CommChurn* value. These indicate the fringe cases that exhibit "near" churner behavior but are actually not. The setting of a threshold enables us to narrow down our target, which is seen evidently from the results provided in Table 3.

Table 3. Summary of Results from Community Churn Propagation

Threshold	1.5*MIN	0.5*AVG
Number of Churners shown using *Commchurn*	54756	18576
Number of Churners actually churned out	37246	13409
False Positives	**17510(31.97%)**	**5176(27.86%)**
Accuracy (Percentage)	**68.02**	**72.18**

Table 3 indicates the evaluation of our predictions after the churn propagation on the communities. We compared the predicted churners over different thresholds to the actual churner list, which was obtained from the following month data records. We have calculated the *Accuracy* and the *False Positives*, which would give a clear indicator with regard to effectiveness of the proposed framework. Accuracy is defined as the ratio between the number of predicted churners actually churned out and the number of churners shown using Commchurn (for a specified threshold). False Positive is defined as the ratio between number of churners shown using Commchurn (for a specified threshold) and the number of predicted churners actually churned out (from the latter set).

We understand that by properly adjusting the threshold value according to the distribution we are able to increase the accuracy of our predictions to a good figure. We have effectively narrowed down the set of churners and have gained on accuracy of the predictions by virtue of community churn propagation. The results shown here thoroughly vindicate our stand with respect to the effectiveness of narrowing down the search for possible churners. The evaluation result given in Table 3 indicates the confirmed churner prediction. With more subscriber data for subsequent months we can see if the high *CommChurn* subscribers continue with their low usage behaviour. As we are predicting only the "possible" churners that characteristic will be well reflected with that data. These results show that we obtain a very good percentage of actual churners in the predicted list along with users whose usage patterns cause a concern regarding their future status in the network.

Fig 5 shows the result of the community effect of churn on the data we have taken. It is not possible to show the effect on the entire data set due to size and visualization constraints. Hence we have taken the propagation effect on a set of nodes labeled A to H. The values in the brackets denote the initial *CommChurn* value and the final *CommChurn* value obtained after the propagation. Take the case of A, whose initial *CommChurn* value is as high as 0.94 initially, after propagation we see that the value increases a lot considering the very strong ties it has with its neighboring nodes (only 3 neighbors have been represented here). A is represented in red specifically as after checking the database, A has been found to be an actual churner. But notice that C and F who have been classified as non churners initially do not show a change in their values due to their strong ties with other nodes with similar non-churner characteristics. A's final *CommChurn* value has increased so much in spite of being associated with a node like C, because of its very high Shapley value of 3.9449 (indicating connectivity strength) and the *CommChurn* values of all its neighbors. Hence we see that a person with such high *CommChurn* value has a very high probability of churning out of the operator network.

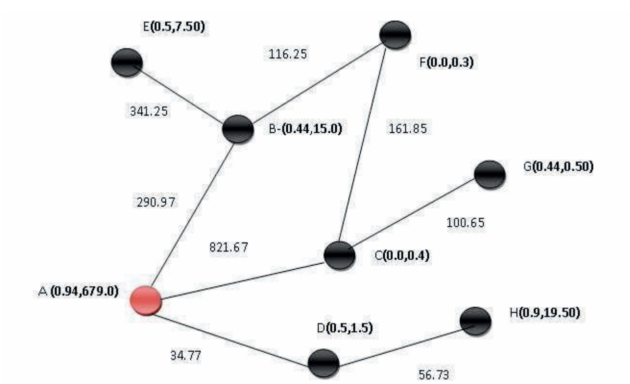

Figure 5. **Snapshot of Community of Users**

Conclusion

The proposed framework uses an evolutionary approach combined with normal tree induction and GP methodss to give a pragmatic churner model. It uses game theory concepts to take into account the effects of neighboring nodes, which contribute to the practical churn values of a customer. The games used to do this are designed with different levels of accuracy, with execution time and necessity in mind. Hence we get the best of induction tree and genetic techniques and make it practical using game theory. Further the more important rules in if-then format can be obtained from the reinforced results in order to optimize the network parameters like coverage, recharge methods, plans etc. We have thus obtained a model that can be depended upon for accuracy of its prediction as well as taking into account the social influence of neighbors on an individual by virtue of the strength of connection existing between them, which in concept drew inspiration from the heat dissipation theory, and has proved to be more than applicable in this context. Another point we would like to highlight here is that the data which is aggregated over time intervals has a scope for proper pattern mining. It appeals as the logical choice for analysis in the time interval aggregated form due to the fact that we are able to trace out the user behavior at a more granular level.

References

[1] Chih-Ping Wei, and I-Tang Chiu. 2002. Turning telecommunications call details to churn prediction: a data mining approach. *Expert System Application,* 23(2), 103-112.

[2] Koustuv Dasgupta, Rahul Singh, Balaji Viswanathan, Dipanjan Chakraborty, Sougata Mukherjea, Amit Anil Nanavati, and Anupam Joshi. 2008. Social ties and their relevance to churn in mobile telecom networks. EDBT 2008: *Proceedings of the 11th international conference on Extending database technology*, 668-677, New York, USA. ACM

[3] Hadden, J., Tiwari, A., Roy, R., and Ruta, D. 2006. Churn prediction: Does technology matter? *International Journal of Intelligent Systems and Technologies, 1* (2), 104--110.

[4] Jae Sik Lee and Jin Chun Lee. 2006. Customer churn prediction by hybrid model, *Advanced Data Mining and Applications*, 4093, 959-966, Springer Berlin.

[5] Wai-Ho Au Chan, K.C.C and Xin Yao. 2003. A novel evolutionary data mining algorithm with applications to churn prediction. *IEEE Transactions on Evolutionary Computation*, 7, .532-545.

[6] Bala, J, Huang, H. Vafaie, K. DeJong and H.Wechsler. 1995. Hybrid Learning Using Genetic Algorithms and Decision Trees for Pattern Classification, *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), August 19-25, 1995, Montreal, Canada*

[7] Archaux C. And Laanaya H. and Martin A. and Khenchaf A. 2004. An SVM based Churn Detector in Prepaid Mobile Telephony", *Proceedings of the International Conference on Information & Communication Technologies: from Theory to Applications (ICTTA*), Damascus, Syria, 19-23 (April 2004).

[8] Aadithya, K. V. and Ravindran, B. 2010. Game Theoretic Network Centrality: Exact Formulas and Efficient Algorithms". *Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010).*

[9] N. Rama Suri and Y. Narahari. 2008. Determining the top-k nodes in social networks using the Shapley value. In AAMAS '08: *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, 1509–1512, Richland, SC, 2008.

[10] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. *The WEKA Data Mining Software: An Update; SIGKDD Explorations*, 11. 1.