International Journal of Database Theory and Application
Vol.9, No.8 (2016), pp.211-232
http://dx.doi.org/10.14257/ijdta.2016.9.8.21

An Ensemble Approach for Efficient Churn Prediction in Telecom Industry

Pretam Jayaswal[+], Bakshi Rohit Prasad[*], Divya Tomar[!], and Sonali Agarwal[#]

Indian Institute of Information Technology Allahabad, India
{[+]pretamjayaswal, [*]rohit.cs12, [!]divyatomar26}@gmail.com, [#]sonali@iiita.ac.in

Abstract

The rise of globalization and market liberalization are changing the face of market competitiveness significantly. The appearance of modern technology in business processes has intensified the competition and put forth new challenges for service providing companies. To cope up with changing scenarios, companies are shifting their attention on retaining the existing customers rather hiring new ones. This is more cost effective and requires lesser resource as well. The phenomenon of abandoning the company by a customer is known as churn and in this context, anticipating the customer's intention to churn is called churn prediction. Data Mining and machine learning techniques, as applied to customer behavior and usage information, can assist the churn management processes. This paper used customer usage and related information from a telecom service provider to analyze churn in telecom industry. The decision trees and its ensembles, Random Forest and Gradient Boosted trees are used as underlying statistical machine learning models for building the binary churn classifier. The implementation part has been done using apache spark which is state of the art unified data analysis framework for machine learning and data mining. In order to achieve better and efficient results, the grid based hyper-parameter optimization is applied.

***Keywords**: Churn Prediction, Random Forest, Gradient Boosted trees*

1. Introduction

The Customer acquisition and retention are the two basic and important concerns of any business. Some business organizations give them equal importance while some try to keep a trade-off between them. The focus on acquisition and retention depends on current marketing scenarios, business type and business requirements. Generally the new entrants in market focuses and invest much on customer acquisition, while the old and mature one focuses to retain its current customer base intact in order to create new business opportunities with them and encourage cross-selling [1]. With the globalization and emergence of business solutions like CRM (Customer Relationship Management) advance data storage technologies empowered the business organizations with better insights of their customer requirements, business trends and other information related to sells and purchase. On the other hand, the same medium also empowered the customers to compare and choose the best available products/services for them. The boom in E-commerce sector is also encouraging the customers to not stick with a single company, by offering their customers a diverse range of products, and it is all without much hassle and within a few clicks away. The customer empowerment is also raising new challenges for the companies. Customers are now more sensitive towards services and products offered to them and any mishap leads to customer attrition. Since the cost of acquiring new customer is much higher than retaining old ones, many companies investing more in customer behavior analysis and equipped themselves with state of the art methodologies to ensure customer retention [2].

The phenomenon of customer attrition, currently or in future is known as 'Churn'. The retention centric practices which are performed to control churn by stakeholders is called 'Churn management' and the purpose of churn management is to keep profitable customers engaged with company. The concept of Churn is observed in all industries, but some subscription based business sectors like Internet service providers, Digital TV, Telecom, *etc.* suffers much from it. During the decade of 1990-2000 the business managers and researchers were exploring the solutions which can help them to minimize churn and maximize their profits by utilizing the customer data that was available with them and find out precious, hidden business patterns with data. There were statistical methodologies like data mining and machine learning which were used to solve similar type of problems in other domains like medical, advertisement, spam detection, *etc.* [3-10]. The various approaches in Data Mining and Machine Learning techniques used and applied over customer data to get insights and patterns which help business managers in better churn management. The Churn with respect to Telecom industry defined as the loss of customers to another company/organization. The Wireless Telecom industry is one of the major industries which are suffering from higher churn rate due to easy portability options, service issues, lack of service contracts and a number of optional service providers in market. The industry was initially focused on rapid customer acquisition but gradually, after 4 to 5 years the market became saturated and since the 'customer-pool' is limited, the operators have to switch and focus on customer retention rather than acquisition because of lower retention costs. Earlier studies [11] outlines that the average estimated monthly churn rate for mobile Telecom companies lie in range 1.9% to 2.2% and if we compound it annually the customer churn will be 25% to 30%. These figures suggest that churn prediction arises as important business intelligence (BI) concern for the Telecom industry and development of efficient churn prediction application is pivotal. As shown in Figure 1, a survey report for two successive years by E-consultancy [12] says that in year 2013, 31% of the participating companies invest and focus more on acquisition, 24% focuses on retention and 45% gave them equal importance. For 2014, the figures changed slightly and the number of companies which focus on acquisition increases by 3% *i.e.* 34% while 18% focus on retention and 47% give them equal importance.

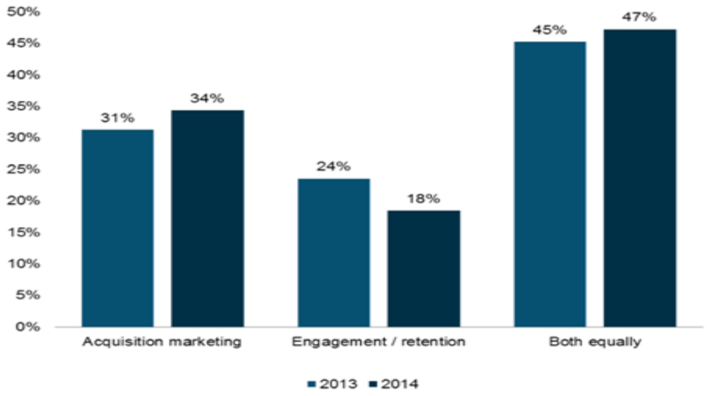


Figure 1. Survey for Investments Trends in Customer Acquisition and Retention

Since wireless telecom industry is major sufferer of customer churn, with 25-30% annual churn rate, therefore, it is essential for telecom industries to have a reliable and accurate churn prediction model that should recognize the customers which will going to churn in close future. The higher churn rates in telecom industry get attention from many research groups and studies. Each study suggests its own approach to handle the problem.

Many hackathons and open competitions are organized to get new and innovative way to solve the problem. The prestigious KDD Cup 2009 was dedicated to churn prediction in telecom industry. The Telecom industry observes three types of churn:

- Active or voluntary churn- customer discontinues contract deliberately and subscribes to another service provider.
- Incidental churn- the customer terminates his service contract with his service provider but does not switch to any competitor. Reasons may be financial, geographical location, personal, *etc.*
- Passive or non-voluntary churn- the service provider itself discontinued the contract.

We are only interested in active or voluntary churn because this is the only churn type we can analyze and predict mathematically, the other two types of churn are hard to predict and generally incidental. In general, the churn prediction problems can be identified as classification problems where the customers are classified into various classes or specifically in binary classes that distinguish them in churners and non-churners. This paper uses tree based machine learning techniques especially 'ensembles' to build an accurate and efficient prediction model for predicting customer churn in Telecom industry.

2. Related Work

Machine learning and data mining techniques are evolved widely since they revived as mentioned in Figure 2. The revival period of machine learning paradigm is assumed to be around mid-1980, when the first learning model based on trees was proposed by John Ross Quinlan which was recognized by the Statistical and Artificial Intelligence research community. Another neural network based statistical model was also invented at same time which were capable of expressing any problem or function as neural networks, inspired by biological nervous system especially by human brain.

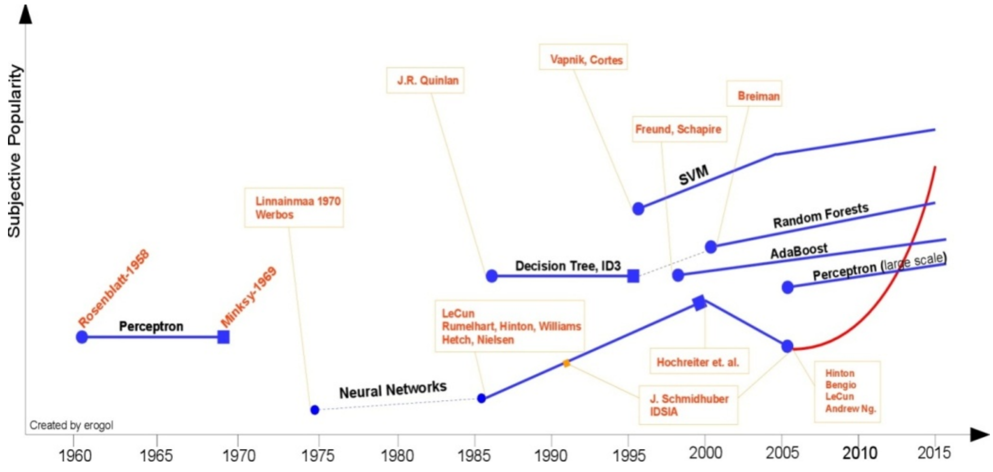


Figure 2. Data Mining Time Line

Other major works were logistic regression and support vector machines developed during mid-90's and are still evolving. Decision trees are still so popular and used widely because of their better performance and versatility. Decision trees are used extensively in field of biomedical, financial analysis, fraud detection, remote sensing, text processing, *etc.* The reasons for such wide acceptance of decision trees are explicability, non-linearity, high level visualization and their faster execution. Decision trees are quite popular in financial and business domain for applications like market analysis, advertisement and customer behavior prediction.

While exploring the literature work for churn analysis we found many related research works done in past and the study of these research works gave us useful insights about customer churn prediction problem and also provided a theoretical base to start from. Several attempts have been made to predict customer churn in various domains like banking, telecom, on line gaming, advertisement *etc.*, Researchers used various established data mining techniques like support vector machine, neural networks, clustering, Regression, decision trees and hybrid methods and their ensembles in their work [13-18] for accurate prediction of active churn. Since, the ratio of churners to non-churners in any business is of very small, so the class distributions are generally highly skewed towards non-churn class which leads to class imbalance problem. Some researchers handles the class imbalance problem in data pre-processing phase and one of the most popular technique used by them was sampling [19] which involves sub-tasks of oversampling and under-sampling. However, contrary to traditional class imbalance challenge in churn prediction, a recent research [20] suggests that oversampling or under-sampling of the churn data does not affect the performance of prediction model.

Wei and Chiu [21], in 2002, were the first who applied the data mining approaches in Telecom industry and attempted to identify the customers who were going to the leave the organization (or going to churn) in near future. The study was based on the contractual information between the customer and service provider and the usage pattern of the customers especially the calling patterns of the customer was considered as influencing factor. The dataset used for the study was highly skewed (98.5-98% non-churners and rest were churners) which was addressed by using multiple-classifier classification technique. For building model, decision trees were applied whereas for testing purpose, multiple learning-testing approach was used to avoid the misleading results and the method used was 10 fold cross validation. The evaluation of prediction model was done using fall and miss alarm rates.

Ho *et al.* [22] in 2003, extended the idea of mere churn prediction to churn likelihood. The research emphasized the importance of likelihood of subscribers and proposed a novel data mining algorithm called data mining by evolutionary learning (DMEL) which was based on genetic algorithms (GAs). DMEL used probabilistic random technique for initial data population thus followed non-random approach. The other striking features of DEML were objective interestingness measurement, estimation of likelihood of predictions and effective handling of missing values. DEML was capable of finding both negative and positive relationship among features for predictive modeling The DEML was tested over various types of datasets and it outperformed other established techniques like SCS, GABL and C4.5.

Hadden *et al.* [23] reviewed some of the most popular contemporary churn prediction and management technologies that were being used for development of churn management platform. They focus was only on the customer base which was worth for retention and also cost effective from business point of view. They also proposed a churn management framework as depicted in Figure 3. This framework had five stages. The first step was selection of best data; the second step focused data semantics that means knowing the data, relationship between various data objects and properties of data. The third step is selection of the best and relevant attributes for prediction and it also helps in removing noisy and non-informative data from the data selected. The remaining stages are model development and validation, the former was done using different technologies like Regression, Markov model, cluster analysis *etc.* and the later was done for verifying the developed model and the approaches used were cross-fold validation [24] and validation was performed using a separate dataset.

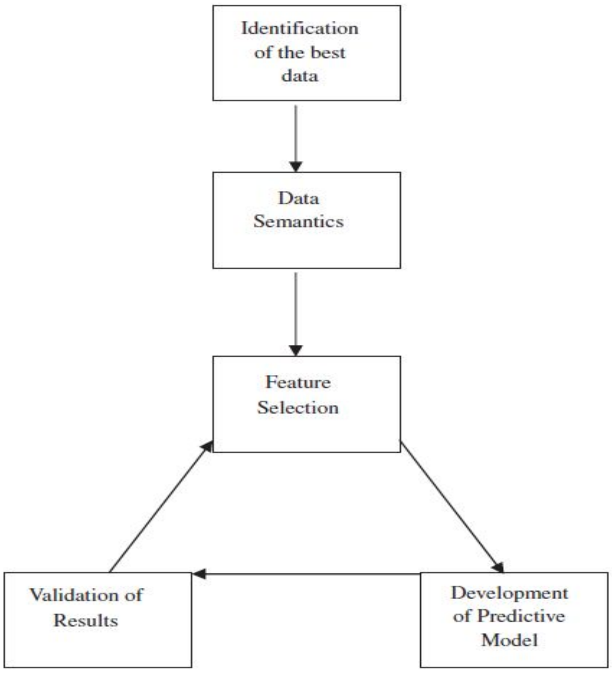

Identification of the best data
Data Semantics
Feature Selection
Validation of Results
Development of Predictive Model

Figure 3. Churn Management Frame Work

Another Instance when need of Churn prediction arose was in Taiwan when it openen telecom market for external players in 2007. Because of the limited customer pool, service providers realized that the key for their survival lies in customer retention rather than customer acquisition. Another study by Mattersion *et al.* [25] also supports the customer retention approach for survival in telecom industry.

Research efforts mentioned till now specifies attempts based on calling and usage patterns. Hung *et al.* [26] considered several other important churn influencing parameters like billing information, payment history, customer demography, call detail records, *etc.* for their study of churn management. The selection of such 40 variables using exploratory data analysis (EDA) was achieved and the variable significance was examined using Z-test. Two different model creation approaches were used, first was decision trees using k-means segmentation and another approach was back propagation neural networks. The training model used was input-hidden-output type as shown in Figure 4.

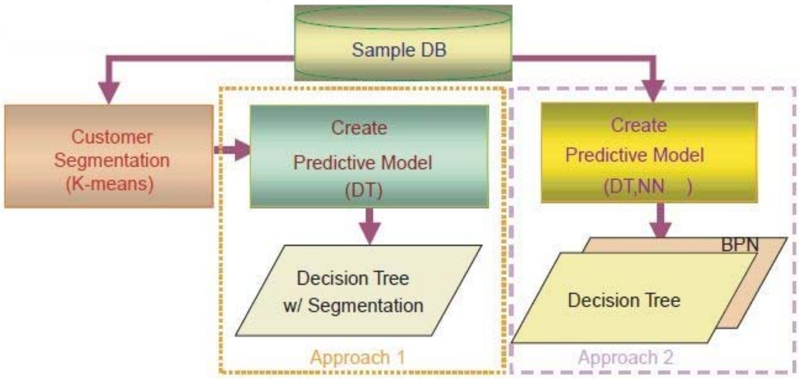

Sample DB
Customer Segmentation (K-means)
Create Predictive Model (DT)
Decision Tree w/ Segmentation
Approach 1
Create Predictive Model (DT,NN)
BPN
Decision Tree
Approach 2

Figure 4. Model Creation Process

The study concluded that decision trees performed better without segmentation whereas back-propagation neural networks performed better than decision trees in terms of capture rate and hit ratio. Till 2006, the researchers were focusing on traditional techniques and algorithms to handle churn prediction problems. In 2006, Lemmans & Croux [27] applied the ensemble approach for predicting churn using Bagging and Boosting approaches with classification tree. The classification was done by applying the equation 1 and equation 2.

$$C_{bag}(x) = sign\left[\hat{\hat{f}}_{bagging}(x) - \tau_B\right] \tag{1}$$

where $C_{bag}(x) \in \{-1,1\}$

τ_B is the cut-off value and will be zero when proportional calibration sample found in bootstrap sample, $\hat{\hat{f}}_{bagging}(x)$ is a aggregation function which aggregates the score of all constructed bootstrap samples B, the aggregation function is given as:

$$\hat{f}_{bagging}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}_b(x) \tag{2}$$

$\hat{f}_b(x)$ is the individual score function of individual boot strap samples and b=1, 2, 3, ..., B. The performance of prediction model was assessed using error rate, top-decile lift, Gini coefficient. The study concluded that Bagging approach is most competitive when compared to boosting and binary logic models so the ensembles performed better than any traditional approaches.

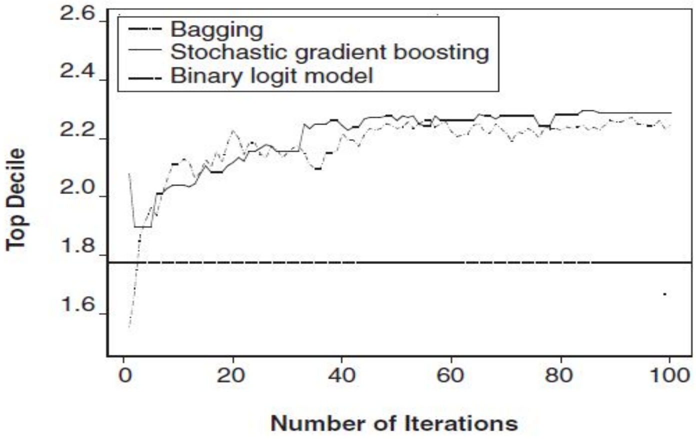


Figure 5. Performance Comparison

The study by Lemmans and Croux invoked many other researchers to explore and try ensemble approaches for classification and regression problems of machine learning and data mining. One of the researchers, Jinbo, Xiu and Wenhuang [28] used another boosting approach AdaBoost to predict the customer churn. Jinbo *et al.* used three different boosting schemes: Modest AdaBoost, Real AdaBoost and Gentle AdaBoost. They performed well in predicting the customer churn and also estimate the likelihood with predictions as well as gave explicit indication of classification rules. Study by Jinbo *et al.* concluded that the Real and Gentle AdaBoost schemes performed good with highly skewed and unbalanced dataset, While the performance of Modest AdaBoost scheme was not up to mark with unbalanced dataset sample but the performance of Modest scheme can be improved using balanced sampling schemes. Tsai *et al.* [29] in 2009, applied hybrid neural network model which included two models; artificial neural networks (ANN) and self-organizing maps (SOM). The hybrid models used are artificial neural networks associated with self-organizing maps (ANN+SOM) and artificial neural networks associated with itself (ANN+ANN). The first hybrid model performed the task of data filtering and reduction, in which irrelevant data is filtered out. The second hybrid model used the output of first model as representative data and creates the prediction model. The evaluation results showed that hybrid models perform better than generic and baseline neural network models in terms of prediction accuracy.

3. Proposed Methodology

The objective of this study was to use an efficient approach for churn prediction in Telecom Industry. There were many learning methodologies for classifier modeling but ensemble was selected as final classifier build-up approach because of the following reasons:

- **Computational Reasons:** The generic base classifiers provides good solutions but often they fail to provide best solutions because they generally follow greedy approaches and stick with the local minima of problem domain rather than global minima which provides best solution. By applying merging operation over multiple classifier it can be assumed that majority of the learners will identify the global minima or the area closer to global minima.
- **Statistical Reason:** In case of multiple classifiers, different representations of learning algorithms are obtained. The different representations have similar properties with slight variation, this variation captures all the aspects of problem and by merging the results we can get the big picture of problem space.
- **Representational Reason:** Ensemble involves different types of learners which are capable of capturing different hypothesis. The each hypothesis represents a unique aspect of the input sample and by combining the entire hypothesis we can model a classifier which includes all the aspects of given data.

3.1. Random Forests

A random forest is an implementation of bagging paradigm of ensemble learning and the first ensemble that was used to build churn prediction model in this study. The "Random Forests" was developed by Leo Bremen [30] and combines the idea of "Bagging" and the random selection of features. The Random forests belong to the family of classifiers which populates a forest of decision trees as shown in Figure 6.

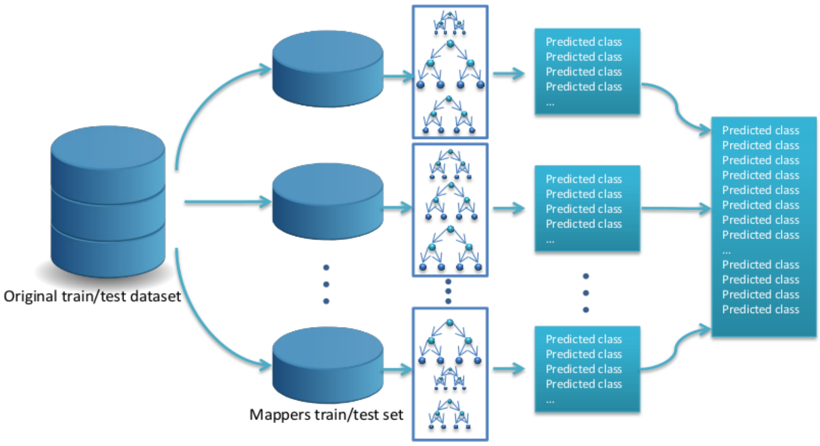


Figure 6. Random Forest

The decision trees in the forest were inducted using random bootstrapped replicate of data. The selection of base classifier depends upon variance and biasing nature of the classifier, In general, the classifiers with high variance and low bias are preferred. In this study, the decision tree is used as a base classifier for Random forests. The reason for selecting decision tree was the low bias and high variance properties of decision trees. Random forests grow and use a forest or collection of classification trees and each tree will be trained independently and can be done in parallel fashion. The Random forest algorithm introduces randomness in training process to increase diversity between the individual trees, the randomness is injected by:

- By sub sampling the original dataset with the replacement at each iteration.
- Using random subsets of features for node splitting in each tree.

After growing the forest, we can classify new objects by majority voting. The pseudo code for step by step procedure to work with random forests is described below:

Step 1. Create a bootstrap sample of the training set of size M, where M denotes the size of training set. The boot sample created will be used as training data for base classifiers in forest.

Step 2. Extract ‘n’ replicate sample with replacement from training set, where ‘n’ is the number of trees in the forest.

Step 3. Grow ‘n’ classification trees in forest by selecting k out of K features at each node to tie the split where k≥ $\sqrt{M}$ or $\log_2 M$.

Step 4. Continue to grow all trees without pruning till forest grows to its maximum limit.

Step 5. Average the outcomes of all trees for prediction of unknown variable.

Random forests are one of the most popular ensemble techniques used for regression and classification without the risk of over fitting. There is no limit on the no. of trees in Random forests but it might be capped by the hardware and time constrains. Since Random forests run in parallel so can be deployed in distributed fashion easily and can be used with large amount of data. In spark MLlib we can configure the following parameters for Random forests:

- The Number of classification trees in the Forest (numTrees).
- Maximum depth of the individual tree in the forest (maxDepth).
- The fraction of data used for sub sampling (subsamplingRate).
- The fraction of features used as the candidate for node in each individual tree (featuresubsetStrategy).

3.2. Gradient Boosted Trees (GBT)

The original gradient boosted decision tree was proposed by Friedman in 1999 [31]. The Gradient boosting tree belongs to the latest line-up of powerful machine learning techniques which proved their capabilities in various practical applications. The boosting methods are different from Random Forests and follow a constructive ensemble formation strategy. The idea behind boosting is to add new learning models in continuous manner while building ensembles. After each iteration, cumulative error is considered and corresponding to the error a new, basic weak learner is trained. In Gradient boosted trees or simply GBTs, learning procedure involves consecutive accommodation of new models to produce more accurate estimation. The new base learners are constructed to correlate optimally with negative gradient of the error or loss function. The consecutive model fitting approach in gradient boosted trees makes it highly customizable and flexible for any type of data task. If (x, y) is labeled input feature vector, then gradient boosting function is given by eq (3).

$$F(x,\sigma,\gamma) = \sum_{i=1}^{n} \sigma_i h(x,\gamma_i) \tag{3}$$

Here h is the base weak learners and the summation denotes the linear combination of weak learners, the gradient boost procedure is performing two things:

- Calculation of σ_i , which is the weight of given classifier in context of ensemble learning.
- Computation of i -th weak classifier, γ_i .

The GBTs are prone to over-fitting when trained with more base classifiers, to prevent over-fitting validation is performed while training the model.

3.3. Random Forests versus Gradient-Boosted Trees

- GBTs take longer to train because Random forests grow in parallel fashion so can able to train multiple trees at a time while GBTs grow sequentially.
- Gradient boosted trees are more prone to over fitting, increasing the number of trees in Random forest can reduces the chances of over fitting but doing so in case of GBTs does not work, increasing the number of trees in GBT reduces bias.
- Random forests can be easily deployed in distributive manner because of parallel execution while Gradient boosted trees cannot as it executes trial after trial.
- Boosting works with shallow trees while Random forest requires deep trees to perform well.
- Random forests are easy to tune, in case of Random forests the performance increases indirect proportion of number of trees in forest while performance decreases in case of Gradient boosted trees.

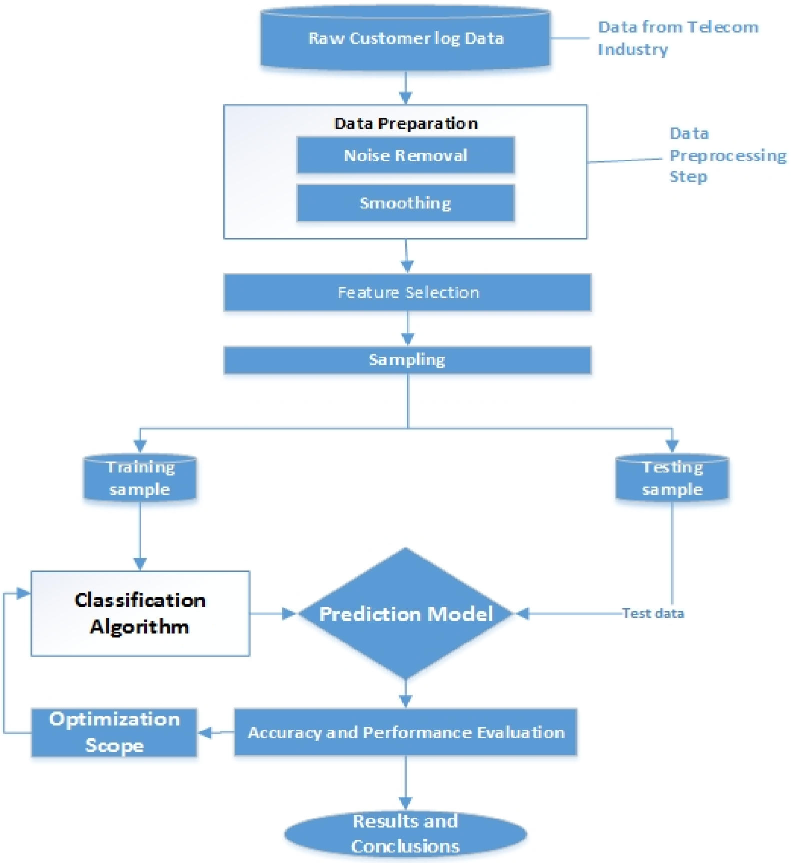


Figure 7. Work Flow of Proposed Methodology

Figure 7 presents the basic workflow that will be followed for proposed methodology. The first and initial step is obtaining the data, the obtained data generally contains many types of error and noisy data so the first step is of data pre-processing which removes noise and smoothes the data wherever needed. The output of data pre-processing step will be noise free and clean data which can be used for further processing steps.

After the data preparation the next step is of feature selection. Feature selection methods are applied to remove irrelevant, nonessential and redundant features that don't contribute much in the prediction performance. At some instances these 'extra' features may degrade the model quality. The feature selection also helped in this study to improve the overall performance of the classifier in respect of processing and memory utilization. The feature selection is different from dimensionality reduction, in case of dimensionality reduction we can create new attributes if needed but in case of feature selection we either take or leave the attributes. The next step is of sampling the data. The sampling is done to divide the data in two disjoint subsets. The sampling step is done to prevent over fitting in our model. Over fitting is an anomaly that happens in model when our model fits to noise rather than for good signals. The Sampling is one of the simplest approaches to check our model against over fitting.

In next step, the training object set is used to train our classifier, the classifiers that are used in this work are Decision trees, Random Forest and Gradient boosted trees, the detailed description of each classifier is given individually in different subsections. Further, the prediction model is built and ready for testing. The testing is done using the testing object set prepared in sampling step. The test objects are provided as input to prediction model and the performance and optimization scope is evaluated, if performance of classifier is not satisfactory then optimization is performed. The optimization step is focused on the parameter optimization and the parameters of decision tree, Random forest and Gradient boosted trees will be optimized to get better results and some performance related optimizations like cache, execution time, tuning, *etc.* is also done. The last step is of the performance and accuracy evaluation. The performance of all three classifiers evaluated using accuracy, confusion matrix and Receiver Operating Characteristic (or ROC) curve.

3.4. Churn Dataset Description

The dataset used in this thesis has been obtained from a wireless telecom company and available in public domain by SGI. The collected dataset has total 21 attributes representing the customer usage and plan details of customers. Dataset was in TSV (Tab separated Value) format and to use with Apache spark, TSV dataset was converted into LIBSVM [32] data format using a python script. The attribute details are listed in Table 1. Originally there was 21 attributes but after selecting relevant features and dropping some irrelevant and nonessential attributes like state, area code, phone number, *etc.* the number of features considered for study were sixteen. There was total 3333 tuples in final data set; out of 3333 the 483 examples were for churning customer and 2850 examples of non-churning or loyal customers. Since the ratio of churners to non-churners is very small so it the class distribution is highly skewed towards non-churn class which leads to class imbalance. Some researchers focus the class imbalance problem in data pre-processing phase and the most popular technique used by them is sampling [19]. However contrary to traditional class imbalance challenge in churn prediction, a recent research [20] suggests that oversampling or under sampling of the churn data does not affect the performance of prediction model and the framework used for study is able to handle it without any problem.

Table 1. Data Set Description

S. No.	Attribute	Attribute Description
1	STATE	The state to which customer belongs
2	ACC_LENGTH	The customer engage duration (in weeks)
3	AREA_CODE	The area code of customer
4	PHONE_NUMBER	The subscriber identification number
5	INTERNATIONAL_PLAN	Subscription status of International calling
6	VOICE_MAIL_PLAN	Subscription status of voice mail service
7	VOICE_MAIL	Number of voice mails
8	TOTAL_DAY_MINUTS	Total day minute usage
9	TOTAL_DAY_CALLS	No. of calls made by customer in day time
10	TOTAL_DAY_CHARGE	Total charge of day calls
11	TOTAL_EVE_MINUTES	Total evening usage in minutes
12	TOTAL_EVE_CALLS	No. of calls made by customer in evening
13	TOTAL_EVE_CHARGE	Total charge of evening calls
14	TOTAL_NIGHT_MIN	Total night usage in minutes
15	TOTAL_NIGHT_CALLS	No. of calls made by customer in night
16	TOTAL_NIGHT_CHARGES	Total bill charged of night calls
17	TOTAL_INT_MINUTE	Total duration of international calls
18	TOTAL_INT_CALLS	Total number of international calls
19	TOTAL_INT_CHARGE	Total bill charged for international calls
20	CC_CALLS	Number of calls made to customer care
21	**CHURN**	Churn status of customer

3.5. Decision Tree Classifier

After the data was pre-processed, the first classification algorithm that was used to build the prediction model was Decision Trees. Among several algorithms of decision tree, C5.0 was selected for primary classification model build-up. The reasons of selecting decision trees were their easy interpretation, ability to handle categorical and non-linear features.

3.5.1. Parameters

The parameters to build the churn prediction model by using decision trees in MLlib have following parameters:

- *Number of classes for Classification (numClassesForClassification):* this parameter holds the total number of classes hold for classification, in this study there are two classification classes one for the 'churners' and other is for the 'non-churners'.
- *Impurity:* the 'impurity' parameter denotes the way of measuring the impurity degree in trees. There are many ways to measure gain information and impurity in s decision trees, "Gini" and "Entropy" as given by eq(4) and eq(5) respectively are the most used ones. For our purpose we selected 'Gini' as our impurity criteria, the reasons for selecting Gini are:

✓ "Gini" tends to minimize the misclassification probability.

✓ "Gini" values are faster to compute.

$$\text{Gini Impurity} = \sum_{i=1}^{M} f_k(1-f_k) \qquad (4)$$

$$\text{Entropy}=\sum_{i=1}^{M} -f_k \log(f_k) \qquad (5)$$

Here f_k is the occurrence of label k at a node and M is the count of unique labels.

- *Maximum Depth (maxDepth):* The maximum depth parameter in decision tree is the cap(maximum) value on the length between the root and the leaf nodes of the decision tree reached during tree building. For initial experiments we would take the value of tree depth equals to six.
- *Number of bins (maxBins):* The number of bins in the decision tree determines the number of split candidates at each node to change continuous features into discrete features. Increasing the value of bin value turns in fine grained split candidates but also increases the computational overhead. The max value of bin should be greater or equal to the number of categories in any categorical feature set.

3.5.2. Model Development Methodology

The first step towards prediction modeling was building model (or training) with help of decision trees and using the parameters described in previous section. So, first of all we divided the dataset in training (75%) and testing (25%) subsets. After that we built the DecisionTreeModel for depth 5 with 53 nodes using Apache spark framework [33-34]. The deduced model gives the accuracy of around 86% and the optimization of model is performed and discussed in next chapter.

3.6. Random Forest Classification

The Random forests were the first ensembles that were used for creating prediction model. The Random forests were averaged by majority voting. The information content at node K, when input to node K is Z is given in eq(6):

$$I(K)=|X|H(Z)-|X_L|H(X_L)-|X_R|H(X_R) \qquad (6)$$

Where |X|= input sample size, *i.e*. 75% of original dataset in our case. Size of right and left subclasses of Z is denoted by X_R and X_L and H(X) denotes the Shannon Entropy.

3.6.1. Parameters

The parameters to build the churn prediction model using Random forests in Spark MLlib have following parameters:

- *Number of Trees in forest (numTrees):* This is the total number of decision trees grown in forest for classification. The number of trees are not fixed, the more trees in forest will decrease the variance thus produce better results but may take longer execution time because of the increased training time. In our experiments the number of trees was kept 15.
- *Maximum depth (maxDepth):* depth is the same as defined in classification trees, increasing the maximum depth value increases expressive power of trees and also makes them more powerful but may prone to overfitting .The deeper trees takes longer to train.
- *Sub sampling rate (ssRate):* The sub-sampling rates means the proportion of the dataset used to train each tree in the forest, the recommended value is 1.0 but it can be modified to speed up the training.
- *Feature subset strategy (fsStrategy):* The fraction of the total number of features used as candidate for splitting at each internal node of the tree is called feature subset strategy.

3.6.2. Model Development

The random forest classifier was built by populating 20 trees in forest, the maximum depth was defined as 5, sub-sampling strategy was fixed at 1.0 and the feature used for splitting at each node was 8. For tree learning 75% of dataset was used and rest *i.e.* 25% was used for testing the model.

3.7. Gradient Boosted Trees

Gradient boosted trees (GBTs) are another ensemble approach used for constructing classifier for churn prediction. GBTs try to minimize the loss function by iterative tree training. The 'Logarithmic loss' (log loss) is selected as the loss function for GBTs because they are least biased for binary classification. If number of instances is M, then logarithmic loss is calculated as per eq(7):

$$logloss = 2\sum_{i=1}^{M} \log\left(1+\exp\left(-2 l_i F(x_i)\right)\right) \tag{7}$$

Where l_i and x_i are the labels and features if instance i respectively. $F(x_i)$ predicted is label for instance i (by the model). Gradient boosting approach is quite simple- iteratively creates a linear collection of predictors, at each step an incremental classifier is introduced to improve performance and weighted average of all these predictors is called final predictor. The pseudo-code is as following:

Step 1. Base (weak) learners *i.e.* all decision trees initialized

Step 2. For each iteration, re-weight the inputs by 'up-weighting' examples which are not

classified by existing forest.

Step 3. Construct new classifier h_i for residuals.

Step 4. Compute weight γ_i for new base classifier.

Step 5. Include the values (γ_i, h_i) to the existing forest.

Step 6. Return the final tree from the forest.

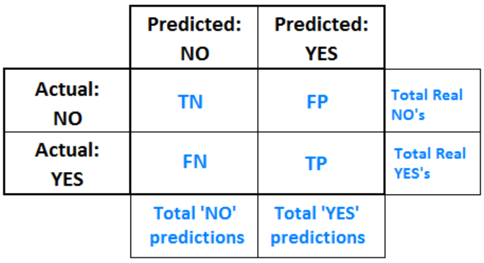
3.7.1. Parameters

The performance of Gradient boosted trees can be tuned using the basic tree parameters like tree depth, number of bins, *etc.*, other parameters are:

- *Number of iterations:* In case of Gradient boosted trees the number of iteration denotes the number of trees in forest. Increasing the number may lead to over fitting and larger training time.
- *Loss:* this parameter defines the log function used. Different loss function can produce different results. For this study 'log-loss' function is used.

4. Result and Discussion

This section mentions and discusses the results obtained in presented research work. After result explanation, the hyper parameter optimization is performed over all three models *i.e.* Decision tree, Gradient boosted trees and Random Forests. The next sub-section gives the primary results that are obtained without any optimization action and in subsequent sub-section the optimization will be performed. The last section specifies the final results which are obtained after applying brute force hyper-parameter optimization. The results have been discussed and based on performance of three classifiers; decision trees, gradient boost and random forest, assessed as per confusion matrix given in Figure 8. The basic abbreviations used in confusion matrix explained further.



	Predicted: NO	Predicted: YES	
Actual: NO	TN	FP	Total Real NO's
Actual: YES	FN	TP	Total Real YES's
	Total 'NO' predictions	Total 'YES' predictions	

Figure 8. Confusion Matrix

TP (True Positive)- the instances when the predicted churner is truly a churner.

TN (True Negatives)- the instances when the predicted non-churner is truly a non-churner.

FP (False Positives)- the instances when the predicted churner is non-churner in real.

FN (False Negatives)- the instances when the predicted non-churner is a churner in real.

The classifier evaluation parameters used for classifier rankings are:

Accuracy- The accuracy specifies correctness of classifiers and defined as per eq(8)

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{8}$$

Sensitivity- The sensitivity measures how often the churners are predicted in actual churners, it is defined as eq(9):

$$Sensitivity = \frac{TP}{FN+TP} \tag{9}$$

Specificity- The specificity tells us how often the non-churners are predicted in actual non-churners, it is given by eq(10):

$$Specifity = \frac{TN}{TN+FP} \tag{10}$$

4.1. Primary Results

4.1.1. Decision Trees

The Decision Tree gave the prediction Accuracy of 86.07% when tree depth was fixed at 5 and the number of iterations was 3. Corresponding confusion matrix is shown in Table 2.

Table 2. Decision Tree Confusion Matrix (Primary)

	Predicted: non-churners	Predicted: Churners
Actual: non-churners	2763	87
Actual: Churners	377	106

4.1.2. Gradient Boosted Trees

The first ensemble approach that we used was Gradient boosted trees and it gives the classification accuracy of 92.32%, which is better than the accuracy of decision trees. The Test: Train dataset split ratio was 7:3 and max depth was fixed at 6 and the number of iterations was 7. Corresponding confusion matrix is shown in Table 3.

Table 3. GBT Confusion Matrix (Primary)

	Predicted: non-churners	Predicted: Churners
Actual: non-churners	2816	34
Actual: Churners	244	239

4.1.3. Random Forests

The Random forest gives the classification accuracy ranging from 92% to 93%. While performing the experiment the number of trees in forest was 10 and the depth of each tree was limited to 5 and the number of bins was 20. Corresponding confusion matrix is shown in Table 4.

Table 4. Random Forest Confusion Matrix (Primary)

	Predicted: non-churners	Predicted: Churners
Actual: non-churners	2820	30
Actual: Churners	255	228

A summary of all the primary results is depicted in Table 5.

Table 5. Summary of Primary Results

Classifier / Parameter	Decision Tree	Gradient Boosted Trees	Random Forest
Accuracy	0.8607	0.9165	0.9144
Sensitivity	0.2194	0.4948	0.4720
Specificity	0.9694	0.9880	0.9894

4.2. Optimized Results

The primary results obtained in previous section are obtained using general input parameters and are not specific to our dataset analysis. To get better prediction results and performance we performed 'Hyper-parameter optimization' in which the parameters for the classification algorithm was decided with Grid search approach. The hyper parameter grid search algorithm works and tune the parameters in following manner:

1. For every input parameter k, a list of possible values is prepared which is based on the theoretical knowledge.
2. Cartesian product set (or grid) of these possible values are prepared and cost of each set is calculated.
3. The parameter set with lowest cost is selected. Parameter grid with lowest cost is also known as 'Hyper parameter' grid.

4.2.1. Decision Trees

In case of decision trees we have two parameters in hand and they are 'Number of iterations' and 'Tree Depth'. If we take number of iteration equals to 5 and tree depth(TD) as TD={3,5,6,7,8,9,10,11,12,13,15,18,20,22,25} then we get a graph as given in Figure 9.

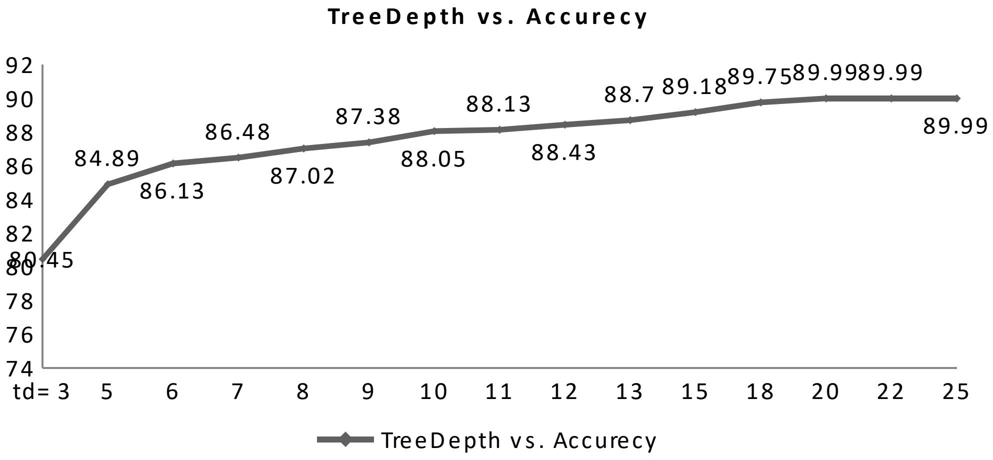


Figure 9. Tree Depth vs. Accuracy (Decision Tree)

By analyzing above table we get can deduce that prediction of grid {5, 20} is highest, Now in next step we kept TD=20 and vary the I *i.e.* Number of iterations. The result obtained is listed in Table 6.

Table 6. Decision Tree Results

Number of iterations	Tree depth	Pred. Accuracy
1	20	89.99
2	20	89.99
3	20	89.99

So finally we can deduce that grid set with highest accuracy is {3, 20} and the accuracy over this hyper-parameter is 89.99% which is around 5% improvement over primary results. With these optimized settings, the corresponding confusion matrix is shown in Table 7.

Table 7. Decision Tree Confusion Matrix (Optimized)

	Predicted: non-churners	Predicted: Churners
Actual: non-churners	2812	38
Actual: Churners	296	187

4.2.2 Gradient Boosted Trees

The parameters that can be optimized for Gradient boosted trees were ‘maximum depth’ and ‘number of iterations’, the sampling rate for training and testing were kept in the ratio of 7:3. The variation of accuracy with iteration is shown in Figure 10. Tree depth (TD) was fixed at 6 and iteration parameter grid was {1,2,3,5,7,10,13,15,20,25}.

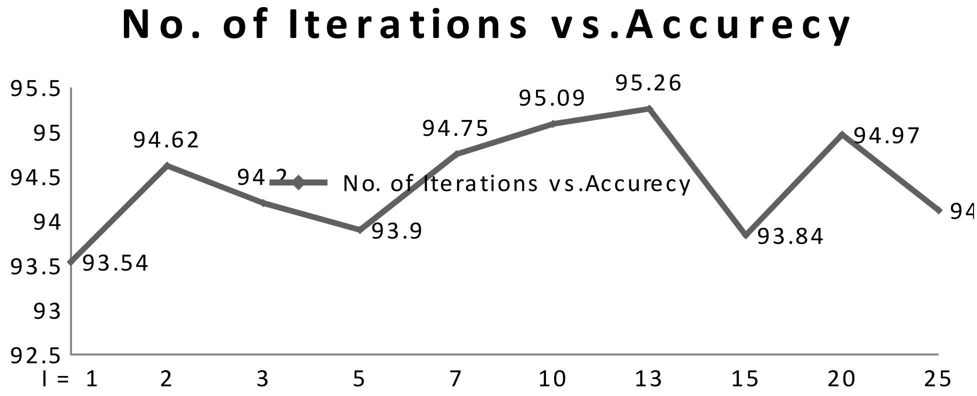


Figure 10. Iteration vs. Accuracy (GBT)

The maximum accuracy was observed when numbers of iterations were in between 11 to 13. The accuracy for the 11 iterations was optimal. Now considering the maximum depth grid {2,3,4,5,7,10,12,15,20}, The cost graph will be as per Figure 11.

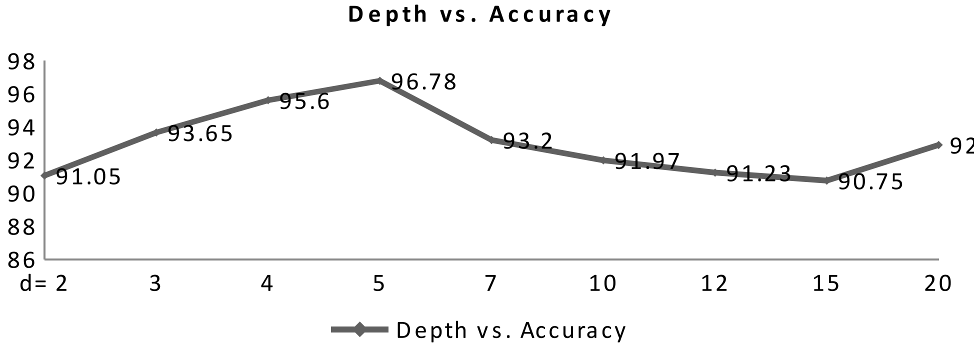


Figure 11. Tree Depth vs. Accuracy (GBT)

The best prediction accuracy is 96.78% and is observed at depth=5, so by observing both parameters vs. accuracy plots the optimal parameters were depth=5 and iterations=11 and corresponding confusion matrix is mentioned in Table 8.

Table 8. GBT Confusion Matrix (Optimized)

	Predicted: non-churners	Predicted: Churners
Actual: non-churners	2828	22
Actual: Churners	85	398

4.2.3 Random Forest

The parameters used in random forests are the number of trees, depth of tree and number of bins used for splitting the nodes. The data was sampled in the ratio of 7:3 for training and testing purpose. The number of bins was fixed at 16. The variation of accuracy with depth parameter set {4,5,6,7,8,9,10,12,15} is given in Figure 12.

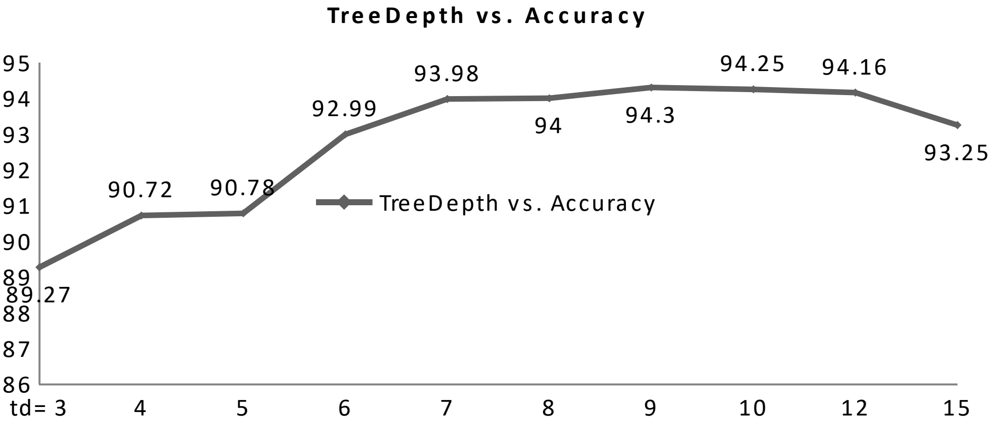


Figure 12. Tree Depth vs. Accuracy Chart (RF)

It is evident that maximum performance was captured on depth =8 or 9, so now the depth is fixed at 8 and The number of trees in the forest is varied to find the hyper parameter set for the Random Forests. The variation of accuracy with number of trees is given in Figure 13.

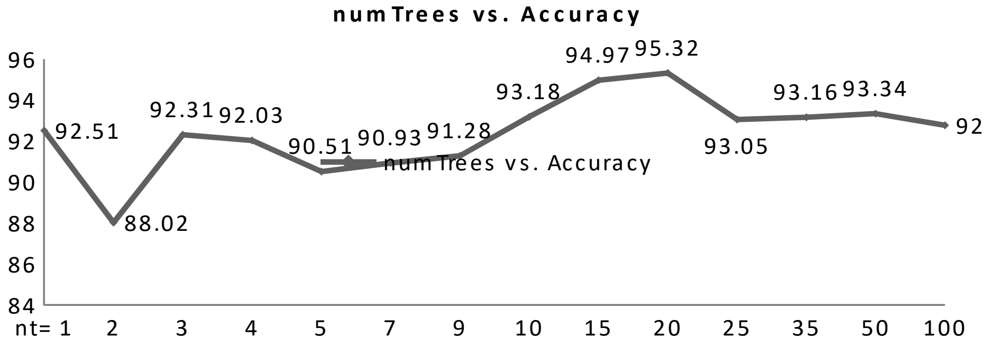


Figure 13. Number of Tree vs. Accuracy (RF)

Finally the peak performance is 95.32% and the observed when number of trees was 20, tree depth=8 and the number of bins was 16. The corresponding Random forest classifier confusion matrix will be as depicted in Table 9.

Table 9. Random Forest Confusion Matrix (Optimized)

	Predicted: non-churners	Predicted: Churners
Actual: non-churners	2834	16
Actual: Churners	140	343

The optimized results for all the classifiers are summarized in Table 10.

Table 10. Summary of Optimized Results

Classifier Parameter	Decision Tree	Gradient Boosted Trees	Random Forest
Accuracy	0.8999	0.9678	0.9531
Sensitivity	0.3871	0.8240	0.7101
Specificity	0.9922	0.9923	0.9943

5. Conclusion and Future Work

This paper presents an efficient methodology to anticipate churn in subscription based industries. Bagging and Boosting based ensembles of decision tress are used and their performance has been evaluated using various performance metrics. The ensembles used are Random forest (Bagging) and Gradient boosted trees (Boosting). These ensemble based approaches especially the residual feedback-improvement based Gradient boosted tree ensemble approach outperformed the other classifiers tested for the study and results obtained in previous studies. The Gradient boosted tree performed better than Random forest in terms of accuracy and sensitivity. The optimization phase makes the results more accurate and refined. The methodology is tested against the data obtained from telecom industry and performance of classifier is found reliable.

In future, the presented methodology can be applied over the modern data sources like real time streaming data to obtain the real time churn forecasting and will be more suitable for data driven industries. The idea of churn prediction can also be extended to other domains like employee churn, college dropout prediction *etc.*

References

[1] C. Paul, “Keep those customers [retaining and expanding existing customer base”, Engineering Management, vol. 14, no. 6, **(2004)**, pp. 22-23.

[2] L. D. Xu, “Enterprise systems: State-of-the-art and future trends”, IEEE Trans. Ind. Inf., vol. 7, no. 4, **(2011)**, pp. 630-640.

[3] D. Tomar and S. Agarwal, “A survey on Data Mining approaches for Healthcare”, International Journal of Bio-Science and Bio-Technology, vol. 5, no. 5, **(2013)**, pp. 241-266.

[4] D. Tomar and S. Agarwal, “A survey on pre-processing and post-processing techniques in data mining”, International Journal of Database Theory and Application, vol. 7, no. 4, **(2014)**.

[5] S. Agarwal, G. N. Pandey and M. D. Tiwari, “Data mining in education: data classification and decision tree approach”, International Journal of e-Education, e-Business, e-Management and e-Learning, vol. 2, no. 2, **(2012)**, pp. 140.

[6] D. Tomar, R. Arya and S. Agarwal, “Prediction of profitability of industries using weighted SVR”, International Journal on Computer Science and Engineering, vol. 3, no. 5, **(2011)**, pp. 1938-1945.

[7] B. R. Prasad, and S. Agarwal, “Modeling risk prediction of diabetes a preventive measure”, In Industrial and Information Systems (ICIIS), 2014 9th International Conference on, IEEE, **(2014)**, pp. 1-6.

[8] N. Singh, S. Agarwal, and R. C. Tripathi, “A Data Mining Perspective on the Prevalence of Polio in India”, International Journal on Computer Science and Engineering, vol. 3, no. 2, **(2011)**, pp. 580-585.

[9] D. Tomar, B. R. Prasad and S. Agarwal, “An efficient Parkinson disease diagnosis system based on Least Squares Twin Support Vector Machine and Particle Swarm Optimization”, 2014 9th International Conference on Industrial and Information Systems (ICIIS), Gwalior, **(2014)**, pp. 1-6.

[10] S. Agarwal and S. K. Sinha, “Data mining based pervasive system design for Intensive Care Unit”, In Computer Communication and Informatics (ICCCI), 2014 International Conference on, IEEE, **(2014)**, pp. 1-6.

[11] A. Berson, K. Thearling and S. Smith, “Building Data Mining Applications for CRM”, New York: McGraw-Hill, **(1999)**.

[12] SEM Research, “The SEMPO Annual State of Search Survey 2015”, Search Engine Marketing Research, Articles and Resources, **(2015)**.

[13] K. Coussement and D. V. Poel, “Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques”, Expert Syst. Appl., vol. 34, no. 1, **(2008)**, pp. 313–327.

[14] P. Datta, B. Masand, D. R. Mani, and B. Li, “Automated cellular modeling and prediction on a large scale”, Artif. Intell. Rev., vol. 14, no. 6, **(2000)**, pp. 485-502.

[15] M. Owczarczuk, “Churn models for prepaid customers in the cellular telecommunication industry using large data marts”, Expert Syst. Appl., vol. 37, no. 6, **(2010)**, pp. 4710-4712.

[16] C. P. Wei and I. T. Chiu, “Turning telecommunications call details to churn prediction: A data mining approach”, Expert Syst. Appl., vol. 23, no. 2, **(2002)**, pp. 103-112.

[17] D. Popović and B. D. Bašić, “Churn prediction model in retail banking using fuzzy C-means algorithm”, Informatica, vol. 33, no. 2, **(2009)**, pp. 235-239.

[18] D. A. Kumar and V. Ravi, “Predicting credit card customer churn in banks using data mining”, International Journal Data Anal. Tech. Strategies, vol. 1, no. 1, **(2008)**, pp. 4-28, **(2008)**.

[19] G. M. Weiss, “Mining with rarity: A unifying framework”, ACM SIGKDD Explorations Newslett., vol. 6, no. 1, **(2004)**, pp. 7-19.

[20] J. Burez and D. V. Poel, "Handling class imbalance in customer churn prediction", Expert Syst. Appl., vol. 36, no. 3, **(2009)**, pp. 4626-4636.

[21] C. Wei and I. Chiu, "Turning telecommunications call details to churn prediction: a data mining approach", Expert systems with applications, vol. 23, no. 2, **(2002)**, pp. 103-112.

[22] W. Au, K. C. Chan and X. Yao, "A novel evolutionary data mining algorithm with applications to churn prediction", Evolutionary Computation, IEEE Transactions on, vol. 7, no. 6, **(2003)**, pp. 532-545.

[23] H. John, "Computer assisted customer churn management: State-of-the-art and future trends", Computers & Operations Research, vol. 34, no. 10, **(2007)**, pp. 2902-2917.

[24] H. Hwang and T. E. Suh, "An LTV model and customer segmentation based on customer value: a case study on the wireless telecommunications industry", Expert Systems with Applications, vol. 26, **(2004)**, pp. 26:181–189.

[25] R. Mattersion, "Telecom churn management", Fuquay-Varina, NC: APDG Publishing, **(2001)**.

[26] S. Hung, D. C. Yen and H. Wang, "Applying data mining to telecom churn management", Expert Systems with Applications, vol. 31, no. 3, **(2006)**, pp. 515-524.

[27] A. Lemmens and C. Croux, "Bagging and Boosting Classification Trees to Predict Churn", Journal of Marketing Research, vol. 43, no. 2, **(2006)**, pp. 276-286.

[28] S. Jinbo, L. Xiu, and L. Wenhuang, "The Application of AdaBoost in Customer Churn Prediction", Service Systems and Service Management, 2007 International Conference on. IEEE, **(2007)**.

[29] C. Tsai and Y. Lu, "Customer churn prediction by hybrid neural networks", Expert Systems with Applications, vol. 36, no. 10, **(2009)**, pp. 12547-12553.

[30] B. Leo, "Random forests", Machine learning, vol. 45, no. 1, **(2001)**, pp. 5-32.

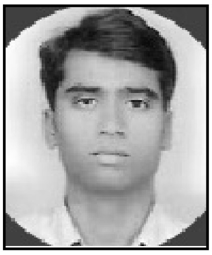
[31] J. H. Friedman, "Greedy function approximation: A gradient boosting machine", Annals of Statistics vol. 29, **(2001)**, pp. 1189-1232.

[32] C. C. Chung and C. J. Lin, "LIBSVM: A library for support vector machines", ACM Transactions on Intelligent Systems and Technology (TIST), vol. 2, no. 3, **(2011)**, pp. 27.

[33] S. Agarwal and B. R. Prasad, "High speed streaming data analysis of web generated log streams", 2015 IEEE 10th International Conference on Industrial and Information Systems (ICIIS), Peradeniya, **(2015)**, pp. 413-418.

[34] Apache Spark, http://spark.apache.org/, retrieved on Mar 2016.

Authors



Pretam Jayaswal, has received his Master of Technology from Indian Institute of Information Technology, Allahabad under the supervision of Dr. Sonali Agarwal. His is an active researcher in the field of data mining and big data analytics and its applications in business intelligence and other pertinent domains.

Bakshi Rohit Prasad, is a research scholar in Information Technology Division of Indian Institute of Information Technology, Allahabad. His primary research interests are Data Mining, Machine Learning, Big Data Storage, Computing and Algorithms to deal with related issues. Also, he has significant publications in high speed streaming data mining, analytics and optimizations and their applications in several domains.

Divya Tomar, is a research scholar in Information Technology Division of Indian Institute of Information Technology, Allahabad, India under the supervision of Dr. Sonali Agarwal. Her primary research interests are Data Mining, Data Warehousing especially with the application in the area of Medical Healthcare. Also she has keen interest machine learning algorithms and its application in variety of domains.

Sonali Agarwal, is working as an Assistant Professor in the Information Technology Division of Indian Institute of Information Technology, Allahabad, India. Her primary research interests are in the areas of Data Mining, Data Warehousing, E Governance and Software Engineering. Her current focus in the last few years is on the research issues in Machine Learning algorithms big data processing and analytics.