CHURN PREDICTION IN THE TELECOMMUNICATION INDUSTRY USING MACHINE LEARNING TECHNIQUES

by

Mehreen Ahmed

A thesis submitted to the faculty of Computer Science Department,
Military College of Signals, National University of Sciences and Technology,
Islamabad, Pakistan, in partial fulfillment of the requirements for the degree of MS in
Computer Science (Software) Engineering

April 2016

SUPERVISOR CERTIFICATE

It is to certify that the final copy of thesis has been evaluated by me, found as per the specified format and error free.

Date__________

____
(Asst Prof Dr Hammad Afzal)

ABSTRACT

Retaining customers is an important challenge for the telecom industry in this day and age. The changing behavior of the users and the type of services being offered every day makes it difficult to predict the churners. Thus the service providers find it more convincing to retain an ongoing customer rather than running after new subscribers. Churn prediction helps big companies in cost savings and they can identify reasons of why the subscriber is unsatisfied with their service. Data mining techniques can help in predictive analysis and creating models that can give accurate classification of the churners and non-churners. The most recent techniques being employed are the ensemble learning techniques, which considers using a combination of learners instead of a single classifier to increase the classification accuracy.

In this thesis, we explore the use of ensemble learning techniques for customer churn prediction. We evaluate the performance, the usage of efficient features and the classification techniques on a public and a private churn data set in the telecom industry. The proposed framework is a combination of the bagging and stacking ensemble learning techniques with three base learners namely Neural Network, K-Nearest Neighbors and Decision Tree. This in turn produces a bagged-stacked Meta Decision Tree that predicts 98% of the churned customers in the UCI dataset and 90% churners in the private data set. The results reveal that the proposed framework is more efficient and accurate as compared to the state of the art and the simple ensemble techniques.

DEDICATION

This thesis is dedicated to
MY FAMILY, FRIENDS AND TEACHERS
for their love, endless support and encouragement

ACKNOWLEDGMENTS

I am grateful to God Almighty who has bestowed me with the strength and the passion to accomplish this thesis and I am thankful to Him for His mercy and benevolence. Without his consent I could not have indulged myself in this task.

TABLE OF CONTENTS

ABSTRACT .... **ii**

DEDICATION .... **iv**

ACKNOWLEDGMENTS .... **v**

LIST OF FIGURES .... **viii**

LIST OF TABLES .... **ix**

ACRONYMS .... **X**

1 INTRODUCTION **11**

1.1 INTRODUCTION 11

1.2 PROBLEM STATEMENT AND OBJECTIVES 11

1.3 CONTRIBUTIONS 12

1.4 THESIS OUTLINE 12

2 LITERATURE SURVEY **14**

2.1. LITERATURE REVIEW 14

2.2. DATA SETS 14

2.3. CLASSIFICATION OF THE ARTICLES 15

2.3.1. STATE-OF-THE-ART TECHNIQUES 17

2.3.2. RULE INDUCTION TECHNIQUES 19

2.3.3. SAMPLING TECHNIQUES 23

2.3.4. ENSEMBLE TECHNIQUES 24

2.3.5. NOVEL APPROACHES 26

2.3.6. NOVEL FEATURES 28

2.3.7. CLUSTERING TECHNIQUES 30

3 ENSEMBLE CLASSIFICATION **32**

3.1. ENSEMBLE CLASSIFICATION 32

3.2. ENSEMBLE LEARNING 32

3.2.1. ENSEMBLE GENERATION 33

3.2.2. ENSEMBLE PRUNING 34
3.2.2.1. OVERPRODUCE AND CHOOSE APPROACH 34
3.2.3. ENSEMBLE INTEGRATION 37

4 MACHINE LEARNING TECHNIQUES 40

4.1. BASE CLASSIFIERS 40
4.1.1. K NEAREST NEIGHBOR 40
4.1.2. LOGISTIC REGRESSION 41
4.1.3. NAÏVE BAYESIAN 41
4.1.4. NEURAL NETWORK 42
4.1.5. CLASSIFICATION / DECISION TREES 43
4.2. ENSEMBLE OR META LEARNERS 45
4.2.1. ADAPTIVE BOOSTING 45
4.2.2. STACKED GENERALIZATION 46
4.2.3. BAGGING 48

5 EXPERIMENTAL STUDY 50

5.1. DATA SETS 50
5.1.1. INPUT FEATURES 50
5.2. PROPOSED FRAMEWORK 52
5.3. DATA PREPROCESSING 55
5.4. EVALUATION MEASURES 57
5.5. ENSEMBLE BUILDING 59
5.5.1. MODEL TRAINING 59

6 CONCLUSIONS AND FUTURE WORK 69

6.1. CONCLUSION 69

BIBLIOGRAPHY 72

LIST OF FIGURES

Figure Number *Page*

Figure 1: Classification Framework for Customer Churn Prediction In The Telecom Industry 14
Figure 2: Customer Data used for Churn Prediction in the Telecom Industry 15
Figure 3: Public Data Sets used in Number of Articles 17
Figure 4: Techniques applied in churn prediction between 2000 and 2015 25
Figure 5: Overproduce and Choose Approach 36
Figure 6: Cascading Classifiers 38
Figure 7: Parallel Classifier 38
Figure 8: Hierarchical Classifier 38
Figure 9: Ensemble Classification 39
Figure 10: Multi-Layer Perceptron 43
Figure 11: Adaptive Boosting 46
Figure 12: Stacked Generalization 47
Figure 13: Bootstrap Aggregation 49
Figure 14: Bag-Stack Ensemble Learner 54
Figure 15: Ensemble Generation 55
Figure 16: ROC Curve 58
Figure 17: Conceptual Ensemble Formation 60
Figure 18: Result of Baseline Experiments on Test Set 61
Figure 19: Experiments with Boosted set with different ensemble size 62
Figure 20: Experiments with Bagged set with different ensemble size 63
Figure 21: Results with k base learners 65
Figure 22: Comparison of Ensemble Techniques 69
Figure 23: Comparison between Bagged-stacked and Boosted-stacked ensemble 70

LIST OF TABLES

Table Number *Page*

Table 1: Public Data Sets........16

Table 2: Summary of Literature for Churn Prediction........20

Table 3: Distribution of Papers by Techniques........31

Table 4: Data sets used in the empirical evaluation........52

Table 5: Confusion Matrix........58

Table 6: True and False cases........58

Table 7: Results of Stand-alone Experiments........60

Table 8: Stacked Model experiments with different base learners........64

Table 9: Best Results with Boosting and Stacking Techniques........65

Table 10: Best Results with Bagging and Stacking Techniques........66

Table 11: Size and Accuracy with different ensembles........66

Table 12: Results with the Top two ensemble models........67

Table 13: Results with Bag-Stack Ensemble Model........67

Table 14: Comparison of literature reported results on real life churn data set (dataset 2)..68

ACRONYMS

Customer Relationship Management	CRM
Call Detail Records	CDR
Chi-squared Automatic Interaction Detector	CHAID
Bayesian Belief Network	BBN
Naïve Bayesian	NB
Artificial Neural Networks	ANN
Support Vector Machines	SVM
Radial basis Function	RBF
Multilayer Perceptron	MLP
Classification and Regression Tree	CART
Adaptive Boosting	AdaBoost
Decision Tree	DT
Back-Propagation Network	BPN
Logistic Regression	LR
Receiver Operating Characteristics	ROC
Area under the ROC curve	AUC

Chapter 1

INTRODUCTION

1.1 Introduction

Customer churn prevention is a major part of the Customer Relationship Management (CRM). Customers who terminate their subscription of a service are called churners. Thus Churn describes the subscribers who have terminated their relationship with the service provider and moved their business to the competitor. These subscribers mostly churn because they are displeased with their current carrier due to several reasons including “poor call quality”, “lack of value-added services” and “support” etc. Churn rate is the rate of churn in a certain period of time. Companies want to decrease the churn rate as much as possible. Churn prediction helps big companies in cost savings and they can identify reasons of why the subscriber is unsatisfied with their service. Customer Churn prediction can help the service providers target the subscribers who are most likely to churn and offer them customer-oriented services. In doing so, time and money can be saved and carriers would focus on a fraction of the customers among the millions. Hence little improvements in customer retention can lead to major profit for the telecom company.

The recent advancement in analysis of information systems methods particularly in use of the information based prediction methods has also inspired the Telecommunication industry. Researchers have been working on developing such models using Machine Learning techniques like Neural Networks, Support Vector Machines and Decision Trees etc. to predict the potential churners [1-3]. Data mining techniques can help in predictive analysis and creating models that can give accurate classification of the churners and non-churners. The most common technique however used by researchers is the decision trees because of their simplicity, high precision and quick results[4]. The most recent techniques being employed are the ensemble techniques, which consider using a combination of learners instead of one that in turn increases the classification accuracy.

1.2 Problem Statement and objectives

In this research, the aim is to generate a perfect ensemble model that is suitable for the telecom industry to predict the potential churners along with the set of features that might play an important role in the prediction. We concentrate on the Stacking ensemble technique and observe how the number of base learners affects the outcome. We investigate if stacked generalization has better results with the increase of number of base learners. And is combining the two models bag-stacking and boosted-stacking better than the individual stacking? Which among the two gives the better performance?

Objectives of this thesis are:

- To design and develop a framework for churn prediction using the new ensemble learning techniques and verify the correct churners and non-churners among the customers with this framework.
- To propose more effective features for churn prediction in the telecom data set to improve prediction rate.
- To explore the ensemble classification process and discover how the perfect ensemble model is generated for utilization in the telecom industry.
- To investigate if a general framework can be designed that is suitable for both balanced and imbalanced data sets.

1.3 Contributions

The contributions of this thesis are summarized as,

- A new ensemble framework for the prediction of the churners in the telecom industry.
- Introduction of new set of features for predicting the telecom churners.
- A detailed analysis of the ensemble classification process.

Here the overview of the proposed technique is given; the detailed explanation of the proposed technique is given in the chapters to follow.

Bagged-Stacked Ensemble Model.

Proposed technique is a combination of the stacking learner with other boosting and bagged learners to make a Bagged-stacked ensemble model that is more precise and accurate (Chapter 5). This ensemble model gathers the churned data and creates bootstrapped sample sets that form diverse set of models stacked together into a single meta-model. The underlying three base learners for this ensemble include the Neural Network, Decision Tree and K-Nearest Neighbor heterogeneous algorithms that are given to the stacked learner and a Meta decision tree model is formed predicting the correctly classified potential churners. Algorithms are evaluated by experiments on two data sets; UCI data set and a private churn data set.

1.4 Thesis Outline

The thesis is organized as follows. In Chapter 2, a literature review takes place on churn prediction and ensemble learning technique. Chapter 3 presents the ensemble classification and Chapter 4 discusses the algorithms applied. Chapter 5 details on the experimental evaluation on the UCI and real life data sets. Finally conclusions are drawn and future work is presented in Chapter 6.

A brief overview of each of the remaining chapters is given below.

- Chapter 1-Introduction: This Chapter contains introduction and objectives. It also contains the contributions we have made in this thesis report.

- Chapter 2 –Literature Survey: This Chapter presents an overview of existing state-of-the-art approaches as well as the new techniques applied on the telecom churn data sets. The unconventional features or novel approaches that have been used so far in the telecom industry.

- Chapter 3– Ensemble Classification: This Chapter presents an overview of the ensemble classification process that is needed to understand the framework proposed in thesis dissertation. The topics discussed in detail in this chapter include the ensemble classification process which comprises ensemble generation and integration.

- Chapter 4 – Machine Learning Techniques: This Chapter presents the machine learning models that have been used in the ensemble generation. It also introduces the ensemble techniques like Bagging, Boosting used for building the final ensemble framework.

- Chapter 5 – Experimental Study: This Chapter deals with the experiments performed on the public and private data sets in detail and discusses the implementation details of the ensemble classification methods which were discussed in earlier chapters. The design and implementation detail of the Bagged-Stacked framework is discussed.

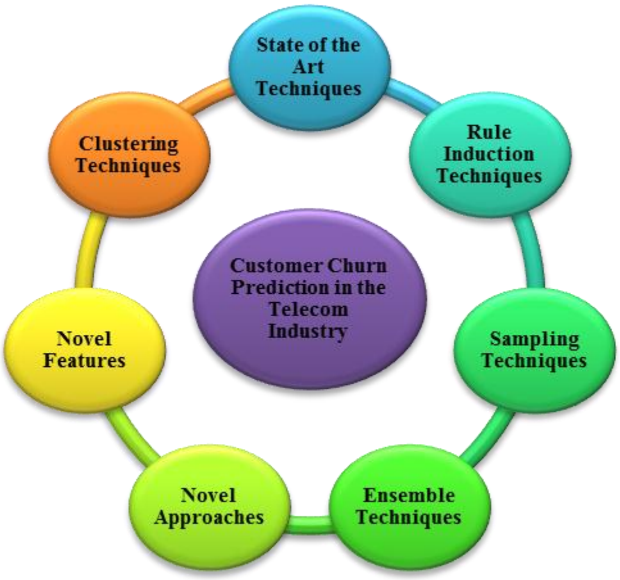
- Chapter 6 – Conclusions and Future Work: The last Chapter concludes the thesis with a brief summary of achievements made during the course of research. It also describes some of the research questions which we identified during our work that can be further developed to improve the ensemble classification process.

Chapter 2

LITERATURE SURVEY

2.1. Literature Review

The articles included in the literature were reviewed and classified into seven categories as shown in Figure 1. The classification method was based on the various techniques, sampling methods, novel approaches and features used over the years in the telecom customer churn prediction.



State of the Art Techniques
Rule Induction Techniques
Sampling Techniques
Ensemble Techniques
Novel Approaches
Novel Features
Clustering Techniques
Customer Churn Prediction in the Telecom Industry

Figure 1: Classification Framework for Customer Churn Prediction In The Telecom Industry

2.2. Data Sets

Generally the customer churn data sets used by researchers are collected from various telecommunication companies. These real life data sets have different number of subscribers and features. Mostly the features used in these data sets include the customer's personal data, billing information and the call detail records (CDR) as seen in Figure 2. A few standard publicly available data sets have been used in different studies. Figure 3 shows that the most famous among them is taken from the University of California (UCI) Machine Learning repository, which is artificially created based on the claims to the real world. The UCI data set has around 20 attributes for 5000 subscribers. The other publicly available telecom churn data set used is taken from the Teradata Center for Customer Relationship Management at the Duke University. The Teradata data set has around 170 attributes for over 100,000 subscribers. The KDD Cup challenge held in 2009 has a large and a small data set taken from Orange Telecom Company. Both data sets have 50,000 samples. Table 1 shows a detailed description of these public data sets.

2.3. Classification of the articles

The articles were classified in seven categories based on the machine learning algorithms like Tree Based models, Artificial Neural Network models, Bayesian algorithms, Clustering models, Rule Induction models and the ensemble models have been applied in the field of telecommunication for predicting the potential churners. Researchers have introduced some novel features and novel techniques. Some studies have used different sampling techniques to handle the class imbalance in the customer churn data sets. Figure 4 shows the techniques applied over the last two decades in telecom churn prediction.

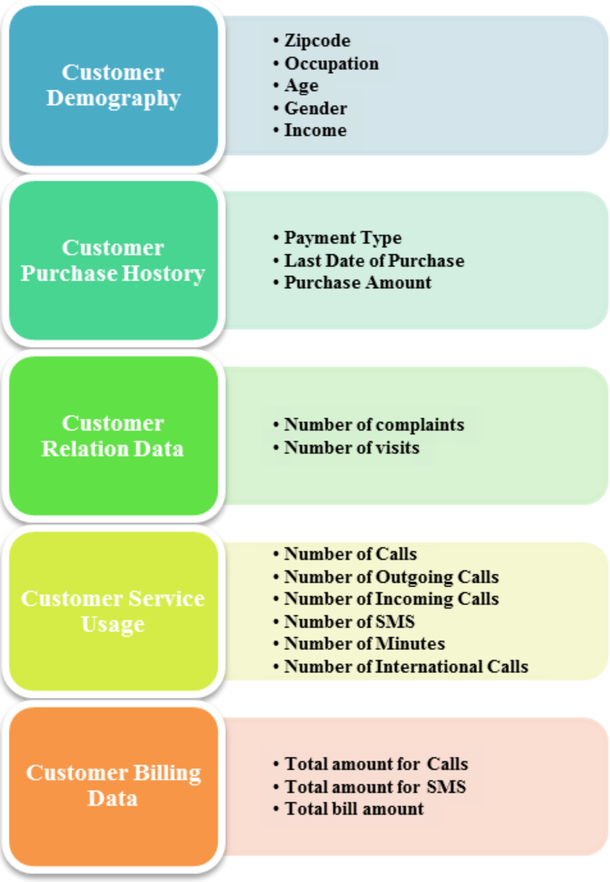


Figure 2: Customer Data used for Churn Prediction in the Telecom Industry

Table 1: Public Data Sets

Data Set	**#of Attributes**	**# of Samples**	**#of Churners**	**Features**
UCI[1]	21	5000	14.3%	State, account length, area code, international plan, voice mail plan, number of voice-mail messages, total number, minutes and charge of international calls, day calls, night calls and evening calls, number of calls to customer service
Teradata Duke[2]	171	Three datasets of 51,306, 100,000 and 100,462 respectively	1.8% per month rate	Demographics data like age, location, number and ages of children, the number of adults in the household, the education level of the customer etc. Credit score data like credit card ownership, product details like handset price, handset capabilities etc. Phone usage including number and duration of various categories of calls etc.
KDD Cup 2009[3]	242	46,933	6.98%	-

[1] https://www.sgi.com/tech/mlc/db/

[2] http://www.fuqua.duke.edu/centers/ccrm/

[3] http://kdd.org/kdd-cup/view/kdd-cup-2009/Data

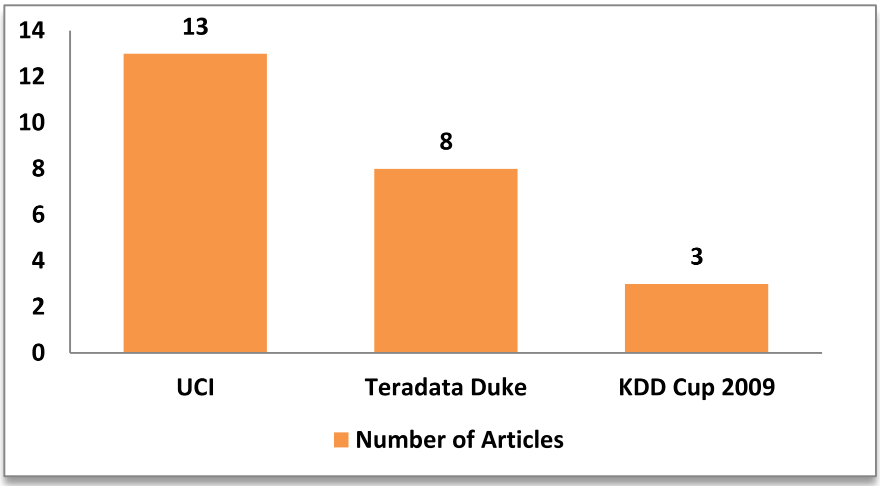


Figure 3: Public Data Sets used in Number of Articles

2.3.1. State-of-the-Art Techniques

Bayesian Models:

The traditional models that were used by researchers include decision tree models, support vector machines and neural networks. In 2003, S. V. Nath and R. S. Behara [1] analyzed the wireless industry customer churners by using the traditional Bayesian classifier: Naïve Bayesian (NB) on the Teradata data set provided by Duke University. The NB classifier gave 68% accuracy on the data set. P. Kisioglu and Y. I. Topcu [5] used Bayesian Belief Network (BBN) to predict the possible churners in a Turkish telecom company and CHAID (Chi-squared Automatic Interaction Detector) algorithm to discretize nominal data. BBN found frequency of calls, average call minutes and billing amount as the most relevant features to predict churn.

Support Vector Machines:

Apart from Bayesian models, the two traditional models Support Vector Machines (SVM) and Artificial Neural Networks (ANN) have been quite popular. X. Guo-en and J. Wei-dong [2] used SVM and compared it with other models to get the highest accuracy of 0.9088. They experimented SVM with different kernel functions and concluded that Radial Basis kernel function gave the best results. On the other hand, the SVM classifier gives a good performance under certain conditions like large amount of data samples, big churn rate, few missing values, and nonlinear data. In 2013, I. Brandusoiu and G. Toderean [6] proposed a predictive model for churn prediction using SVM based on four kernel functions: Radial Basis Function kernel, Linear kernel, Polynomial kernel and Sigmoid kernel. The SVM method used depends on a subset of the training data samples that are near the class boundaries. The SVM classifier needs an equal distribution of the class variable to give a good performance. Hence boosting was performed to get an equal number of churners and non-churners. The polynomial kernel gave the best result with 88.56% accuracy on the test set while sigmoid kernel performed the worst. The proposed strategy is good for only predicting 80% of the potential churners. Y. Zhao et al. [7]introduced an improved one class SVM model to predict churn. On comparison of different kernel functions, Gaussian Kernel gave the highest accuracy of 87.15%.

Artificial Neural Network Models:

In 2011, A. Sharma and P. Kumar Panigrahi [8] applied a neural network approach on a publicly available churn data set. They used the Feed Forward Back Propagation Neural Network, to get an accuracy of *92.35%.* The best performance was achieved with a generated model composed of a neural net having one neuron per numeric attribute for input layer, one hidden layer with three neurons and the output layer with two neurons for churners and non-churners. The proposed model is good for predicting the loyal customers but predicts only two third of the possible churners. In 2014, I. Brandusoiu and G. Toderean [9] proposed another classification model based on two types of neural networks; Multilayer Perceptron and Radial Basis Function Neural Net. The Multilayer Perceptron (MLP) has number of layers connected in series with one or more processing units, the perceptron that are connected in a feed forward way and run through layers to get to the output. The Radial Basis Function (RBF) is a feed forward neural net with an input layer, only one hidden layer called the radial basis function layer and the output layer. The accuracies of 93.7% and 90.4% were achieved with MLP and Radial basis Function neural network (RBF) respectively. They concluded that MLP performs better than RBF for predicting the churners.

Decision Tree Models:

Decision Trees are used mostly because of their high computing power, high classification accuracy and simplicity. S. Y. Hung, D. C. Yen, and H. Y. Wang [10] used Decision Trees to predict churners in a Taiwan telecom company. Experiments were performed using Decision Trees with and without segmentation. The latter gave better results with the highest lift. Oseman et al. [11] discussed data mining approaches to predict churn and performed experiments with ID3 decision tree algorithm on a telecom churn data set with features like length of service, number of minutes engaged and area. They concluded that ‘area’ was the most prominent attribute according to the ID3 algorithm. Tsai and Chen [12] reported that decision trees gave a better performance than the neural networks when applied on a churn data set for a Taiwan telecom company that offers Multimedia on demand services. The attributes included features like upload and download speed; basic and extra pay. They added an additional preprocessing step to select variables with association rules and got an accuracy of 90.98% on the validation set.

On the contrary, M. Owczarczuk [13]reported decision trees as being unstable for predicting the prepaid churners of a Polish telecom company. The author used lift curves to predict the churners with linear models such as logistic regression and compared them to decision trees. He concluded that for the cheap telecom services offered by the quickly evolving Poland telecom industry, logistic regression was more stable. Table 2 shows a description of the data sets used and the results achieved in the telecommunication churn prediction.

2.3.2. Rule Induction Techniques

Certain rule induction techniques have been employed by researchers over the last few years, like the traditional C4.5 and Repeated Incremental Pruning to Produce Error Reduction (RIPPER), was used by W. Verbeke et al. [14] along with some advanced rule induction techniques like AntMiner+ and Active Learning Based Approach (ALBA). AntMiner+ induces rules with synthetic ants and is based on Ant Colony Optimization (ACO). Ant Colony Optimization is inspired from the activities of a real ant colony. The ants walk through different paths in a synthetic environment where each path symbolizes a classification rule. The selected paths by the ants make up the predictive rules. The other advanced rule extraction technique ALBA that employs SVM. In ALBA class labels are replaced with SVM predicted classes and support vectors are added close to the decision boundary to incorporate active learning. These two advanced techniques were applied on a publicly available churn data set. The data set was oversampled, discretized and feature selection was performed using Chi-squared filter. Experiments were performed with different combinations of the techniques on original and oversampled data set.

Table 2: Summary of Literature for Churn Prediction

Paper	Data set	Time Interval	#of Subscribers	# of Churners	# of Attributes	Features	Prediction Method	Results
[5]	Turkish Company	From January 2008 to July 2008	2000	534 churners	23	Place of residence, Age, Tenure, Tariff type, Average billing amount, Trend in billing amount, Average minutes of usage, Average frequency of usage	Bayesian Belief Network	Average minutes of calls, average billing amount, the frequency of calls to People from different providers and tariff type are the most important variables
[2]	Data set 1: UCI, Data set 2: Home Telecommunication Carrier	Data set 2: From July to November 2006	Data set 1: 5000 Data set 2: 1474 training data 966 testing data	Data set 2: training data 622 and testing data 432 churn customers	Data set 2: 10	Voice call factor, message sending factor, and message receiving factor	Support Vector Machine using Radial basis kernel function	Data set 1: 0.9088 Accuracy rate Data set 2: 0.5963 Accuracy rate

[10]	Wireless telecom company offering MOD services in Taiwan	From July 2001 to June 2002	160,000	14,000 churners	10	Age, Tenure, Gender, Monthly fee, Billing amount, Count of overdue payment, In-net call duration, Call type, MSISDN change count, Count of bar and suspend	Back Propagation Neural Net	Hit rate of 98% LIFT (10%) is about 10
[12]	Telecom company in Taiwan	From 2005/10/01 to 2006/12/21	37,882	-	22	Customer personal information, MOD service usage information	Decision Tree	96.67% Accuracy
[13]	Polish mobile operator	2007 and 2008	85,274	-	1381	Call Detail Records, tariffs and components	Logistic, Linear regression	Linear models, especially Logistic regression, are a very good choice when modeling churn of the prepaid clients
[14]	UCI	3 months	5000	14.3%	21	Call Detail Records	RIPPER	93.87 average percentage correctly classified

								(PCC)
[15]	European Company	-	100.205	2.983	73	-	Logistic Regression	0.6790 AUC
[16]	American telecom company	From July 2001 to January 2002.	51,306	34,761	-	CRM data	ANN	Two hybrid models outperform the Single neural network baseline model
[17]	Teradata Duke	July to December 2001	Three datasets of 51,306, 100,000 and 100,462	-	171	Mean unrounded minutes of customer care calls, the number of Adults in the household, the education level etc.	Boosted CART	Bagging and boosting provide better performance
[18]	UCI	3 months	5000	14.3%	21	Call Detail Records	Boosted Support Vector Machine	96.85% Accuracy
[19]	East Asian Mobile Operator	-	2180	3.21%	15	-	Boosted Decision Tree ADT	97.4 AUC
[20]	Teradata Duke	July to December 2001	Three datasets of 51,306, 100,000 and 100,462	-	171	Mean unrounded minutes of customer care calls, the number of Adults in the household, the education level etc.	Holdout SVM	94.13 AUC

[21]	Ireland Telecom company	-	18,600	5000	84	Number of calls, duration, fees, the changed number of calls, changed duration, changed fees, the rates of the increased number of calls, the rates of the increased duration and the rates of the increased fees etc.	Decision Tree C4.5	90.23% Overall accuracy

The best performance of 93.87% accuracy was achieved with the combination of ALBA and RIPPER with extra pruning, in terms of average percentage correctly classified (PCC) on the original data set.

ecently four rule generation methods Exhaustive, Genetic, Covering and LEM2 were used by A. Amin et al. [22] one by one to extract decision rules. These rules were used with Rough Set Theory and results were compared with traditional state-of-the-art classifiers to get an accuracy of 0.981. The rule generation methods (Exhaustive, Genetic, Covering and LEM2) when compared with the traditional classifiers gave the best performance.

2.3.3. Sampling Techniques

Churn data sets typically have a class imbalance issue, with the majority of customers being non-churners or loyal customers and a few among them are the churners. To resolve this imbalance, studies present different sampling techniques like oversampling or under-sampling. L. Peng et al.[23]introduced two re-sampling techniques: Sampling Based Clustering (SBC) and Sampling Based Cluster Boundary (SBCB). Sampling Based Clustering clusters the samples first, then resampling is performed by choosing a sample from every cluster. In Sampling Based Cluster Boundary, the boundary is found by two cluster density thresholds. One threshold is used to cluster the data samples further and is based on features and the mean distance. The other threshold is used to find the objects of cluster boundary. These boundaries are used to train the classifier in SBCB. Authors found that when SBCB was applied on a churn data set and classified using SVM classifier it gave an AUC of 0.9053. Thus SBCB proved to outperform SBC.

Some interesting results were reported by J. Burez and D. Van den Poel [15] when they applied basic under-sampling and advanced sampling technique CUBE on a churn data set. CUBE had no effect on the data set while under-sampling improved accuracy using AUC evaluation metric with both Logistic Regression (LR) and Random Forest (RF) classifiers.

2.3.4. Ensemble Techniques

Ensemble Techniques have been quite popular due to the fact that combination of two learners is better than a single classifier which in turn improves the predictive performance of a learner. In the telecom industry, ensemble classifiers have been used to predict the potential churners by using standard ensemble learners. Mozer et al. [24]explored techniques including Logistic Regression, Decision Trees, Boosting and Neural Network on US telecom company data set. Instead of using a single neural network, an ensemble of the model was used along with Adaptive Boosting (AdaBoost). C. Tsai and Y. Lu [16]used a similar technique and proposed a hybrid neural network model, which combines two neural networks. The first neural network performs data reduction and the second predicts the churners. They got an accuracy of 94% on the American telecom company data set with this hybrid model. Another hybrid predictive model [25]was proposed in which bagging, boosting and LOLIMOT algorithm were combined using ordered weighted averaging (OWA) technique on the Duke University Teradata Center telecom data set to get the highest top decile lift of 2.41 by this approach.

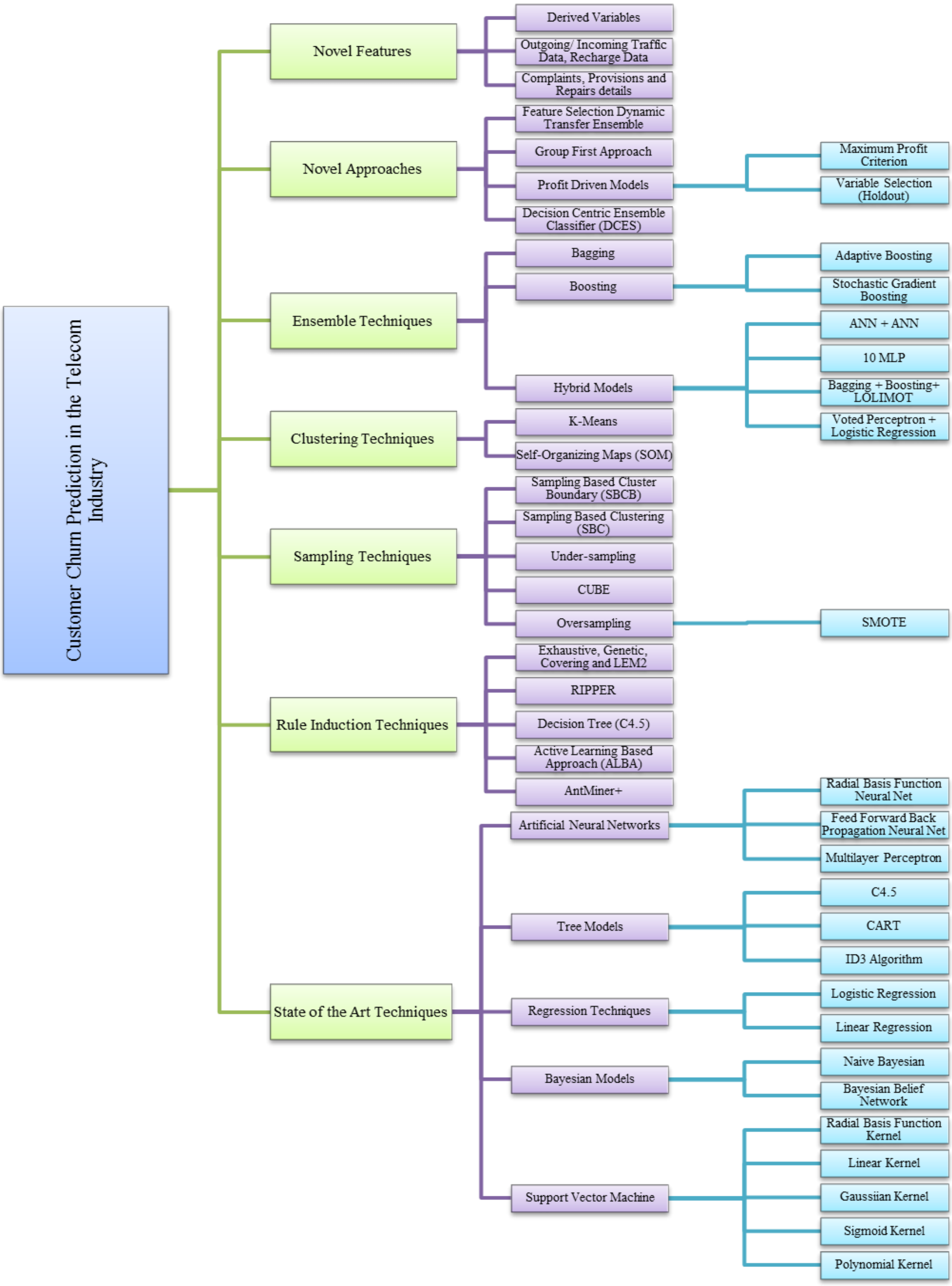

Customer Churn Prediction in the Telecom Industry
Novel Features
Derived Variables
Outgoing/ Incoming Traffic Data, Recharge Data
Complaints, Provisions and Repairs details
Novel Approaches
Feature Selection Dynamic Transfer Ensemble
Group First Approach
Profit Driven Models
Maximum Profit Criterion
Variable Selection (Holdout)
Decision Centric Ensemble Classifier (DCES)
Ensemble Techniques
Bagging
Boosting
Adaptive Boosting
Stochastic Gradient Boosting
Hybrid Models
ANN + ANN
10 MLP
Bagging + Boosting+ LOLIMOT
Voted Perceptron + Logistic Regression
Clustering Techniques
K-Means
Self-Organizing Maps (SOM)
Sampling Techniques
Sampling Based Cluster Boundary (SBCB)
Sampling Based Clustering (SBC)
Under-sampling
CUBE
Oversampling
SMOTE
Rule Induction Techniques
Exhaustive, Genetic, Covering and LEM2
RIPPER
Decision Tree (C4.5)
Active Learning Based Approach (ALBA)
AntMiner+
State of the Art Techniques
Artificial Neural Networks
Radial Basis Function Neural Net
Feed Forward Back Propagation Neural Net
Multilayer Perceptron
Tree Models
C4.5
CART
ID3 Algorithm
Regression Techniques
Logistic Regression
Linear Regression
Bayesian Models
Naive Bayesian
Bayesian Belief Network
Support Vector Machine
Radial Basis Function Kernel
Linear Kernel
Gaussiian Kernel
Sigmoid Kernel
Polynomial Kernel

Figure 4: Techniques applied in churn prediction between 2000 and 2015

Bagging and Boosting that were applied on the Classification and Regression Tree (CART) by A. Lemmens and C. Croux [17]. Both ensemble techniques were used to upgrade the performance of the base or weak learner. It was found that Bagging and Boosting outperform the basic decision tree CART after a few iterations while evaluating with measures like top decile and gini coefficient. Stochastic Gradient Boosting classifier gave the best results. The authors conclude that boosting is much better than bagging, though the outcome depends on the data set under experimentation. In 2015, T. Vafeiadis et al. [18] compared boosted (using Adaptive Boosting AdaBoost) and non-boosted versions of the traditional classifiers like Support Vector Machine (SVM), Decision Tree (DT) and Back-Propagation Network (BPN) to investigate which gives better performance. On a public churn data set, they created 100 Monte Carlo simulations for the parameter scenarios for every classifier (SVM, BPN and DT) and achieved an accuracy of 96.86%. Boosted Support Vector Machine (SVM-POLY kernel using AdaBoost) proved to be the best technique. They concluded that when boosting technique was applied on traditional classifier methods, it gave a higher accuracy.

C. Phua et al. [26]experimented with the tree based models and bagging on an under-sampled churn data set and concluded that Random Forest performed well among J48, Simple Cart models while Decision Stump gave the highest precision of 93.4% to predict the near future churners.

2.3.5. Novel Approaches

Some researches introduced a number of unique approaches to predict churn in the telecommunication industry. Y. Richter et al.[27]proposed a Group First Approach, in which the social group's impact is considered as a significant part in the churn activity of the customers. They used Decision trees that were applied on a real life data set to predict churners within 14 days. However they used only call details without any personal information to predict such churners.

As retaining a subscriber is less expensive then seeking for new subscribers for a carrier, W. Verbeke et al. [19] propose a profit driven model where they employ Maximum Profit Criterion to target only the top 10% of the customers to save expenses. They compared twenty-one different classification techniques and tested the impact of input selection and oversampling on eleven real life wireless telecom data sets. All experiments were conducted once with and without the selection of attributes or oversampling with each classifier and evaluated using profit criterion as well as statistical measures like AUC and top decile lift. The top decile lift is chosen to limit and target only the priority customers. Oversampling was applied using Synthetic Minority Over-Sampling Technique (SMOTE), which creates synthetic or artificial samples of the minority class from the original data set based on the nearest neighbors. The input selection was based on a wrapper approach where first all redundant data was filtered by selecting attributes with the highest Fisher score. The authors believed that input selection and the choice of the classification technique plays a vital role to achieve good performance while oversampling can have positive or negative effect depending on the data set and the classification technique that is used. They stated that Decision Trees gave the best overall performance and Decision Trees along with Naïve Bayesian are the easiest to operate. Another profit driven churn model was proposed by S. Maldonado et al. [20] in 2015. The proposed methods employ variable selection based on Holdout SVM (HOSVM). The embedded methods, which eliminate the features that have less impact on profit along with the classifier construction, are inspired by the Holdout SVM (HOSVM) method. They are measured using profit based metrics like Maximum Profit Criterion for Churn (MPC), Expected Maximum Profit Measure for Churn (EMPC) and H measure (H). The HOSVM method is a modification of the backward elimination method: Recursive Feature elimination (RFE), which discards the features that give the largest class separation margin, with an additional holdout step. In HOSVM, the classifier is trained on a training subset and the misclassified classes are constructed on a validation subset. The authors applied this model on a public churn data set to get an AUC of 64.6.

Researchers proposed a number of novel ensemble approaches as well. In 2014, J. Xiao et al. [28]proposed a new transfer learning approach named the *Feature Selection Based Dynamic Transfer Ensemble (FSDTE)* in which they combined Transfer Learning, Multiple Classifier Ensemble (MCE) and Group Method of Data Handling (GMDH) Type Neural Network. Their approach aims to transfer information from source to assist in modeling for the target domain. They employ Transfer Learning which assists in a learning task in a new environment by the knowledge acquired in one environment. The Multiple Classifier Ensemble *(MCE)* achieves higher classification accuracy. *FSDTE* is a three phase approach: a two layer feature selection; the first layer selects the feature by GMDH-type neural network and in the second layer further feature selection is done by making appropriate patterns. Then classification is done to get the result. The highest accuracy they got was 80.8%. In 2015, A. Baumann et al. [29]proposed a decision centric framework that targets business objectives in the model building. In their proposed framework called the Decision Centric Ensemble Selection (DCES), they used the lift maximizing candidate selection strategy and employed the popular ensemble selection approach for predictive modeling in which they choose the candidates that give the highest lift for a data set. Though their method gives better performance than the traditional models (ANN, Logistic Regression) and the standard ensemble learners (Bagging, AdaBoost) but it works on a hit and trial basis. It requires human intervention to test which candidates selected from the model library would give good results for a particular data set.

Feature selection is an important step that can have a huge impact on the predictive performance of the classifiers. Some authors introduced a couple of new feature selection approaches to solve the class imbalance in churn data sets. Y. Huang et al. [30]proposed a new filter feature selection approach based upon the dependency between the class and the attributes. The new approach as compared to the traditional feature selection approaches like Chi-Square and information gain gave the best AUC of 0.801 with Naïve Bayesian classifier on the telecom company customer data. B. Huang et al. [21]introduced a new multi-objective feature selection that was applied on an Ireland Telecom company. This feature selection approach was based on NSGA-II algorithm, modifying it to get the local sample feature subsets. The accuracy varied for different sizes of features. The highest overall accuracy of 90.23% on the whole set was achieved with Decision Tree C4.5 after applying this approach.

2.3.6. Novel Features

Generally the features used for telecom churn prediction include customer personal data, call information, complaints data and billing information. Though researches do try to find new attributes or features to predict the churners and investigate if they get some interesting results. J. Hadden et al.[31]introduced features like complaints, provisions and repairs details. The complaints data includes the type, duration of complaints, and the number of days the complaint lasted and if any money was refunded to the customer. The provisions data contains features like the estimated time it will take to resolve a complaint by the company and how many days it was delivered late. The repairs data was comprised of variables like the type of fault, duration of repairs, number of appointments and if any engineers visited the site. The authors found the complaints and repairs data had the most impact as they were picked by regression trees and neural networks.

In 2010, G. Kraljević and S. Gotovac[32] identified the features that would play a significant role in finding the potential prepaid churners. They included features like customer data, outgoing/incoming traffic data and recharge data. The customer data had attributes like the rate plan and month duration. The outgoing and incoming traffic data had features like number of SMS, number of Calls, number of Call minutes, number of service calls, number of competitor service calls. The recharge data included features like number of recharges and total amount recharged. Many models were created and compared like decision trees DT, neural net NN and Logistic Regression on a data set of 3000 samples. The highest accuracy of 91% was achieved with decision tree. U. Droftina et al. [33]aimed to find features that could determine what factors play a vital role in the churn prediction. They discovered that the ‘age’ attribute shows a significant increase in young churners and even different locations of the customers can present some interesting facts. They studied the detailed data on relationship of the subscribers to their service providers, where they noticed that changes in price plans had a huge impact on the churners.

In 2012, V. Umayaparvathi and K. Iyakutt[34] extracted derived variables from a data set with 18000 samples in the training set and 6000 samples in the test set. Derived variables are the variables that are taken out of the original data either by aggregating or subtracting some features. The data set had 252 attributes and 50% of these attributes were derived variables. The data included customer demography, bill and payment, call detail records and customer care service. Decision Tree and Neural Network were applied on this data to get 98.8% accuracy with Decision Tree. C. Kirui et al.[35]also used derived features from call details, contract related details and call pattern details. They extracted features like changes in minutes of use, frequency of use, subscriber activity, duration of calls to competitors and percentage of calls to competitors etc. After applying decision tree C4.5 and Naïve Bayesian algorithms, they compared the results of both new and old features and saw a visible improvement with the new set of features. Qureshi et al. [36]reported a significant change in accuracy with the inclusion of derived variables duration per on net incoming and outgoing calls etc. that raised the previous accuracy of 70% to 75% with the decision tree.

2.3.7. Clustering Techniques

Different researchers present different clustering strategies. While some use clustering techniques like K-Means algorithm to cluster the churners into three low, medium and high cluster categories after churn prediction like E. Shaaban et al.[37] proposed a model with six steps which involves clustering the churners into 3 categories and using the knowledge we get from this clustering to decide which churners are profitable or are a priority for retaining or even dissatisfactory. On the other hand, Y. Liu and Y. Zhuang [38] proposed to use customer segmentation with K-Means and cluster the subscribers based on the cost level into low, medium and high customer groups. Classification is performed on each group using C4.5, Logistic Regression and Neural Net. The highest accuracy achieved is 89% with C4.5.

In 2014, L. Peng et al.[39]introduced Cluster Stratified Sampling Logistic Regression Model (CSS-LRM) for churn prediction. In the proposed model, they applied K-Means clustering in layers with stratified sampling to deal with the imbalanced data in the churn data sets. Using K-Means to make random clustered samples of the data set and then applying Logistic Regression (LR), they claim to solve class imbalance problem. With the amalgamation of the clustered stratified sampling and LR, they achieved an AUC of 0.914.

Table 3 shows a distribution of the papers in which the techniques have been used. The ensemble classifiers like bagging or boosting have been used by only a hand full of researchers regardless of the fact that they tend to improve the classification performance. Hence the ensemble classifiers need to be further explored for the telecom churn prediction.

Table 3: Distribution of Papers by Techniques

Techniques	**Number of Papers**
Support Vector Machine	7
Neural Network	7
Decision Trees	12
Regression	6
Bayesian Network Classifier	3
K-Means	3
Rule Induction	2
Random Forest	2
Self-Organizing Map	1
Bagging	3
Boosting	4

Chapter 3

ENSEMBLE CLASSIFICATION

3.1. Ensemble Classification

Researchers have proposed techniques for generation of multiple classifier systems[40, 41]. The ensemble classification process has two phases i) the ***learning phase*** and ii) the ***application phase***. The learning phase in which a set of base classifiers are generated. These base classifiers are trained on the training set to get a number of predictions. Later these predictions are combined in some way to get a single final prediction. In the application phase, a new instance without the label is given to predict the class value. Below these phases are discussed in detail and summarized in Figure 9.

3.2. Ensemble Learning

The learning phase of the ensemble classification has two phases. The first phase is ***ensemble generation*** and the second phase is ***ensemble integration***. The generated ensemble could be either homogeneous or heterogeneous. If same induction algorithms are selected, it is called *homogeneous* otherwise *heterogeneous*. The second phase is ensemble integration. The ensemble integration can be either done with *selection* or *combination*. The combination approach combines the predictions of different models to get a final prediction. The selection approach can be *static* or *dynamic*. The static selection approach selects one best model on the whole data based on the prediction performance. The dynamic selection considers the characteristics of the new instance and selects the best performance on the validation set, only based on the selected model. One best selection approach is the *Cross Validation Majority* (CVM)[42]. CVM uses cross validation to get the accuracy for each base classifier and selects the one with the highest accuracy.

The most famous combination approach is the *voting method*[43]. The most sophisticated algorithm that uses the combination of classifiers is the *stacked generalization* technique[44]. The resampling techniques like *boosting*[45] and *bagging* [46]are all combination approaches. Bagging and boosting use a single learning algorithm to train different classifiers on samples of the training set and use *majority voting* to combine the classifiers.

3.2.1. Ensemble Generation

In ensemble generation, the goal is to get a set of models $L = 1 \ldots N$. They can have the same induction algorithms (homogeneous) or have different induction algorithms (heterogeneous). Heterogeneous ensemble has more diversity as compared to homogeneous as the base learners are different. Thus diversity can be controlled in homogeneous ensemble while there is a lack of control over diversity in heterogeneous ensemble. Heterogeneous ensemble is believed to be more accurate as compared to homogeneous, as it leads to the reduction of the ensemble variance[47]. The ensemble of individual learners has small correlation and thus the variance is small. Different induction algorithms are also used with different parameter sets in a heterogeneous ensemble [48, 49]. Heterogeneous ensembles can have a homogeneous ensemble as the base learner. One common method for making a homogeneous ensemble is by using sampling techniques and making repeated samples of the learning set. These sampling techniques include Bagging and Boosting.

Generally the ensemble generation process involves *Data Manipulation*. The data is manipulated either by taking subsamples to get the model either by *Subsampling from the training set, Manipulating the Input Feature and Manipulating the Output Variable.* The Subsampling method generates models using different subsamples and assumes the model is unstable. In ensembles, the base learning algorithms are sensitive to variations that is why mostly Decision Trees, Neural Nets are used which are unstable [50]. The instability of the base learning algorithm proves that the ensemble has properties of accuracy and diversity. The popular methods used for such manipulation include Bagging, Sub-bagging and Boosting. The *Manipulation of Input Feature*s has two approaches. One way is to use a subset of features from all the attributes. The second is obtained by applying some transformation to the original attributes. The *Manipulation of Output Variable* involves getting different training sets by making some transformation in the output variable. One such method proposed is Output Smearing [51]. Rather than manipulating the data, one other approach is to manipulate the model process either by manipulating the parameter sets, the induction algorithms or the model itself.

3.2.2. Ensemble Pruning

Once the ensemble is generated, the model needs to be pruned [52]. The subset of models is selected from the pool of models to get the final ensemble. The *"overproduce and choose approach"* [53] can be used for ensemble pruning. Pruning method can cut costs and improve accuracy as well. Model can be pruned by using a i) partitioning method or ii) search method. Partitioning method includes the use of algorithms like the clustering algorithms. The search method uses the hit and trial approach to test the model by removing a candidate from the subset of models. Partition-based methods assume that the pool of models contains similar models. Then the models are divided into subgroups by some clustering algorithm. K-Means algorithm can be used for such purpose and the value of *K* should be tested by running it multiple times for different values[54]. Search based methods include i) exponential, ii) randomized or iii) sequential methods. Exponential algorithms search the complete input space to select the sub sets from the pool of models. Randomized algorithms like evolutionary algorithms perform a heuristic search to look for the final subset. Sequential algorithms[55] perform backward, forward or combined search by iteratively adding or removing models.

3.2.2.1. Overproduce and Choose Approach

Overproduction and choose approach [56, 57] can be used for final ensemble creation, as seen in Figure 5. For the *ensemble overproduction* phase, bagging and boosting techniques can be used. These techniques manipulate the data set, creating a number of models. The different classifiers can be designed using these techniques and by changing the parameters, the classifier types or even the classifier architectures. In the *ensemble choosing* phase, a subset of classifiers is selected which is combined to give the best optimal accuracy. The subset can be obtained by exhaustive enumeration. This exhaustive enumeration is performed on the validation set to get the accuracy for all possible subsets. Then the subset with the best performance is chosen. The performance of the subset is subjective to the selected combination function like the majority vote. Let the size of the set produced by overproduction phase be *M*. Then we get the following number of subsets:

$$\sum_{i=1}^{M} \binom{M}{i}$$

The overproduction and choose approach is used to create a set of ensembles made up of *M* classifiers.

$$L = \{L_1, L_2, L_3, \ldots, L_M\}$$

Next the subset L' with the optimal accuracy is chosen. There are some options available for choosing the right subset. These options include methods based on heuristic rules that were proposed by Partridge and Yates [58] or the search based methods.

The *Heuristic or Randomized method* includes either "*Choosing the best*" or "*Choosing the best in the class*". In the "*Choose the best*" method, the subset is formed by selecting the classifiers from the set with the highest classification accuracy. With this method the classifiers show the same error diversity. The selection of classifiers is based on the accuracy value. The "*Choose the best in the class*" method chooses the classifier with the highest accuracy for each classifier class. This method considers that the different types of classifiers would be more error independent than the classifiers of same type. This in turn reduces the computational complexity of the ensemble.

The *search or sequential based* methods are mostly applied on validation sets to avoid over fitting problems. They can be exhaustive, forward or backward search. The *exhaustive* search is a natural way of selecting the best subset but assuming that the candidate set is very small, as proposed by Sharkey et al.[53]. In *forward* search, the initial learner is selected to form the ensemble of a single classifier. This selection is based on either the highest accuracy measure or random selection. Then another classifier is added to form subsets with an ensemble of two classifiers. From both the subsets, the one with the highest accuracy is chosen. Now another classifier is added to make an ensemble of three classifiers. The classifiers are added until the evaluation measure like accuracy, lowers with the addition of the classifier; like for a subset with size n, the ensemble has more accuracy than the subset with size n+1. The subset of size n that had the highest value is selected as the final ensemble. In *backward* search, the full set of classifiers with size n is selected. Then one classifier is eliminated which in turn leaves the remaining classifiers of size n-1. All the possible subsets are generated from these classifiers and the subset with the highest accuracy is selected. Then a classifier is eliminated, which leaves a subset of size n-2. This process stops when the accuracy or other evaluation measure from the subset of size n lowers than the obtained with the subset of size n+1. The subset of size n+1 that had the highest value is selected as the final ensemble. The forward and backward search stops when the accuracy lowers, whereas the *tabu* search keeps on going even if the evaluation measure lowers. It implements both the forward and the backward search. It starts with the full set of classifiers and then at each step adds or subtracts one to get new subsets. The one with the highest accuracy is chosen to form the new subsets. The classifier that has been deleted or added in the previous steps cannot be used for deletion or addition in a certain number of search steps.

In clustering method [59], the classifiers are clustered based on some diversity measure. The ones with large number of errors are grouped in one cluster and the others with small errors are grouped in other clusters. While iterating, one candidate is selected from each cluster to form an ensemble. The ensemble is combined with majority voting and the one with the highest accuracy is chosen.

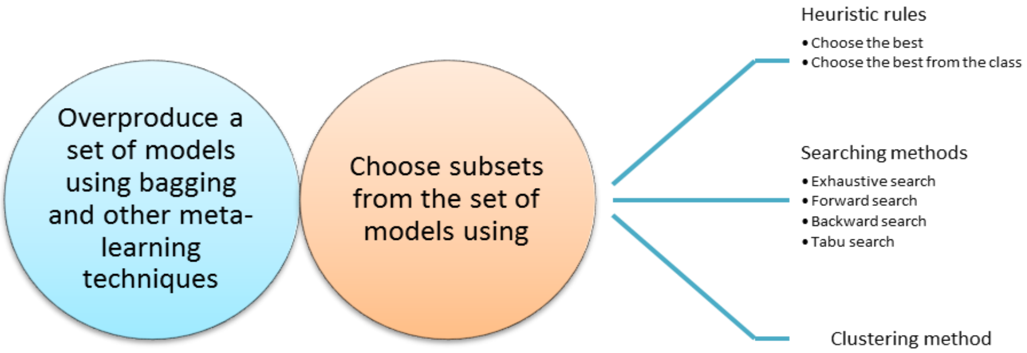

Overproduce a set of models using bagging and other meta-learning techniques
Choose subsets from the set of models using
Heuristic rules
• Choose the best
• Choose the best from the class
Searching methods
• Exhaustive search
• Forward search
• Backward search
• Tabu search
Clustering method

Figure 5: Overproduce and Choose Approach

3.2.3. Ensemble Integration

Once the ensemble is pruned by selecting the right amount of models, the classifiers need to be integrated. The classifiers can be integrated in three different categories i) *cascading* ii) *parallel* and iii) *hierarchical*. The cascading classifier inputs the second classifier the output of the first classifier and so on. The disadvantage of the cascading classifier is the inability of the later classifiers to correct the predictions of the early classifiers. Parallel classifiers integrate the learners in a single location. The decision that needs to be made in this category is to select the appropriate combination method. Hierarchical classifiers are a hybrid of the cascading and the parallel classifiers. All three categories are shown in Figure 6, Figure 7 and Figure 8.

Once the right topology is selected, the classifiers are integrated either by combining or selecting the predictions to get the final answer. There is a set of classifiers, each with their own hypothesis. Now these hypotheses are combined using a proper combination function. These combination methods can be categorized as linear or non-linear. If the linear functions like (sum and product) are used then they are categorized as linear combination methods whereas if rank based methods are used they are the non-linear combination methods. The combination method must be effective to handle the correlation among the models. Normally some weight is assigned to each model. Some weighting schemes require the weight to be greater than zero while other need it to be equal to one. There are two types of weighting methods[60]. One that fixes the weights at the end of training is called the constant weighting function. Method that varies the weight according to the example currently being predicted is the non-constant weighting function. In Weighted majority scheme, the class that receives the most votes is the final prediction. The simplest way to combine predictions of models is the “majority vote”. Other methods include “weighted average vote”, “stacking until convergence” and “belief integration”.

There is a set of different combination schemes available that can be used for ensemble integration. The famous among them is Bagging, Boosting and Stacking. In the next section we discuss the different models and combination schemes used for this study.

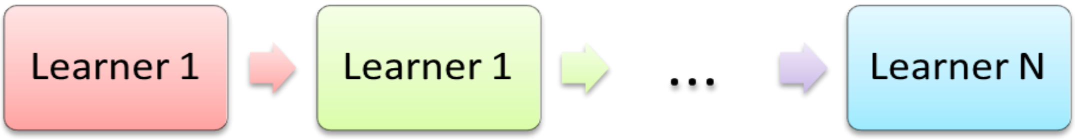


Figure 6: Cascading Classifiers

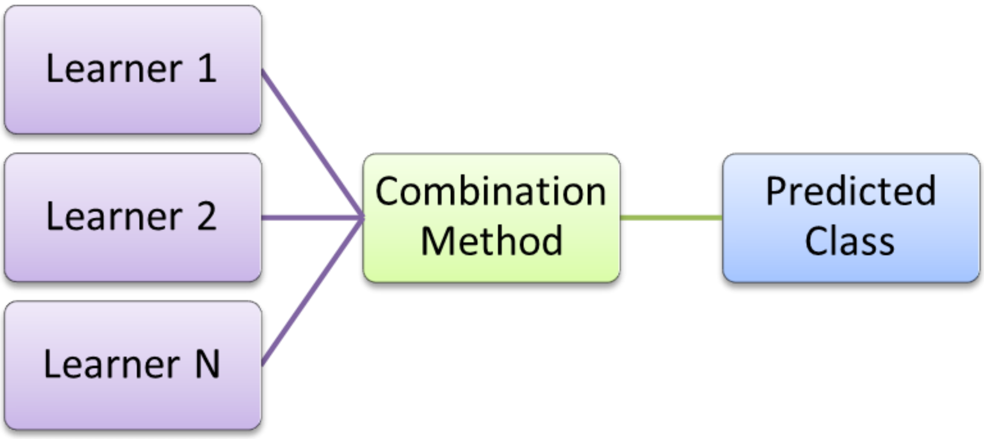


Figure 7: Parallel Classifier

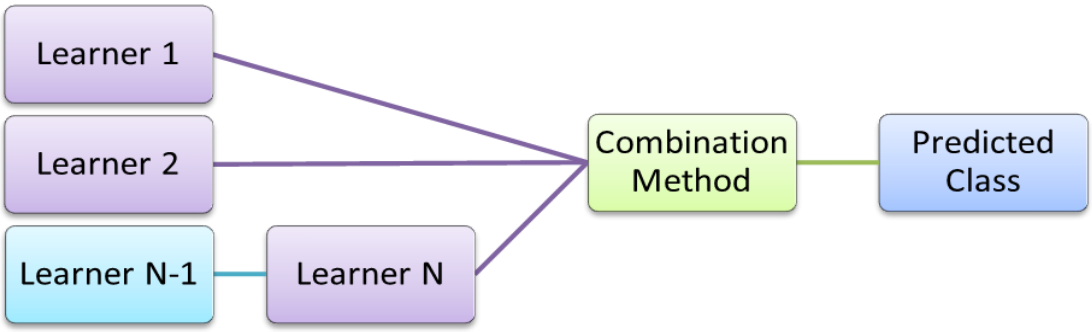


Figure 8: Hierarchical Classifier

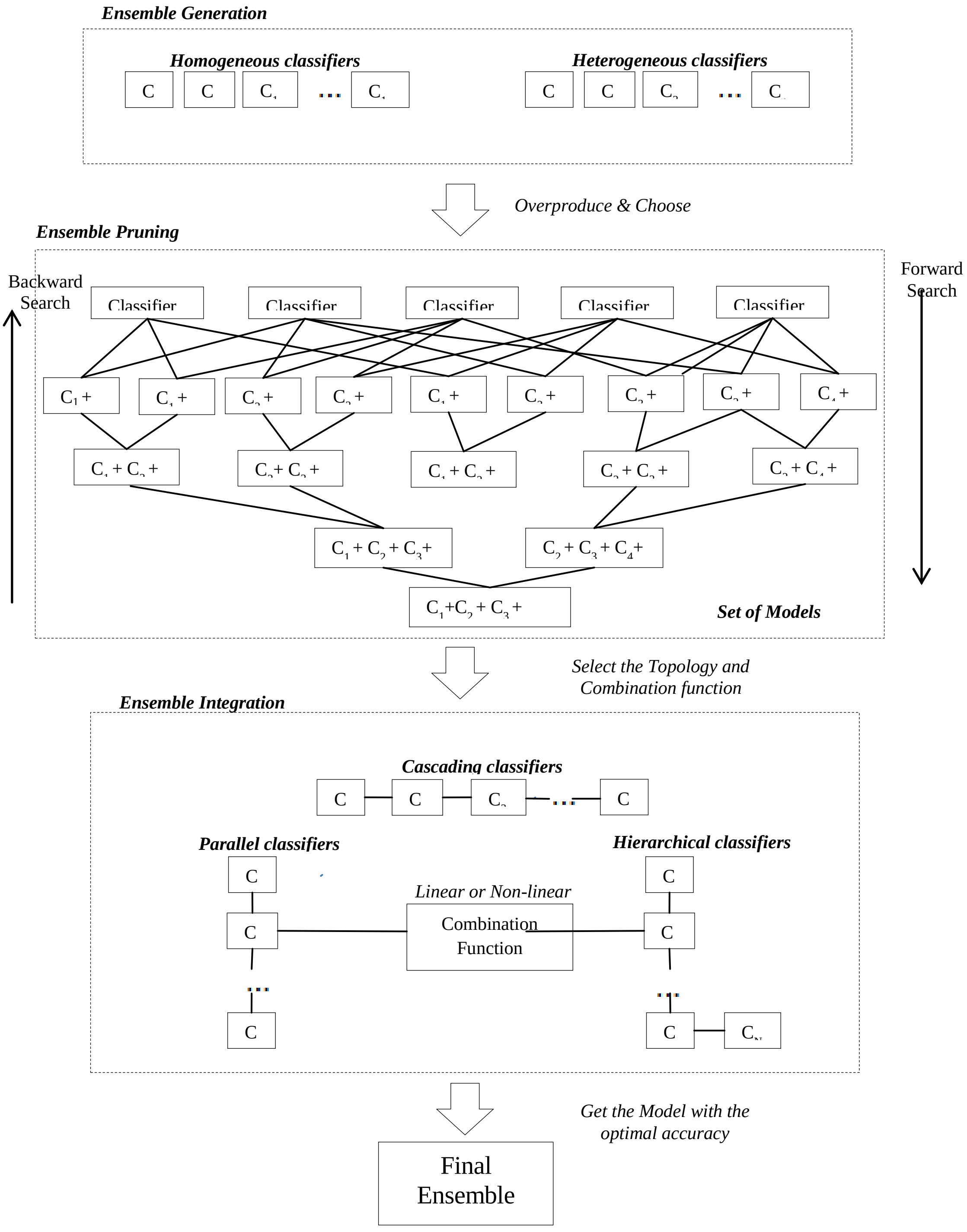


Figure 9: Ensemble Classification

Chapter 4

MACHINE LEARNING TECHNIQUES

4. Models

In this chapter, a brief overview of the models used for ensemble building and the training procedure is given.

4.1. Base Classifiers

The popular classification algorithms used for the training procedure based on their diverse behavior are listed below.

4.1.1. K Nearest Neighbor

Nearest-Neighbor classifiers were introduced by Fix and Hodges in 1951[61]. K-Nearest Neighbor [62] is a type of instance based learner which classifies the output by taking a majority vote of its neighbors. An instance based learner [63] classifies an instance by comparing it to a collection of pre-classified samples. It assumes that similar instances have similar classification. The distance function determines how similar two instances are and the classification function specifies how the similarities between two instances can get a final classification for a new instance. The class which is dominant among the nearest neighbors' predictions is selected as the classification for the new instance. During classification, an unknown instance or tuple is classified by searching for the pattern space for the *K* training tuples that are closest to that unknown instance. This closeness is defined by the distance metric such as the Euclidean distance. This distance between two points *X* and *Y* is given by. Let us suppose $X = \{x_1, x_2, x_3, \cdots, x_n\}\ and\ Y = \{y_1, y_2, y_3, \cdots, y_n\}$, then distance is:

$$dist(X, Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

However other distance measures like the Manhattan distance may also be used. The distance measure assumes that the attributes are numeric. For nominal attributes like color, the difference is *0* if both the data points *X* and *Y* have the same color red. If *X* has red color and *Y* has yellow color, the difference is 1. If the values are missing then the maximum possible difference is taken. The value of k can have an impact on the prediction value. Thus to determine the good value for k, the initial value of 1 is taken i.e. k=1 and the error rate of the classifier is tested. The value of k is incremented until the minimum value for the classifier is reached.

4.1.2. Logistic Regression

Logistic Regression[64] is a popular statistics based classification technique. It is like the linear regression classifier but suitable for binary dependable attributes (like 1 or 0 and yes or no) and not for continuous attributes. The response function makes sure that the value of the dependable variables falls between zero and one. The value predicted is the probability of an event in the range of zero and one. Maximum likelihood (ML) is mostly chosen as the method for parameter estimation in logistic regression[65]. The logistic function is:

$$\pi'(X) = \frac{1}{1 + e^{-x}}$$

And when there are multiple predictor variables, given by $X = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_n x_n$, then above function becomes:

$$\pi'(X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_n x_n)}}$$

Where X is the input or the predictor variable and $\pi'(X)$ is the estimated output. β_0 is the intercept and $\beta_1, \beta_2, \beta_3, \dots, \beta_n$ are the regression coefficients of the predictor variables.

4.1.3. Naïve Bayesian

Naïve Bayesian is a simple probabilistic classifier that falls under the Bayes Theorem[66]. It is very computationally efficient. It assumes that the features are independent. It evaluates the relationship between the class and the feature for every instance and calculates the conditional probability of this relationship[67].

Let *D* be the training set of tuples with their associated labels. Each tuple is represented by a set of attributes $X = \{x_1, x_2, \cdots, x_n\}$ with *n* attributes and *C* be the particular class of an instance or data sample *X*. Suppose *H* is the hypothesis that a tuple *X* belongs to a class *C*. In classification, the probability $P(H|X)\ or\ P(C|X)$ of a sample is determined; that the hypothesis holds for an observed tuple *X*. This is the **posterior probability**, which basically determines if an instance *X* belongs to a class *C*, given we know the description of the attributes. Whereas, **prior probability** $P(C)\ or\ P(H)$ is the probability of a class or the amount of times it occurs in a data set. Prior probability is independent of the set of attributes. Suppose there are m number of classes $C_1, C_2, \cdots, C_m$. The classifier will predict for a given tuple *X* that *X* belongs to a class with the highest posterior probability conditioned on *X*. Suppose the attributes are independent known as the **class-conditional independence**, then the probability becomes the product of the probabilities of every single attribute. The probability that an instance *X* belongs to a class *C* can be computed by the following Bayes formula:

$$P(C_i|X) = \frac{P(C_i) \prod_{k=1}^{n} P(x_k \mid C_i)}{P(X)}$$

Where $\prod_{k=1}^{n} P(x_k \mid C_i) = P(x_1 \mid C_i) \times P(x_2 \mid C_i) \times P(x_3 \mid C_i) \times \cdots \times P(x_n \mid C_i)$

4.1.4. Neural Network

The artificial neural network (ANN) is a mathematical representation of a human brain[68]. The core element of the ANN is the neuron. ANN[69] has a large set of these simple nodes known as the neural cells. It has a multilayer architecture in which neurons are connected to each other with a set of links called the synapses. Each link has a synaptic weight. The neurons are placed in the layers of the network and work in parallel. The first layer in the network is the input layer. The input nodes at this layer are simply the unprocessed information that enters the network. The input layer does not perform any computations. Then we have the hidden layer. A network can have many or zero hidden layers. The hidden layer is responsible for increasing the performance of the network. The last layer is the output layer. The output layer performs calculations that give us the output for the whole network. The behavior of the output layer depends on the activity of the hidden layers. Figure 10 shows the structure of a Multi-Layer Perceptron Neural Network.

Neural Networks are usually of two types; feed-forward network or the recurrent network[70]. The feed forward network works only in one direction that is from the input to the output layer while the recurrent network can run in any direction from input to output or output to input layer. Similarly neural networks can have a single layer or multiple layers. Generally there is no default number of hidden layers or neurons. To determine the optimal number, usually a trial and error approach is used. After the selection of the number of layers and neurons in each layer, the weights need to be set. Firstly random values are assigned to the weights of the units. While processing, the signals will progress forward through each layer by a sigmoid function $f(x) = \frac{1}{(1+ e^{-x})}$. If the expected output is not obtained then the weights are modified for each layer to decrease the error.

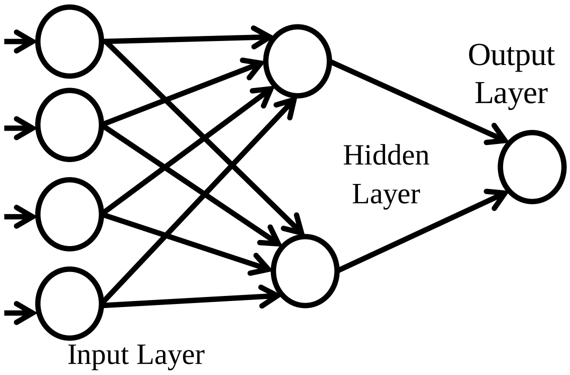


Figure 10: Multi-Layer Perceptron

4.1.5. Classification / Decision Trees

Decision Tree is a hierarchical structure of the decisions and their outcomes. They are used to identify the path for reaching a specific goal. When used for classification purpose, they are referred as classification trees. A predefined class is provided to classify an instance by these classification trees. Decision Trees are very popular because of their simplicity. It is made up of nodes and edges. The node with no incoming edges is called the “root”. The node with outgoing edges is called the “internal node”. All other nodes are called “leaves”. Usually in decision trees the internal node is split according to the value of a single attribute[71].

Splitting Criteria:

Decision trees search for the best attribute to perform the split on. There are more than one splitting criterion used for decision trees like information gain, gain ratio and gini index[72].

Information Gain:

Information Gain is an impurity based criteria. It uses entropy measure. Entropy is an information-theoretic measure of the “uncertainty” in a data set as there is more than one possibility to classify an instance. It is measured in bits of information and is defined by the formula:

$$Entropy\ or\ Info(X) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

Where $p_i \neq 0$ and p_i is the probability of an instance X that belongs to a Class $C_{i..}$ There can be $i=1 \ldots n$ classes.

The average entropy is the weight of the proportion of the original instances that are present in the training subset or subsets resulting from splitting on a specified attribute. Average Entropy is the weighted sum of the entropies of the *j* subsets, as defined by formula:

$$Average\ Entropy\ or\ Info_A(X) = \sum_{j=1}^{m} \frac{|X_j|}{|X|} Info(X_j)$$

Where *A* is the specific attribute.

So information gain is the difference between original information requirement and the new requirement after partitioning on a specified attribute, as seen in equation 3.

$$Gain(A) = Info(X) - Info_A(X)$$

or

$$Gain = Entropy - Average\ Entropy$$

Gain Ratio:

Gain Ratio outperforms information gain and normalizes it as follows:

$$Gain\ Ratio\ = \frac{Information\ Gain}{Split\ Information}$$

4.2. Ensemble or Meta Learners

4.2.1. Adaptive Boosting

Adaptive Boosting (AdaBoost) [73] was introduced by Yoav Freund and Robert E. Schapire in 1995. The boosting algorithm works with labeled training data set $D = (s_i, t_i)$ where s_i are the instances of i samples, $i=1...N$ and t_i is the associated label with every instance s_i. On every iteration $k=1...T$, a weight w_k is assigned to each sample s_i of the training data set D. The weak learner is trained to get a weak hypothesis $h_k(s_i) = t_i$.Then the learning error ε_k $\{where\ \varepsilon_k = p(h_k(s_i) \neq t_i)\}$ is calculated and the weights are updated. The weights are updated until the last iteration T is reached. The AdaBoost algorithm, adapts to the errors of the weak hypothesis that the weak learner produces, unlike its predecessors. A single prediction rule from the combination of the entire weak hypothesis is generated. AdaBoost solves over fitting problems by focusing on the misclassified examples. If the sample is misclassified, the assigned weights will increase and decrease for correctly classified samples. AdaBoost selects the most informative samples on every iteration *k*. AdaBoost converts a weak learning algorithm into a strong one[74]. Figure 11 explains the methodology used in AdaBoost.

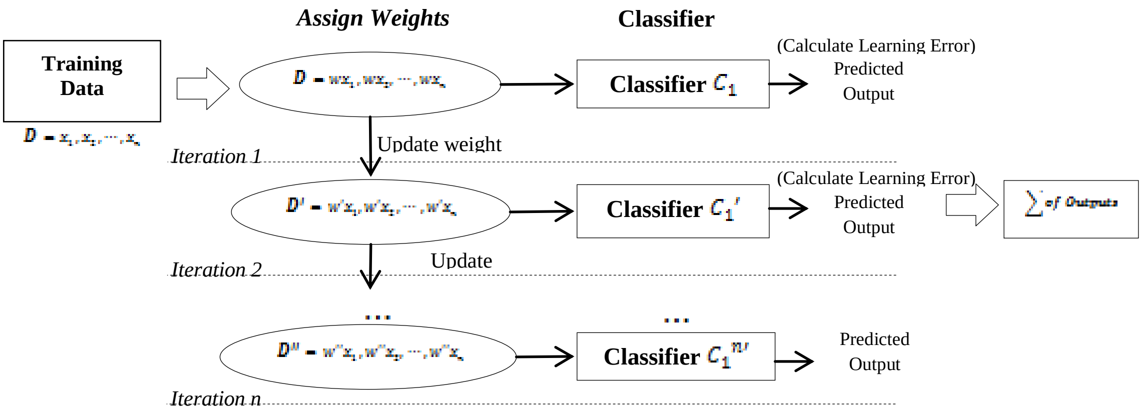


Figure 11: Adaptive Boosting

4.2.2. Stacked Generalization

Stacking or Stacked Generalization[75], proposed by David H. Wolpert, is the combination of diverse heterogeneous learning algorithms applied on a data set. This meta-model has base models referred as *level-0* learners and a *level-1* learner. *Level-1* learner combines the set of outputs of the *level-0* learners and corrects their mistakes there by improving the classification results. Stacking generates a meta-dataset, comprising the tuples of the original dataset using the predictions made by the classifiers as the input attributes. With the same target attribute for both the original and the stacked data set. This meta-model combines these output predictions of the level-0 learners into a single level-1 prediction. Figure 12 shows the architecture of stacked generalization model.

Stacked generalization is a method for reducing the error rate of one or more classifiers C_i *where* $i=1...n$. These classifiers or generalizers may include the popular algorithms like Decision Trees or Neural Networks. Stacked Generalization presumes the biasness of a classifier C_i with in the data set D, and the correct guess is gotten by using the outputs of these classifiers as input, along with the learning set. When there is more than one base classifier, it becomes a refined version of the cross validation approach.

Stacking is multi-leveled, having two levels. The first level is the space occupied by the original learning set D called the “level 0 space” and the generalizers C_i that train the learning set in this space are the “level 0 generalizers”. The first step in stacking is to partition the learning set into training D_1 and test set D_2. The outputs for the training set D_1 from the generalizers C_i make the input for the test set D_2 and are fed to the level-1 generalizer in the "level 1 space". A question in the level 0 space is passed to the level 1 space and is answered by the level 1 learning set.

Stacking is a parallel combination scheme in which the data set is divided into two disjoint sets. The base models are trained on the training set and the level 1 learner is tested on the test set. This is similar to the cross validation approach except for the fact that stacking uses a non-linear combination approach.

However one of the advantages of the stacking model is the free choice of base learners. One can use either a single classifier or multiple classifiers as they wish. Researchers have been investigating the best methods for constructing the ensemble classifiers with stacking. One such model for classification is the *Stacking with Model Trees (SMT)*[76].

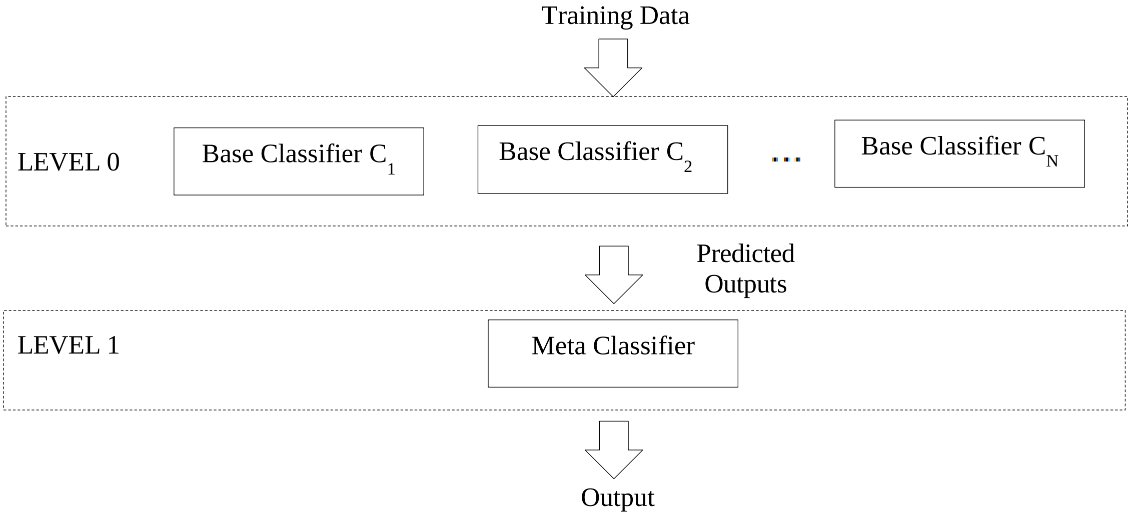


Figure 12: Stacked Generalization

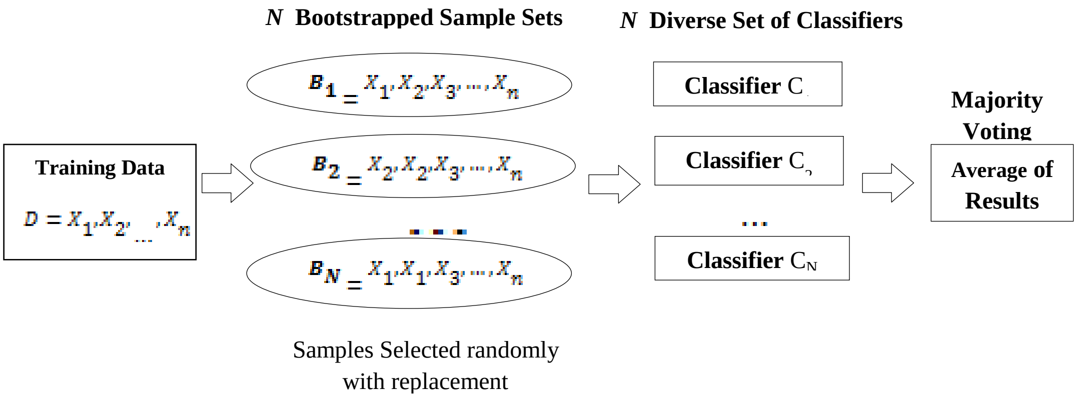
Wolpert discusses the fact that the classifier selection in the level 0 space is not clear. There is no specific generalizer that should be used for stacking. Getting the best results with stacking is still a black art as what number of base learners to select is indecisive. Some propose trees as the meta-learner while some propose a linear model. K. Ming and W. Ian suggest that stacked generalization works best with three different base learners and outperforms voting method[77]. S. Džeroski and B. Ženko also agree that the best performance was achieved with a stacking model with three base classifiers and the impact of the number of base classifiers is inconclusive as three base learners gave the same output as seven base learners[78].

4.2.3. Bagging

Bagging or *Bootstrap Aggregation*[46] is an ensemble technique used to improve the performance of a base learner. Bagging makes *N* number of bootstrap samples B_i (where $i= 1…N$) of a data set *D*. The learning algorithm is applied to each bootstrap sample B_i and the classification results are averaged in the end. Bootstrap selects samples randomly from the data set with replacement. When the process runs *n* times, *n* bootstrapped sample sets are retrieved. The probability of a sample being selected each time is 1/n, as the samples are selected randomly. In a bootstrapped sample set, a sample is repeated a number of times or not appear at all. The size of each bootstrap sample set is the same as the original training set[79].

Bagging uses bootstrap to manipulate the training data and generate different classifiers. It uses a voting scheme that decides the final decision for the class chosen by most of the classifiers for any given instance. Mostly the bagging classifier is trained using an unstable learning algorithm as it leads to creation of diverse base learner. A classifier is said to be unstable if the accuracy improves with minor changes in the training data set. Decision Trees (DT) and Neural Networks (NN) are unstable learning algorithms, as slight changes in the training data set will give different results, while with stable learners like K- Nearest Neighbor (KNN) and Linear Regression (LR) show no changes in accuracy [50].

Bagging gives interesting results with a data set of an inadequate size. A large number of the samples are drawn into each bootstrapped subset that causes distinct training subsets to overlap significantly, with the same samples appearing repeatedly in most of the bootstrapped subsets. To ensure diversity, an unstable learning algorithm is used with the aim of getting different decision boundaries and small disruptions in different training data sets. Thus NNs and DTs are good candidates. Bagging is usually applied to DT algorithms due to their instability. However the selection of free parameters can control the instability of such unstable learners. Bagging has a preprocessor that takes the bootstrap replicates of the set to give to the unstable learner and a postprocessor that aggregates the output by taking majority vote[80].Figure 13 shows the methodology of the bagging classifier.



N Bootstrapped Sample Sets
N Diverse Set of Classifiers
Training Data
$D = X_1, X_2, \ldots, X_n$
$B_1 = X_1, X_2, X_3, \ldots, X_n$
$B_2 = X_2, X_2, X_3, \ldots, X_n$
$B_N = X_1, X_1, X_3, \ldots, X_n$
Classifier C
Classifier C
...
Classifier C_N
Majority Voting
Average of Results
Samples Selected randomly with replacement

Figure 13: Bootstrap Aggregation

Chapter 5

EXPERIMENTAL STUDY

5. Experimental Study and Discussion

5.1. Data Sets

In the present study, we used two different data sets of customer relationship management (CRM) to build predictive models and assess the performance. One publicly available data set from the UCI [81] machine learning data repository that has 5000 samples and 20 attributes with 14.3% churn rate (data set 1). The second data set used is real life data provided by a major wireless telecom operator in Pakistan (data set 2).

Most of the attributes in the data sets are associated with call detail records (CDR), billing and personal information. For data set 2, the carrier provided data containing 2000 subscribers. All of these subscribers were not contract based and had a monthly based subscription. The subscriber data was extracted from the time interval of two months i.e. August and September 2015. To overcome the class imbalance problem in churn data sets, we requested an equal amount of churners and active subscribers from the carrier. So the data has 50% churner's information. A description of the data sets used is given in Table 4.

5.1.1. Input Features

Churn occurs due to the dissatisfaction of a subscriber with their present service provider. The reason for this dissatisfaction may be due to a number of reasons which may include poor service or pricing. Over the years, researchers have introduced some unique set of features in their churn prediction models. J. Hadden et al. introduced features like customer complaints, provisions and repairs information[82]. G. Kraljević and S. Gotovac used the customer recharge information like the number of recharges and the total amount recharged[83]. While C. Kirui [35] and Qureshi et al.[36] used the derived features from call detail records.

For data set 2, we used the revenue information, customer account and usage details. Some new features introduced were the information regarding the data usage (Data volume and Revenue). One other important new feature used was the favorite other network information. Below these features are discussed in detail.

Revenue Details:

Revenue is the amount of the money that the company receives during a time period, in this case, for the months of August and September 2015. The revenue generated for SMS, calls, data, the off-network, on-network and the total overall monthly revenue was collected for both the months (Aug, Sep) and then aggregated.

1. Aggregate of Total Revenue: The overall monthly revenue earned in Rupees by the carrier in the months August & September 2015.

2. Aggregate of SMS Revenue: The revenue earned through the SMS service used by the subscriber.

3. Aggregate of Data Revenue: The revenue earned through the Data service used by the subscriber.

4. Aggregate of Off Net Revenue: The revenue earned by the calls etc. made to the off-network (not the same network as the subscriber) customers by the carrier's present subscriber.

5. Aggregate of On Net Revenue: The revenue earned by the calls etc. made to the on-network (on the same network as the subscriber) customers by the carrier's present subscriber.

Subscriber's Account Details:

1. Network Age: The time passed since the subscriber started using the services of the carrier.

2. Package Name: The names of the packages the subscriber has registered. The carrier offers a number of packages. This information can help the carrier in knowing the demands of the subscriber and can have a huge impact on churn information.

3. User Type: This detail helps in knowing if the user is subscribed to a 2G or 3G service.

4. Aggregate of Complaint Count: The number of complaints made by the subscribers.

Subscriber's Usage Details:

1. Favorite Other Network: This information can certainly have a huge impact on churn ratio as it gives the information about which other network or operator the subscribers makes the most of the calls to and thus might influence the customer to move to that network to save money.

2. Aggregate of Data Volume: The volume of the data service used by the subscriber.

Table 4: Data sets used in the empirical evaluation

Data Set	Number of Attributes	Number of Subscribers	Number of Churners	Features	
				Common Categories of Features	**Other Features**
UCI (data set 1)	20	5000	14.3%	**Customer Demographic Data:** State, Area Code, **Call Detail Records:** Number of Voice-Mail Messages, Total Number, Minutes and Charge of International Calls, Day Calls, Night Calls and Evening Calls, Number of Calls to Customer Service. **Customer Account Information**: Account Length, International Plan, Voice Mail Plan	
Pakistan Telecom Company (data set 2)	13	2000	1000	**Customer Account Information**: Network Age, Package Name, Total Revenue, User type **Call Detail Records:** Calls, Complaint Count	**Revenue Details:** SMS Revenue, Data Revenue, On-net Revenue, Off-net revenue **Usage Details:** Data Volume , Favorite other network

5.2. Proposed Framework

The proposed framework is made up of a Bagged and Stacked Three Base Learner called the *Bag-Stack Ensemble Learner*. This ensemble model gathers the churned data and creates bootstrapped sample sets that form diverse set of models stacked together into a single meta model, cross validating every tenth fold of the training data. The underlying three base learners for this ensemble include the Neural Network, Decision Tree and K-Nearest Neighbor heterogeneous algorithms. The training data is subsampled into a number of sets that are given to the stacked learner and a meta decision tree model is formed predicting the correctly classified potential churners. The framework can be seen in Figure 14.

Feature selection is performed for each churn data set individually. A subset of features is selected based on the increase or decrease in the evaluation criteria. The data set may contain all attributes or a subset of the attributes. After the feature selection, the data is transformed that is compatible with the machine learning algorithms. After the preprocessing, the data is passed to the base classifiers.

The model was built uisng the ensemble generation method described in Section 3. A set of models was constructed using the meta-learning techniques i.e. Bagging, Boosting, and the Stacking technique with different number of base learners. We started with standalone baseline models to establish a baseline. These models were used in the ensemble generation as the base learners. In order to get a perfect ensemble, a proper selection of the base classifiers, integration and combination method needs to be made. The overproduction and choose approach was used, creating different set of models. These sets were created using different types of classifiers namely Neural Network, Decision Trees, Naïve Bayesian, Logistic Regression and K-Nearest Neighbor. Each classifier was used in the ensemble by changing some parameters. The pool of models is pruned to get the best model using the search based or heuristic approach and ended up with a Bagged and stacked learner with three base learners (NB, DTs, KNN) at level 0 and a Meta Decision tree at level 1. The sets generated with Bagging, Boosting and Stacking are referred as bagged-set, boosted-set and the stacked-set respectively.

The performance of each model was computed based on the ten-fold cross test procedure[42]. In this procedure, the data set is partitioned into test and training subsets. Nine of which are used as training or validation set and the tenth is the test set. This process is repeated ten times such that all subsets are used as the test set at least once. In the stacked learner itself, 60% of the data set is used for training the base model and the remaining is tested on the stacked level 1 meta-learner. This ensures that the test set is independent and has not seen the trained model. The whole training set could be used for evaluation purpose as in[84] , but this might lead to over-fitting problem. Thus the other approach could be to withhold a part of the data set for evaluation, like in [52, 85].Generally for ensemble creation, the best split ratio is 50% for training set, 25% for validation set and 25% for test set. We selected the two churn data sets, one public (data set 1) and the other private (data set 2) and randomly subdivided them into training (50%), validation (25%) and test sets (25%). The strategy we used was to train the model using the ten-fold approach and then test the final ensemble on the test set.

A conceptual diagram of how the ensemble was generated is shown in Figure 15.

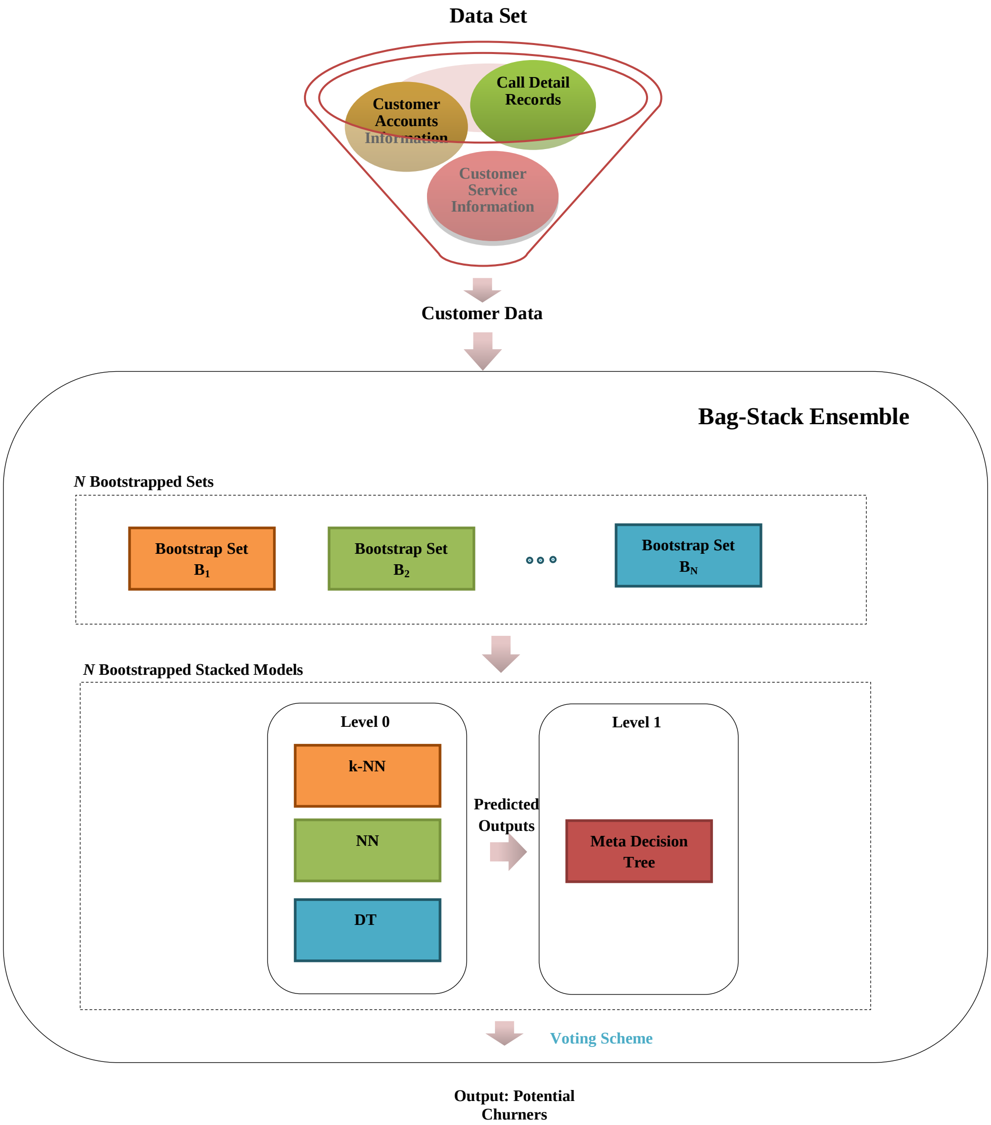


Figure 14: Bag-Stack Ensemble Learner

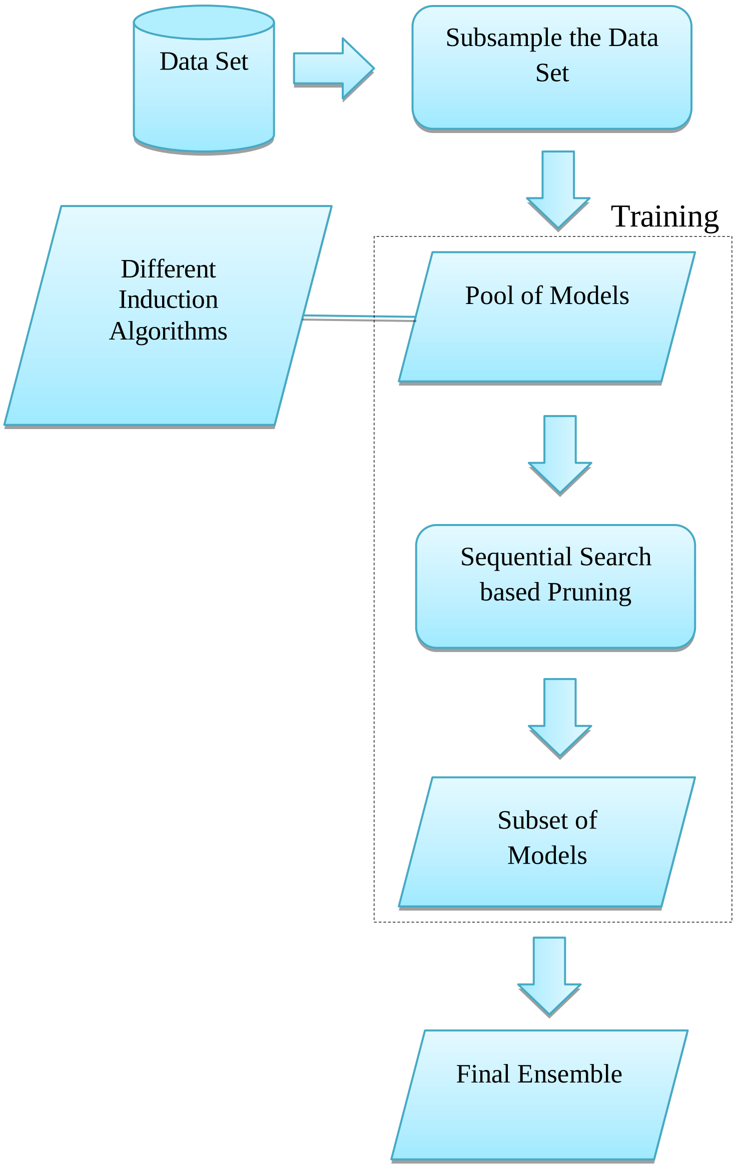


Figure 15: Ensemble Generation

5.3. Data Preprocessing

The data set was preprocessed in the form of numeric data transformation of textual attributes, feature selection and using sampling techniques. And other preprocessing tasks including class labeling, formation of training/ test sets and creation of derived variables like aggregate monthly revenue that is the sum of the monthly revenues.

Generally, the telecommunication data sets are imbalanced that results in poor prediction performance of the minority class. The learner becomes more biased towards the majority class. Sampling techniques can be utilized to handle such imbalance like oversampling[86] or under-sampling[87]. However under-sampling, discards relevant information that could play an important part in the learning process. On the other hand, oversampling can cause over fitting[88]. Thus using these sampling techniques were avoided. However, ensemble techniques require data manipulation to increase diversity, for this reason we employed the bootstrap or random sampling with replacement. Ensemble models respond well to unstable data. Thus some kind of data or process manipulation is usually performed in ensemble generation. A learning algorithm is said to be unstable if it shows a significant improvement in accuracy with slight changes in the trained dataset [89]. For unstable learning algorithms like Decision Trees (DT) and Neural Networks (NN), small changes in training data will give different results while stable learners like K- Nearest Neighbor (KNN) and Linear Regression (LR) show no changes [90]. To ensure diversity, the unstable learner is used so that different decision boundaries can be obtained for small disruptions in different training datasets. For this purpose, NNs and DTs are good candidates. The instability of such unstable learners can be controlled by the selection of the free parameters. That is why bagging is usually applied to DT algorithms.[91]. We used the subsampling technique like random sampling with replacement to manipulate the data set.

Data Transformation:

The attributes in the churn data sets contained both textual and numerical information. Thus the data was transformed to numerical data that is compatible with most of the data mining algorithms.

Feature Selection:

Some attributes were excluded for training the prediction model, like the “state” attribute in data set 1. The state attribute has all the US states. The phone number, area code attributes are also specific for US, in the UCI data set (data set 1). The goal is to make a generic model so these features can be excluded.

Random Sampling with replacement:

In ensemble models, mostly unstable algorithms are used that need diversification. A small change in the data set can bring a huge improvement in the accuracy. In order to diversify, some sort of sampling can be incorporated. One way to manipulate the data set is using the subsampling method. In our experiments, we explore the bootstrap or random sampling with replacement[79] to see how the learner performs. The samples are selected randomly from the data set with replacement.

5.4. Evaluation Measures

In order to test the model for churn prediction, we evaluated the Accuracy, Precision, Recall, F-Measure and AUC of the proposed model on a private and a public data set. The Precision and Recall values indicate the correctly classified churners. These evaluation measures [92] can be obtained as follows:

Accuracy:

Accuracy is the measure of the ratio of the correct predictions made over the total predictions. True positive and negative are denoted as TP, TN. False positive and negative are denoted as FP, FN as shown in Table 5 and explained in Table 6.

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

Precision:

Precision is the measure of the ratio of the correct positive predictions made over the total positive predictions as shown in the equation:

$$Precision = \frac{TP}{TP + FP}$$

Recall:

Recall is the measure of the ratio of the correct positive predictions made as shown in the equation:

$$Recall = \frac{TP}{TP + FN}$$

F-Measure:

F-Measure is a performance measure that is a combination or harmonic mean of Precision and Recall as seen in equation:

$$F\text{-}Measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

AUC:

Area under the ROC (Receiver Operating Characteristics) curve [93] is an alternative measure for evaluating performance of an algorithm. It is a way of measuring ranking. In a binary classification problem, the ROC curve is formed by the pairs of True Positive Rate (TPR) and False Positive Rate (FPR).

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

Figure 16 shows an example of a ROC curve. AUC is a better measure than accuracy and is a reliable measure for imbalanced data sets[94].

Table 5: Confusion Matrix

	Predicted Positive Class	Predicted Negative Class
Actual Positive Class	TP (True Positive)	FN (False Negative)
Actual Negative Class	FP (False Positive)	TN (True Negative)

Table 6: True and False cases

	Definition
True Positive (TP)	Number of correctly predicted positive cases.
False Positive (FP)	Number of incorrectly predicted negative cases as positive.
False Negative (FN)	Number of incorrectly predicted positive cases as negative.
True Negative (TN)	Number of correctly predicted negative cases.

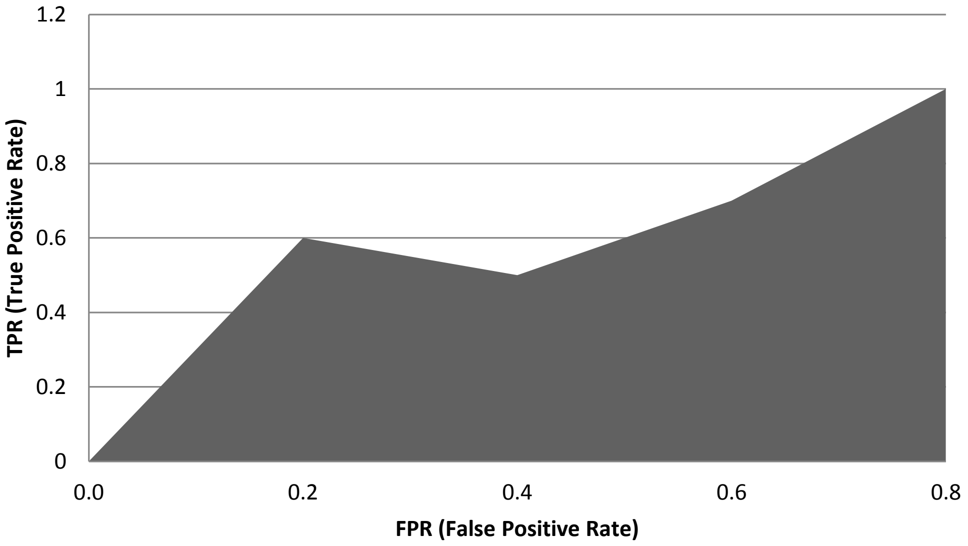


Figure 16: ROC Curve

5.5. Ensemble Building

The goal was to build a perfect ensemble model using ensemble classification method. To build the proposed framework, we started with some baseline standalone experiments, in which we experimented with some machine learning algorithms namely Naïve Bayesian, k-Nearest Neighbor, Support Vector Machine, Logistic Regression, Neural Network and different types of decision trees. As ensemble models perform much better with diverse models, a set of heterogeneous induction algorithms was selected as baseline. These diverse induction algorithms were used to form a set of models. From these models a subset was selected. This selection was based on the sequential search based method. For evaluation purpose, we used measures like Accuracy, F-Measure etc. (explained in the Section above).

Once the ensemble was pruned, the ensemble was integrated based on the combination approach. The ensemble methods like Stacking, Bagging, and Boosting use the non-linear (weighted voting average or majority vote) combination approach to assign weights to the prediction with the most votes or simply select the final prediction with the most votes. Figure 17 shows the conceptual diagram of the formation of different sets of models.

A full description of the steps involved for building the proposed framework is given below.

5.5.1. Model training

In order to select models for final ensemble we use a cross validation scheme for model training. Due to the fact that the models are initiated with different model parameters (number of nearest neighbors, type of split criteria, etc.), cross validation helps us to find proper values for these model parameters. The cross validation is done in several training rounds on different subsets of the entire training data. In every training round the data is divided in a training set and a test set. The trained models are compared by evaluating their prediction performance on the (unseen) test set. The model with the highest performance accuracy on the test set is chosen as a member of the ensemble.

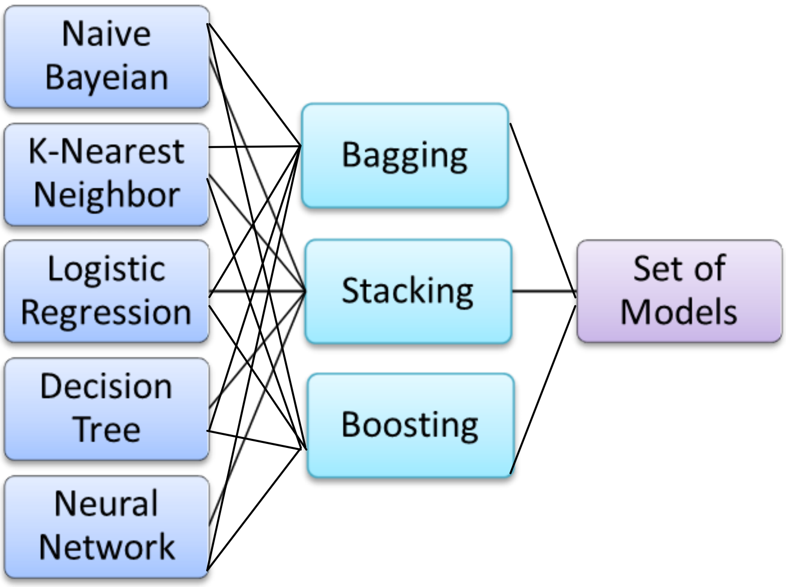


Figure 17: Conceptual Ensemble Formation

Step 1:

Step 1 was to establish a baseline. Once we had a baseline we selected the top five that performed the best on the test set. The results of the standalone experiments are shown in Table 7. As we can see that the best performance was achieved with Neural Network followed by Decision Trees. For a successful ensemble, the base classifiers selected should be diverse and should have good individual performance. Thus these top five base classifiers were selected for the final ensemble generation as seen in Figure 18.

Table 7: Results of Stand-alone Experiments

Learners	**Accuracy**	**Precision**	**Recall**	**F-Measure**	**AUC**
K-Nearest Neighbor	91.73	72.07	72.07	72.07	0.500
Neural Network	***96.27***	93.68	80.18	86.40	0.915
Decision Tree	***95.87***	90	81.08	85.31	0.897
Naïve Bayesian	88.27	61.86	54.05	57.69	0.830
Logistic Regression	85.73	58.33	12.61	20.73	0.799

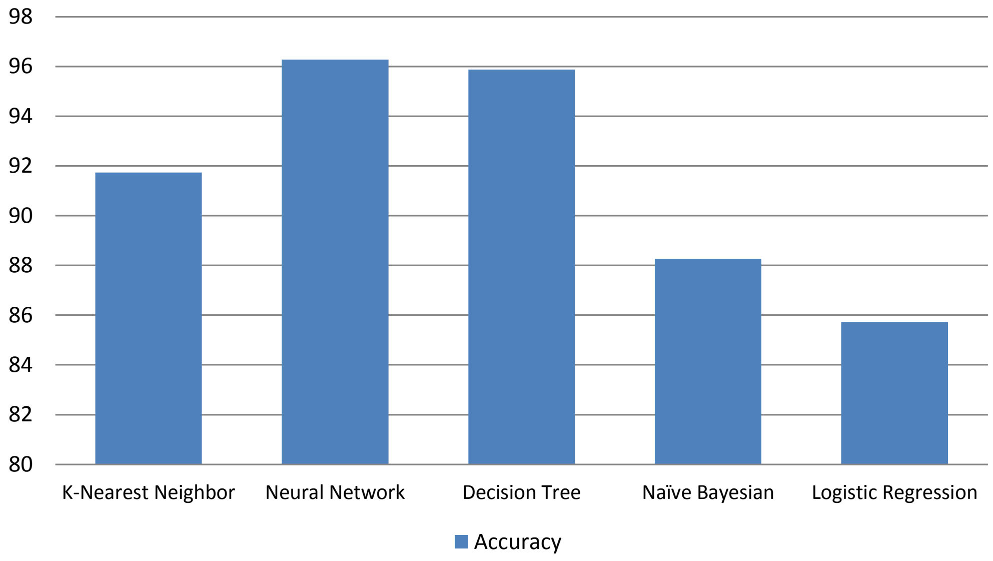


Figure 18: Result of Baseline Experiments on Test Set

Step 2:

The next step was to start assembling these base learners (NN, KNN, DT, LR, NB) using the popular meta-learners Bagging, Boosting and Stacking. The sets generated with Bagging, Boosting and Stacking are referred as bagged-set, boosted-set and the stacked-set respectively. We use the five baseline learners (taken from step 1) in the ensemble models to get the final ensemble. We get a pool of models and prune these models using search based or heuristic approach.

Table 11 reports the size and accuracy of the ensembles created with the bagged, boosted and stacked-sets.

Experiments with Boosted-set:

The boosted set gave a set of homogeneous base learners. We evaluated the performance of these boosted classifiers on the training set. The best classifier that achieved the highest accuracy in standalone experiments was used in this set of experiments and also the classifiers that were designed by different types of parameters. A model was rejected on the basis of the lowest accuracy among the different ensembles. The values reported in the ensemble refer to the validation set. For the boosted meta learner, the models that performed the best on the training set includes the

Boosted Decision Tree and the ***Boosted Neural Network*** with ***96.24%*** and ***97.04%*** accuracy respectively. The size of ensemble that gave best performance was 25 after which the accuracy stopped increasing. Figure 19 shows the relationship between the size and accuracy of the ensemble.

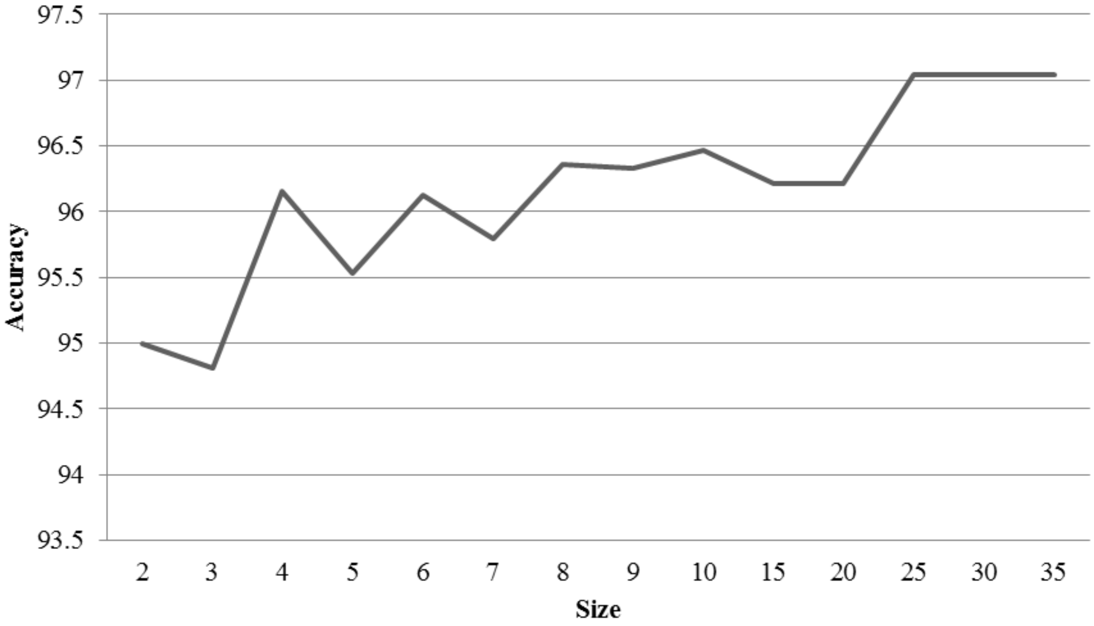


Figure 19: Experiments with Boosted set with different ensemble size

Experiments with Bagged-set:

Using bagging as a meta-learner gives a set of homogeneous learners that has the training set divided into a set of bootstrapped samples and passed to a classifier forming diverse set of models. With bagging, we experimented by forming different number of bootstrapped sets and the ratio of training set to sample from. This gave a number of bagging models formed by changing the parameters and the types of base classifiers. ***Bagged Decision Tree*** performed the best with ***96.5%*** accuracy. A small size of 8 base learners gave an accuracy of 96.5% while the accuracy decreased with increasing the size further as seen in Figure 20. Thus we stopped at the size of 8 base learners for bagging while pruning with DT homogeneous learners.

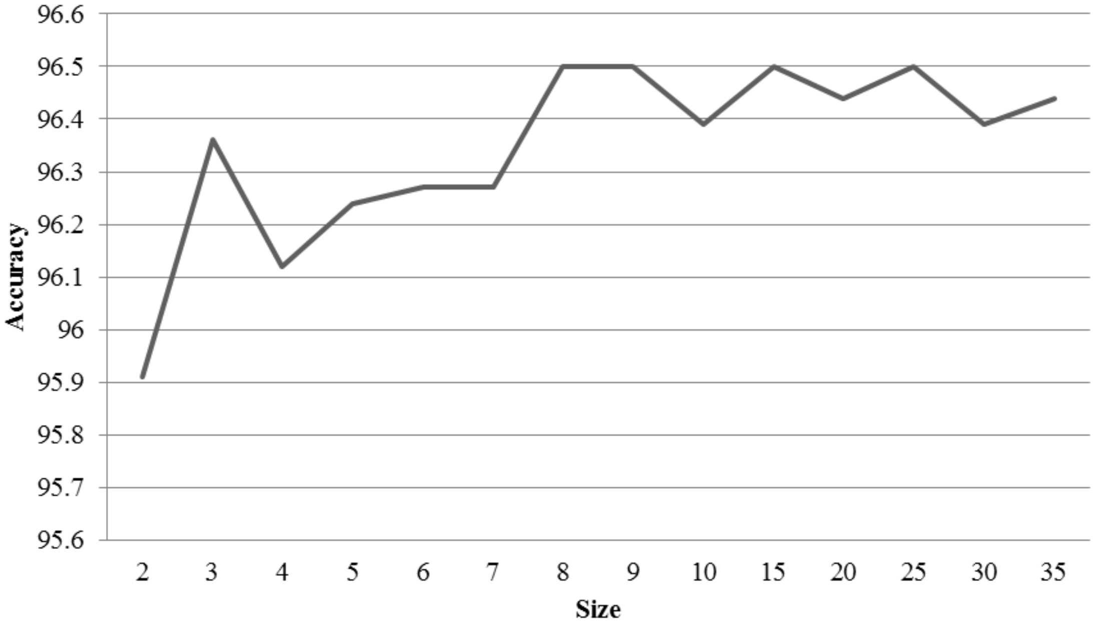


Figure 20: Experiments with Bagged set with different ensemble size

Experiments with Stacked-set:

For the Stacking meta-learner we ended up with different number of combinations for the 2 base, 3 base, 4 and 5 base learners. The stacked ensemble gave a number of heterogeneous models. A number of combinations were made with the five baseline learners giving us some interesting results. In the 2, 3, 4 and 5 base stack learners the models that outperformed on the training set can be seen in Table 8.

Experiments with k base learners:

The ensemble of two base learners was made using 2 different induction algorithms at level zero of the stacked model. The stacking experiments were started with k=2 base learners from the baseline set of learners. We discovered that DT when selected as the stacked-learner at level one gave the best results as compared to others. The highest accuracy with 2 base learners was near ***96.8***%. In the stacked learner, a new classifier was added in the process of ensemble generation, to get the model with the highest accuracy as discussed in heuristic method. Once the highest accuracy is achieved, the process of adding a classifier is stopped at a certain point. The process was stopped with k=5 base learners, as we had five baseline learners. Optimal accuracy was achieved with the size of three base learners of ***96.86***%. The increase in the base learners did not increase the accuracy as seen in Figure 21. Table 8 shows the results with different number of base learners in the stacked model.

Table 8: Stacked Model experiments with different base learners

Base Learner (Level 0)	**Level 1**	**Accuracy**
Naïve Bayesian and Decision Tree	Decision Tree	96.38
Decision Tree and Neural Network	Decision Tree	***96.8***
Logistic Regression and Decision Tree	Decision Tree	96.15
Neural Network, Decision Tree and Logistic Regression	Decision Tree	96.74
Naïve Bayesian, Neural Network and Decision Tree	Decision Tree	96.77
K Nearest Neighbor, Neural Network and Decision Tree	Decision Tree	***96.86***
Naïve Bayesian, K Nearest Neighbor, Decision Tree and Neural Network	Decision Tree	***96.86***
Neural Network, Naïve Bayesian, Decision Tree and Logistic Regression	Decision Tree	96.68
Neural Network, K Nearest Neighbor, Logistic Regression and Decision Tree	Decision Tree	***96.8***
Naïve Bayesian, Neural Network, K Nearest Neighbor, Logistic Regression and Decision Tree	Decision Tree	***96.86***

Researchers have experimented with combining these meta-classifiers over the years. Stacking and Bagging was combined with 2 base learning algorithms NB and C4.5 at level zero and Linear Regression at level 1[95]. However the study uses stable algorithms in the ensemble creation. Boosting and Stacking were used to make a multiple classifier where boosting was performed on base classifiers at level zero[96]. Thus different combinations can be made to form the perfect ensemble using the three techniques. Experiments were performed with this approach to get a bagged-stacked and a boosted-stacked set.

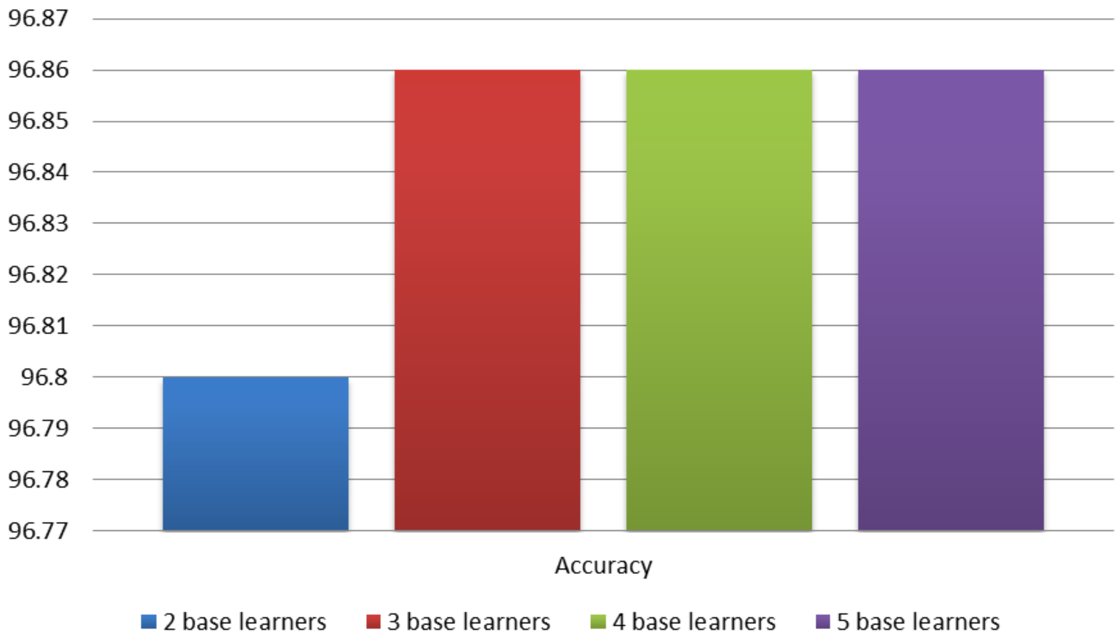


Figure 21: Results with k base learners

Experiments on Boosted-stacked set:

Another set of models is generated with the boosting algorithm, creating a mixture of heterogeneous and homogeneous ensembles. Among them the best performing model on validation set is reported in Table 9.

Table 9: Best Results with Boosting and Stacking Techniques

Meta Learner	**Base Learner (Level 0)**	**Level 1**	**Accuracy**
Boosting, Stacking	K Nearest Neighbor, Neural Network and Decision Tree	Decision Tree	97.18
Stacking	K Nearest Neighbor, Neural Network and Decision Tree	Boosted Decision Tree	97.19
Boosting, Stacking	K Nearest Neighbor, Neural Network and Decision Tree	Boosted Decision Tree	***97.27***

Experiments on Bagged-stacked set:

A set of experiments were performed to get a bagged-stacked set with the base classifiers previously used in the standalone experiments. By changing the classifiers or parameters, a number of ensembles were made. Among them the best models that had the highest accuracy are reported in Table 10.

Table 10: Best Results with Bagging and Stacking Techniques

Meta Learner	**Base Learner (Level 0)**	**Level 1**	**Accuracy**
Bagging, Stacking	K Nearest Neighbor, Neural Network and Decision Tree	Decision Tree	***97.57***
Stacking	K Nearest Neighbor, Neural Network and Decision Tree	Bagged Decision Tree	97.30

Table 11: Size and Accuracy with different ensembles

Meta- Learner	**Size**	**Accuracy**
Stacking	2	96.8
	2	96.15
	3	96.77
	3	***96.86***
	4	***96.86***
	4	96.8
	5	***96.86***
Boosting	4	96.15
	8	96.36
	10	***96.47***
	5	96.24
Bagging	9	96.21
	8	***96.5***
Boosting, Stacking	15	97.18
	9	97.19
	45	***97.27***
Bagging, Stacking	24	***97.57***
	11	97.30

Step 3:

At the end of Step 2, the best ensemble was obtained with Bagged and Stacked set with ***97.57%*** accuracy on the validation set. Now this model is tested on the Test data. Once the subset of models was trained, the best performance was obtained by the top two models (bagged-stacked subset and the boosted-stacked subset). These models were tested on the test set to get an accuracy of **98%** and **97.2**% respectively on the data set 1. On data set 2 we got ***90.33%*** accuracy with the Bag-Stack ensemble as seen in Table 12. The accuracy, precision and other evaluation metrics for the model can be seen in Table 13.

Table 12: Results with the Top two ensemble models

Data Set	**Proposed Ensemble**	**Accuracy (%)**
Public UCI data set (data set)	Bagged-Stacked Ensemble	***98***
	Boosted-Stacked Ensemble	***97.2***
Private Telecom data set (data set)	Bagged-Stacked Ensemble	90.33
	Boosted-Stacked Ensemble	86.33

Table 13: Results with Bag-Stack Ensemble Model

Data Set	**Accuracy**	**Precision**	**Recall**	**F-Measure**	**AUC**
Public UCI data set (data set 1)	98	98.98	87.39	92.82	0.95
Private Telecom data set (data set 2)	90.3	91.2	87.9	89.6	0.95

Comparison with Literature:

Many researchers have explored churn prediction by reporting results on different private data sets taken from different private telecom companies. However, the techniques used are mostly simple machine learning algorithms that gave just satisfactory results on the respective data sets. Table 14 shows the results obtained on the data set 2, by applying the techniques reported by different researchers. The same experimental settings were used for using their techniques. This proves that when it comes to private data sets, it cannot be made sure those same results will be achieved on other data sets. However, using ensemble techniques has not been explored in detail by researchers on these data sets. Our proposed technique was used to experiment on both public and real life churn data sets and proved to give far better results than reported.

Table 14: Comparison of literature reported results on real life churn data set (data set 2)

Paper	**Results on Data set 2**	**Reported Results**
M. C. Mozer et al (2000)[24]	65.58% churners	100% churners
S. Y. Hung et al. (2006)[10]	65% hit ratio	Hit ratio: 98%
J. Hadden (2006)[82]	69.59%	82%
J. Burez and D. Van den Poel (2009)[15]	0.808 AUC	0.6790 AUC
C. F. Tsai and M. Y. Chen (2010)[97]	66.8%	90.98%
G. Kraljević and S. Gotovac (2010)[83]	66.27%	91%
V. Umayaparvathi and K. Iyakutti(2012)[34]	74.4%	98.8%
C. Phua et al. (2012)[26]	74%	74.5%
E. Shaaban et al. (2012)[37]	72.25%	83.7%
G. Olle(2014)[98]	65.67%	76%

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1. Conclusion

The objective of this thesis was to investigate the performance of ensemble classification on telecom churn data sets. To this end, the performance of the proposed framework Bagged-Stacked Ensemble proved to be the best. The experiments were conducted on test sets of two data sets (public and private) and the results were evaluated using the evaluation metrics like accuracy, f-measure etc. The Bag-stack three base learner ensemble model runs on the test sets of both the public and private data set. The model was compared to individual classifiers and meta-classifiers. In the experiments, we observed that the best individual models that performed well were the unstable learning algorithms namely Decision Tree and the Neural Network, proving the theory that *unstable learning algorithms work best in ensemble classification*.

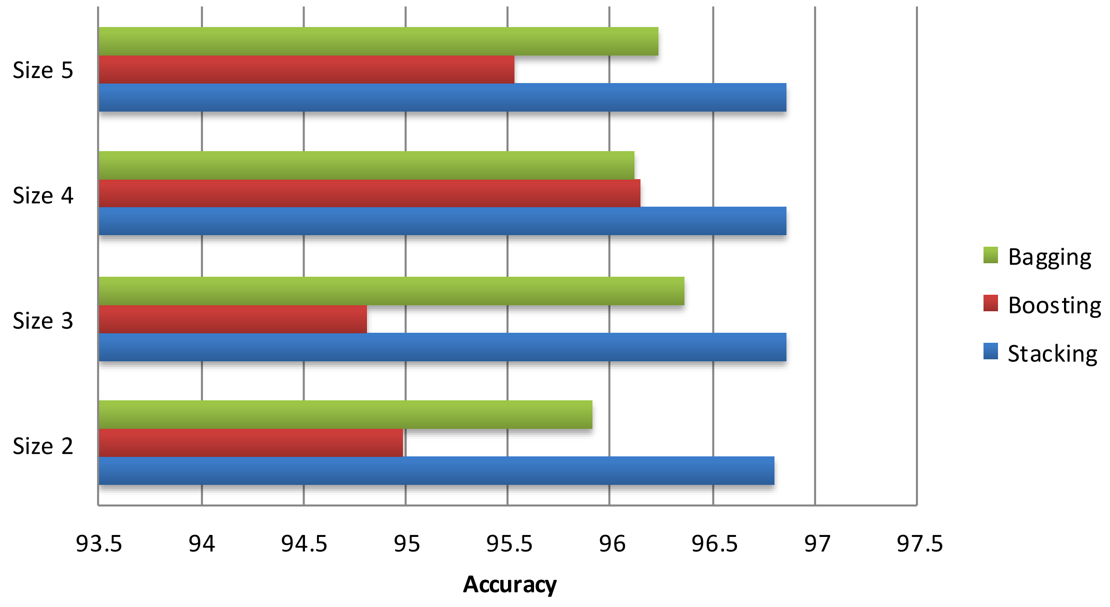


Figure 22: Comparison of Ensemble Techniques

It can be seen from Figure 22 that *Stacking can prove to be more reliable than bagging and boosting techniques* with a small size of base learners while boosting is the best among the three meta-classifiers. Among the sampling techniques, *Boosting is more accurate than Bagging*. Stacking technique gave the optimal accuracy with three base learners at level zero. The further increase in the base learners had no impact on the performance, thus proving that stacking *works best with three base learners* at level zero. Ensemble techniques overall give far better results than the individual classifiers.

Combination of the Bagging and Stacking meta-classifiers proved that heterogeneous ensembles are more diverse and accurate. The remarkable result achieved with this amalgamation proves the power of ensemble techniques. All necessary steps were taken to avoid the over-fitting problems that might occur with using the sampling techniques. The Bagging and Stacking techniques had a stronger affect as compared to the combination of Boosting and Stacking. However both outperformed the individual and meta-classifiers. Figure 23 shows that Bagged-Stacked ensemble gave better accuracy as compared to the Boosted-stacked ensemble.

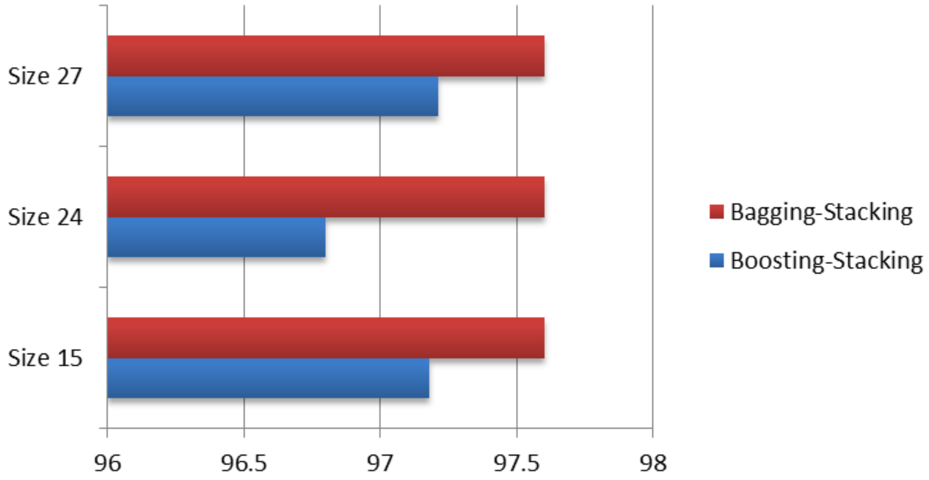


Figure 23: Comparison between Bagged-stacked and Boosted-stacked ensemble

The Meta Decision Tree obtained with the Bag-Stack ensemble model shows that the top five attributes in the real life data set (data set 2) include the ***Package, network age, SMS, total, off-network and on-network revenue***. While *Favorite other network* falls in the top ten attributes to predict the churners. With this model, ***91%*** of the potential churners were predicted. As compared to the other prediction models in literature, usually the overall performance of the active and churned customers is counted in an imbalanced data set with a large amount of active subscribers and a hand full of churners. By using a balanced set, with all real life instances we can guarantee that the model predicted an accurate amount of churners with no bias. The framework proved to be suitable for both balanced (data set 2) and imbalanced (data set 1) data sets. The best accuracy of ***98%*** was achieved for the UCI churn data set which is the highest reported to date. The top five attributes for this set include the ***total night minutes, day calls, international minutes, number of service calls and account length***.

In conclusion, the effect of ensemble classification with the combination of heterogeneous meta-learning techniques is notable. Future work will investigate if this ensemble approach can be applied to various industries in addition to the telecommunication sector, proving if this hybrid ensemble is applicable to all data sets (balanced or imbalanced). And what other combinations of meta-learning techniques to use with the Stacking, Bagging and Boosting techniques.

BIBLIOGRAPHY

[1] S. V. Nath, and R. S. Behara, "Customer churn analysis in the wireless industry: A data mining approach." pp. 505-510.

[2] G.-e. Xia, and W.-d. Jin, “Model of customer churn prediction on support vector machine,” *Systems Engineering-Theory & Practice,* vol. 28, no. 1, pp. 71-77, 2008.

[3] M. R. Ismail, M. K. Awang, M. N. A. Rahman, and M. Makhtar, “A Multi-Layer Perceptron Approach for Customer Churn Prediction,” *International Journal of Multimedia and Ubiquitous Engineering,* vol. 10, no. 7, pp. 213-222, 2015.

[4] K. Oseman, S. M. Shukor, N. A. Haris, and F. A. Bakar, “Data mining in churn analysis model for telecommunication industry,” *Journal of Statistical Modeling and Analytics,* vol. 1, no. 19-27, 2010.

[5] P. Kisioglu, and Y. I. Topcu, “Applying Bayesian Belief Network approach to customer churn analysis: A case study on the telecom industry of Turkey,” *Expert Systems with Applications,* vol. 38, no. 6, pp. 7151-7157, 2011/06, 2011.

[6] I. Brandusoiu, and G. Toderean, “Churn prediction in the telecommunications sector using support vector machines,” *Annals of The Oradea University. Fascicle of Management and Technological Engineering* 2013.

[7] Y. Zhao, B. Li, X. Li, W. Liu, and S. Ren, "Customer Churn Prediction Using Improved One-Class Support Vector Machine," *Advanced Data Mining and Applications,* Springer Science + Business Media, 2005, pp. 300-306.

[8] A. Sharma, and D. P. K. Panigrahi, “A Neural Network based Approach for Predicting Customer Churn in Cellular Network Services,” *International Journal of Computer Applications (0975 – 8887),* vol. 27, no. No.11, 2011.

[9] Ionut B. Brandusoiu, and G. Toderean, “A Neural Networks Approach for Churn Prediction Modeling in Mobile Telecommunications Industry,” *Annals of The University of Craiova,* vol. 11, no. 1, pp. 9-16, 2014.

[10] S. Y. Hung, D. C. Yen, and H. Y. Wang, "Applying data mining to telecom churn management," *Expert Systems with Applications,* vol. 31, no. 3, pp. 515-524, 2006/10, 2006.

[11] K. b. Oseman, N. A. Haris, and F. b. A. Bakar, "Data Mining in Churn Analysis Model for Telecommunication Industry," *Journal of Statistical Modeling and Analytics,* vol. 1, pp. 19-27 2010.

[12] C. F. Tsai, and M. Y. Chen, "Variable selection by association rules for customer churn prediction of multimedia on demand," *Expert Systems with Applications 37,* 2010.

[13] M. Owczarczuk, "Churn models for prepaid customers in the cellular telecommunication industry using large data marts," *Expert Systems with Applications 37,* pp. 4710–4712, 2010.

[14] W. Verbeke, D. Martens, C. Mues, and B. Baesens, "Building comprehensible customer churn prediction models with advanced rule induction techniques," *Expert Systems with Applications,* vol. 38, no. 3, pp. 2354-2364, 2011/03, 2011.

[15] J. Burez, and D. Van den Poel, "Handling class imbalance in customer churn prediction," *Expert Systems with Applications,* vol. 36, no. 3, pp. 4626-4636, 2009/04, 2009.

[16] C.-F. Tsai, and Y.-H. Lu, "Customer churn prediction by hybrid neural networks," *Expert Systems with Applications 36,* pp. 12547–12553, 2009.

[17] A. Lemmens, and C. Croux, "Bagging and boosting classification trees to predict churn," *Journal of Marketing Research,* vol. 43, no. 2, pp. 276-286, 2006.

[18] T. Vafeiadis, K. I. Diamantaras, G. Sarigiannidis, and K. C. Chatzisavvas, "A comparison of machine learning techniques for customer churn prediction," *Simulation Modelling Practice and Theory,* vol. 55, pp. 1-9, 2015.

[19] W. Verbeke, K. Dejaeger, D. Martens, J. Hur, and B. Baesens, "New insights into churn prediction in the telecommunication sector: A profit driven data mining approach," *European Journal of Operational Research,* vol. 218, no. 1, pp. 211-229, 2012/04, 2012.

[20] S. Maldonado, Á. Flores, T. Verbraken, B. Baesens, and R. Weber, "Profit-based feature selection using support vector machines – General framework and

an application for customer retention," *Applied Soft Computing,* vol. 35, pp. 740-748, 2015/10, 2015.

[21] B. Huang, B. Buckley, and T.-M. Kechadi, "Multi-objective feature selection by using NSGA-II for customer churn prediction in telecommunications," *Expert Systems with Applications 37* pp. 3638–3646, 2010.

[22] A. Amin, S. Shehzad, C. Khan, I. Ali, and S. Anwar, "Churn Prediction in Telecommunication Industry Using Rough Set Approach," *New Trends in Computational Collective Intelligence*, Springer Science + Business Media, 2015, pp. 83-95.

[23] L. Peng, Y. Xiaoyang, S. Boyu, and H. Jiuling, "Telecom Customer Chum Prediction Based on Imbalanced Data Re-sampling Method," in 2013 International Conference on Measurement, Information and Control (ICMIC), 2013, pp. 229-233.

[24] M. C. Mozer, R. Wolniewicz, D. B. Grimes, E. Johnson, and H. Kaushansky, "Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry," *IEEE Trans. Neural Netw.,* vol. 11, no. 3, pp. 690-696, 2000/05, 2000.

[25] J. Basiri, F. Taghiyareh, and B. Moshiri, "A Hybrid Approach to Predict Churn," in 2010 IEEE Asia-Pacific Services Computing Conference, 2010, pp. 485-491.

[26] C. Phua, H. Cao, J. B. Gomes, and M. N. Nguyen, "Predicting Near-Future Churners and Win-Backs in the Telecommunications Industry," 2012.

[27] Y. Richter, E. Yom-Tov, and N. Slonim, "Predicting customer churn in mobile networks through analysis of social groups," *Proceedings of the 2010 SIAM International Conference on Data Mining*, Society for Industrial & Applied Mathematics (SIAM), 2010, pp. 732-741.

[28] J. Xiao, Y. Xiao, A. Huang, D. Liu, and S. Wang, "Feature-selection-based dynamic transfer ensemble model for customer churn prediction," *Knowledge and Information Systems*, vol. 43, no. 1, pp. 29-51, 2014/01/16, 2014.

[29] A. Baumann, S. Lessmann, K. Coussement, and K. W. D. Bock, "Maximize What Matters: Predicting Customer Churn With Decision-Centric Ensemble Selection," in ECIS 2015 2015.

[30] Y. Huang, B. Q. Huang, and M. T. Kechadi, "A New Filter Feature Selection Approach for Customer Churn Prediction in Telecommunications," in 2010 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) 2010, pp. 338-342.

[31] J. Hadden, A. Tiwari, R. Roy, and D. Ruta, "Churn prediction: Does technology matter," *International Journal of Intelligent Technology 1*, no. no. 2, pp. 104-110, 2006.

[32] G. Kraljević, and S. Gotovac, "Modeling data mining applications for prediction of prepaid churn in telecommunication services," *AUTOMATIKA: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije 51*, no. no. 3, pp. 275-283, 2010.

[33] U. Droftina, A. Košir, and M. Štular, "Subscriber Churn Analysis with Selected Mobile Provider," in 20th International Electrotechnical and Computer Science Conference ERK, 2011.

[34] V. Umayaparvathi, and K. Iyakutti, "Applications of Data Mining Techniques in Telecom Churn Prediction," *International Journal of Computer Applications,* vol. 42, no. 20, pp. 5-9, 2012/03/31, 2012.

[35] C. Kirui, L. Hong, W. Cheruiyot, and H. Kirui, "Predicting customer churn in mobile telephony industry using probabilistic classifiers in data mining," in IJCS 10, 2013.

[36] S. A. Qureshi, A. S. Rehman, A. M. Qamar, A. Kamal, and A. Rehman, "Telecommunication subscribers' churn prediction model using machine learning," in Eighth International Conference on Digital Information Management (ICDIM 2013), 2013.

[37] E. Shaaban, Y. Helmy, A. Khedr, and M. Nasr, "A proposed churn prediction model," in IJERA 2, 2012, pp. 693-697.

[38] Y. Liu, and Y. Zhuang, "Research Model of Churn Prediction Based on Customer Segmentation and Misclassification Cost in the Context of Big Data," *JCC,* vol. 03, no. 06, pp. 87-93, 2015.

[39] L. Peng, B. Tingting, L. Yang, and L. Siben, "Telecom Customer Churn Prediction Method Based on Cluster Stratified Sampling Logistic Regression," in International Conference on Software Intelligence Technologies and

Applications & International Conference on Frontiers of Internet of Things 2014, 2014.

[40] L. Xu, A. Krzyżak, and C. Y. Suen, “Methods of combining multiple classifiers and their applications to handwriting recognition,” *Systems, Man and Cybernetics, IEEE Transactions on,* vol. 22, no. 3, pp. 418-435, 1992.

[41] J. Kittler, and F. Roli, “Multiple classifier systems,” *Lecture notes in computer science*, 2002.

[42] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection." pp. 1137-1145.

[43] D. Opitz, and R. Maclin, “Popular ensemble methods: An empirical study,” *Journal of Artificial Intelligence Research*, pp. 169-198, 1999.

[44] T. G. Dietterich, “Machine-learning research,” *AI magazine,* vol. 18, no. 4, pp. 97, 1997.

[45] R. E. Schapire, “The strength of weak learnability,” *Machine learning,* vol. 5, no. 2, pp. 197-227, 1990.

[46] L. Breiman, “Bagging predictors,” *Machine learning,* vol. 24, no. 2, pp. 123-140, 1996.

[47] J. Wichard, C. Merkwirth, and M. Ogorzalek, “Building ensembles with heterogeneous models,” 2003.

[48] N. Rooney, D. Patterson, S. Anand, and A. Tsymbal, "Dynamic integration of regression models," *Multiple Classifier Systems*, pp. 164-173: Springer, 2004.

[49] C. J. Merz, "Dynamical selection of learning algorithms," *Learning from Data,* pp. 281-290: Springer, 1996.

[50] L. Breiman, “Heuristics of instability and stabilization in model selection,” *The annals of statistics,* vol. 24, no. 6, pp. 2350-2383, 1996.

[51] L. Breiman, “Randomizing outputs to increase prediction accuracy,” *Machine Learning,* vol. 40, no. 3, pp. 229-242, 2000.

[52] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes, "Ensemble selection from libraries of models." p. 18.

[53] A. J. Sharkey, N. E. Sharkey, U. Gerecke, and G. O. Chandroth, "The “test and select” approach to ensemble combination," *Multiple Classifier Systems,* pp. 30-44: Springer, 2000.

[54] A. Lazarevic, and Z. Obradovic, "Effective pruning of neural network classifier ensembles." pp. 796-801.

[55] L. C. Molina, L. Belanche, and À. Nebot, "Feature selection algorithms: a survey and experimental evaluation." pp. 306-313.

[56] G. Giacinto, and F. Roli, “Design of effective neural network ensembles for image classification purposes,” *Image and Vision Computing,* vol. 19, no. 9, pp. 699-707, 2001.

[57] F. Roli, G. Giacinto, and G. Vernazza, "Methods for designing multiple classifier systems," *Multiple Classifier Systems,* pp. 78-87: Springer, 2001.

[58] D. Partridge, and W. B. Yates, “Engineering multiversion neural-net systems,” *Neural Computation,* vol. 8, no. 4, pp. 869-893, 1996.

[59] G. Giacinto, and F. Roli, “An approach to the automatic design of multiple classifier systems,” *Pattern recognition letters,* vol. 22, no. 1, pp. 25-33, 2001.

[60] C. J. Merz, “Classification and regression by combining models,” UNIVERSITY OF CALIFORNIA IRVINE, 1998.

[61] E. Fix, and J. L. Hodges Jr, *Discriminatory analysis-nonparametric discrimination: Small sample performance,* DTIC Document, 1952.

[62] B. V. Dasarathy, “Nearest neighbor ({NN}) norms:{NN} pattern classification techniques,” 1991.

[63] D. W. Aha, D. Kibler, and M. K. Albert, “Instance-based learning algorithms,” *Machine learning,* vol. 6, no. 1, pp. 37-66, 1991.

[64] D. R. Cox, and E. J. Snell, *Analysis of binary data*: CRC Press, 1989.

[65] G. King, and L. Zeng, “Explaining rare events in international relations,” *International Organization,* vol. 55, no. 03, pp. 693-715, 2001.

[66] N. Friedman, D. Geiger, and M. Goldszmidt, “Bayesian network classifiers,” *Machine learning,* vol. 29, no. 2-3, pp. 131-163, 1997.

[67] P. Domingos, and M. Pazzani, “On the optimality of the simple Bayesian classifier under zero-one loss,” *Machine learning,* vol. 29, no. 2-3, pp. 103-130, 1997.

[68] L. H. Tsoukalas, and R. E. Uhrig, *Fuzzy and neural approaches in engineering*: John Wiley & Sons, Inc., 1996.

[69] D. E. Rumelhart, J. L. McClelland, and P. R. Group, *Parallel distributed processing*: IEEE, 1988.

[70] L. Fausett, “Fundamentals of neural networks: architectures, algorithms, and applications,” 1994.

[71] L. Rokach, and O. Maimon, *Data mining with decision trees: theory and applications*: World scientific, 2014.

[72] D. J. Hand, H. Mannila, and P. Smyth, *Principles of data mining*: MIT press, 2001.

[73] Y. Freund, and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting." pp. 23-37.

[74] Y. Freund, R. Schapire, and N. Abe, “A short introduction to boosting,” *Journal-Japanese Society For Artificial Intelligence,* vol. 14, no. 771-780, pp. 1612, 1999.

[75] D. H. Wolpert, “Stacked generalization,” *Neural networks,* vol. 5, no. 2, pp. 241-259, 1992.

[76] S. Džeroski, and B. Ženko, *Stacking with multi-response model trees*: Springer, 2002.

[77] K. Ming, and W. Ian, *Stacked Generalization: when does it work?,* Working Paper 97/3, Department of Computer Science, University of Waikato, 1997.

[78] S. Džeroski, and B. Ženko, “Is combining classifiers with stacking better than selecting the best one?,” *Machine learning,* vol. 54, no. 3, pp. 255-273, 2004.

[79] B. Efron, and R. J. Tibshirani, *An introduction to the bootstrap*: CRC press, 1994.

[80] P. Liatsis, *Recent Trends in Multimedia Information Processing*: World Scientific, 2002.

[81] C. Blake, and C. J. Merz, “{UCI} Repository of machine learning databases,” 1998.

[82] J. Hadden, A. Tiwari, R. Roy, and D. Ruta, “Churn prediction: Does technology matter,” *International Journal of Intelligent Technology,* vol. 1, no. 2, pp. 104-110, 2006.

[83] G. Kraljević, and S. Gotovac, “Modeling data mining applications for prediction of prepaid churn in telecommunication services,” *AUTOMATIKA: časopis za*

automatiku, mjerenje, elektroniku, računarstvo i komunikacije, vol. 51, no. 3, pp. 275-283, 2010.

[84] G. Martınez-Munoz, and A. Suárez, "Aggregation ordering in bagging." pp. 258-263.

[85] R. E. Banfield, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer, "Ensemble diversity measures and their application to thinning," *Information Fusion,* vol. 6, no. 1, pp. 49-62, 2005.

[86] M. Kubat, R. Holte, and S. Matwin, "Learning when negative examples abound," *Machine Learning: ECML-97,* pp. 146-153: Springer, 1997.

[87] M. Kubat, and S. Matwin, "Addressing the curse of imbalanced data sets: One sided sampling."

[88] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: special issue on learning from imbalanced data sets," *ACM Sigkdd Explorations Newsletter,* vol. 6, no. 1, pp. 1-6, 2004.

[89] E. Alpaydin, *Introduction to Machine Learning*: MIT Press, 2004.

[90] D. Zhang, and J. J. P. Tsai, *Machine Learning Applications in Software Engineering*: World Scientific, 2005.

[91] *Recent Trends in Multimedia Information Processing*: World Scientific, 2002.

[92] D. M. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," 2011.

[93] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern recognition,* vol. 30, no. 7, pp. 1145-1159, 1997.

[94] T. Fawcett, "ROC graphs: Notes and practical considerations for researchers," *Machine learning,* vol. 31, no. 1, pp. 1-38, 2004.

[95] K. M. Ting, and I. H. Witten, "Stacking bagged and dagged models." pp. 367-375.

[96] N. Hatami, and R. Ebrahimpour, "Combining multiple classifiers: diversify with boosting and combining by stacking," *International Journal of Computer Science and Network Security,* vol. 7, no. 1, pp. 127-131, 2007.

[97] C.-F. Tsai, and M.-Y. Chen, “Variable selection by association rules for customer churn prediction of multimedia on demand,” *Expert Systems with Applications*, vol. 37, no. 3, pp. 2006-2015, 2010.

[98] G. Olle, “A Hybrid Churn Prediction Model in Mobile Telecommunication Industry,” *IJEEEE,* 2014.