Predicting Customer Retention using XGBoost and Balancing Methods

Atallah M. AL-Shatnwai[1]
Department of Information Systems
Al al-Bayt University
Mafraq, Jordan

Mohammad Faris[2]
Altibbi
Amman, Jordan

***Abstract*—Customer retention is considered as one of the important concerns for many companies and financial institutions like banks, telecommunication service providers, investment services, insurance and retail sectors. Recent marketing indicators and metrics show that attracting and gaining new customers or subscribers is much more expensive and difficult than retaining existing ones. Therefore, losing a customer or a subscriber will negatively impact the growth and the profitability if the company. In this work, we propose a customer retention model based on one of the most powerful machine learning classifiers which is XGBoost. The latter classifier is experimented when combined wit different oversampling methods to improve its performance in the used imbalanced dataset. The experimental results show very promising results compared to other well-known classifiers.**

***Keywords*—Customer retention; churn prediction; oversampling; XGBoost**

I. Introduction

The sector of telecommunications has grown into a principal industry in the developed countries [1]. Technical progress, in addition to the growing number of operators, increased the degree of competition in this sector. The firms are, thus, working hard so as to survive in this highly-competitive market and are using varying strategies for this purpose. In this respect, three major strategies have been suggested for generation of further revenues [2], [1]: (i) gaining new customers, (ii) increasing the sales to the present customers, and (3) increasing the period of retention of the customers. A comparison of effectiveness of these strategies based on the Return on Investment (RoI) achieved from each revealed that the last strategy is the most rewarding strategy [1]. This suggests that retaining a current customer costs the firm much less money than gaining a new one. Moreover, retention is much easier than increasing the sales to the current customers [1]. For successful application of the third strategy, the firms need to reduce the potential for churn of the customers.

Customer churn can be defined as a customer who terminates their relationship with the service provider and switches to another competitor in the market [3]. In the dynamic market, there are different types of factors that influence the decision of customer when he/she decides to churn. These factors include: Technological factors where a customer will be motivated to switch to a competitive company which offers more technological advanced products. Economic factors, for example, cheaper price or product offered by a competitor. Quality, as poor quality of customer service will push him to leave his company like the Weak coverage. Or even the difficulties in the usage of a product and delays in delivering a service or product [4].

Nowadays, telecom service providers pay a great attention for customer retention than customer acquisition because the cost of getting a new customer is more than the cost of keeping an existing one. Consequently, well-known telecom companies consider that their current customer database is the most beneficial asset [3].

Customer churn prediction is a concept of predicting and identifying those customers who are about to churn by leaving the telecom company to another competitors [5], [3]. Churn prediction models assistance in effective customer churn management. Predicting churn assists the preparation of targeted preservation strategies to limit the losses and to enhance marketing decisions build customer loyalty and increasing profitability [6]. For example, specific incentives and offers can be given to the most risky customer segments. Marketing department, can plan awarding customers discounts, other promotions/events, other products of other sister companies wherever applicable [7].

Churn management is defines as a concept that investigates operator's process which save profitable customers [7]. Whenever a given company attempts to determine customers whom want to churn before they do that is considered as proactive churn management approach. After that company gives special offers (promotions) for such customers to prevent their churning. Those considered offer programs gain very important advantages of getting lower cost. On the other hand, if churn predictions of such approaches are inaccurate, they will be considered as wasteful since companies will waste money for customers whom will not churn. Consequently, to get a major success of customer incentive programs, there is a crucial need to have an accurate customer-churn prediction model [7], [8].

In the last two decades, several machine learning algorithms were proposed in literature to tackle the churning prediction problem. The first type of these algorithms is the basic machine learning algorithms and the most popular ones. Such algorithms include Artificial Neural Networks, Decision Trees learning, and Logistic Regression, Support Vector Machines (SVM), Naïve Bayes and many others [9], [3], [5].

Another type of machine learning models for churn prediction is the ensemble algorithms which are based on the concept of ensemble learning. Ensemble learning is a way of developing various weak classifiers from which a new classifier is derived which performs better than any of the weak classifiers [3]. These weak classifiers may differ in the algorithm used, hyperparameters, the training samples or the included features. Examples of Ensembles of churn prediction include Random forest, RotBoost, Rotation Forest [10], [5], [11].

One of the major problems with churn prediction is the imbalanced distribution of the classes as the number of non-churning customers are much more than the churning ones [12], [1], [3]. This makes it challenging for the machine learning classifiers to discover the churning customers. Different approaches were proposed for handling this issue for churn prediction such as over-sampling and under-sampling [3], [11], [13].

In this work, we propose and experiment the application of one the most powerful and effective classification algorithms in the last few years which is Extreme Gradient Boosting (XGBoost) for churn prediction in telecommunication business. The performance of XGBoost is experimented after combining it with different popular oversamling methods to improve its performance when applied for imbalanced dataset. This application is introduced by a technical framework and that explains the application in details. The performance is measured using different evaluation metrics and compared with common and well-known classifiers.

This paper is structured as follows: In Section II we review and discus the previous imbalanced distribution data studies, whereas in Section III we review the main methods that will be used in the proposed framework. In Section IV we describe the framework of XGBoost combined with oversampling methods. In Section V, the dataset used in this work is described. The evaluation measures are listed in Section VI. The experiments and results are described and discussed in details in Section VII. Finally the conclusion of this work is given in Section VIII.

II. Related Works

Many researchers have looked at the imbalanced data problem where the numbers of the classes of the churned customers are lower than those of the classes of active customers, which is quite a serious issue in the churn prediction [3].

Idris et al [11] suggested an approach that is based on genetic programming using AdaBoost so as to model churn problem in the telecommunications sector. These researchers employed a Particle Swarm Optimization-based under-sampling in order to tackle the imbalance in the telecom data. This method provides an unbiased distribution of the training set to a prediction system that is depending on GP-AdaBoost. Performance of this proposed approach was evaluated on two standard datasets, one for Orange Telecom and one for cell2cell. The generated churn prediction accuracies were 0.86 AUC for the Orange Telecom data and 0.91 AUC for the cell2cell data.

Faris [3] presented hybrid model that is based on the over-sampling method, which integrates Particle Swarm Optimization (PSO) with a Random Weight Network for solving the problem of churn in the telecommunications data. The researchers applied the ADASYN over-sampling method in order to enhance learning from the imbalanced churn data. The results showed that use of the ADASYN method could significantly enhance the rate of coverage of the churn customers. Furthermore, this hybrid model is characterized by high interpretability since the allotted feature weights provide indicator of importance of their respective features in the process of classification.

Hanif and Azhar [14] employed the data-balancing methods of Synthetic Minority Over-sampling (SMOTE), random under-sampling, and random over-sampling in their efforts to tackle the problem of class imbalance. They found that random over-sampling generated the best data-balancing results. Hence, these researchers concluded that the extracted features that are most important are actually the features related to the customer's call. Amin et al. [13] held comparison in levels of performance of six over-sampling methods, namely, the SMOT, Mega-Trend Diffusion Function (MTDF), coupled top-N reverse k-nearest neighbor, adaptive synthetic sampling, immune centroid over-sampling, and majority weighted minority over-sampling methods. Their empirical findings indicated that the net predictive performance of the MTDF and the rules generation based on genetic algorithms methods were better than the levels of performance of the other assessed over-sampling approaches and the rule-generation algorithms.

Faris [15] applied the Neighborhood Cleaning Rules (NCL) under-sampling method for balancing churn data. The NCL method takes quality of the removed data into consideration by conducting data cleaning rather than data reduction. After application of the NCL method, a modfied version of the PSO, commonly known as the Constricted PSO, is trained so as to develop the churn prediction model. Tests showed that the NCL could signficantly improve the rate of coverage of the churn classes.

Idris et al [16] suggested intelligent churn prediction system for the telecommunication data by using efficient feature extraction method and an ensemble method. They employed under-sampling in order to handle the data imbalance problem and found that the minimal redundancy and maximal relevance (mRMR) technique could return the most explanatory features when compared with the Fisher's ratio and the F-score. Moreover, the RotBoost method combined with the mRMR features provided appreciably high prediction performance when applied on the standard telecommunication datasets.

Idris et al. [10] employed PSO-based under-sampling for the purpose of churn prediction. The function of the PSO is to search for the examples that are most informative of the majority class, order them, and integrate them with the minority class so as to maximize the classification accuracy. These researchers chose maximizing AUC as the fitness function in conjunction with the Random Forest (RF) and the k-NN classifiers. The evaluation results uncovered that the PSO-based technique enhanced performance of the k-NN and the RF classifiers.

Qureshi et al. [17] presented discussion of use of under-sampling and over-sampling approaches for solving the class imbalance problem to identify the customers who are close to churn on the basis of historical data. Burez and Van den Poel [8] researched into the data imbalance problem in the models for churn prediction and compared the levels of performance between Advanced Under-Sampling, Random Sampling, Weighted Random Forests, and the Gradient Boosting Model. They employed the AUC and Lift as the model performance metrics. The evaluation outcomes revealed that the under-sampling technique outpaced the other examined methods.

In order to tackle the churning prediction problem for the telecommunication companies, an approach based on the XGBoost classifier with over-sampling methods is proposed. Four common and renowned over-sampling methods are used and a comparison is held between them in terms of their abilities to handle the data imbalance problem, which are random over-sampling, SMOTE, ADASYN, and Borderline SMOTE.

III. Methods

This section describes the oversampling methods and the XGBoost which are applied to build the retention prediction model in this work.

A. Oversampling Methods

In this section the used oversampling methods for churn prediction will be described.

1) Random Oversampler: One possible method to tackle the oversampling problem is to create new samples in the classes which are under-represented. The most easy and basic approach is to generate new rare samples by randomly sampling with replacement the current available rare samples which is simply copying some of them.

2) SMOTE: [18] is concerned with creating new synthetic data points that are very similar to the real data points. Given a dataset D with N number of data instances. Where a^M is the number of instances of the major class M and b^m is the number of instances of the minor class m. Mainly, SMOTE focuses on increasing the ratio of the minor class by synthesizing new data points. It starts by selecting a minority-labeled data point i with a number of nearest neighbors k. In which, the selected neighbors are from the minor class. Depending on a predefined sampling rate, a random number of the points of the selected neighbors are chosen. Hence, those randomly-selected neighbor points create a line segment that links them with the data point i. From each line segment, a random point is selected to be the new synthetic data point. This process is repeated for all minority-labeled data points.

3) ADASYN: In 2008, Another over-sampling strategy was designed, which is ADASYN [19]. ADASYN was suggested to alleviate the bias during the learning process and to perform adaptive learning by adaptively setting the decision area of the minority points that are difficult to learn. In ADASYN, the ratio of the minority instances to the majority instances d is calculated to find the appropriate number of synthetic instances G for the minority class (Eq. 1).

$$G = (b^m - a^M) \times \beta, \qquad \beta \in [0, 1] \tag{1}$$

For each minority-labeled point, a k number of nearest neighbors is determined. The ratio of the majority class of the neighbor set is computed by $r_i = No.majority\ points/k$. Whereas, the normalization of r_i represents the density distribution ($r_i^{'}$). The adoption of a density distribution makes ADASYN different than the previous algorithms to adaptively learn the data points of the minority class. The density distribution is used to find the number of synthetic points for each minority data point, as in Eq. 2.

$$g_i = r_i^{'} \times G \tag{2}$$

For each minority data point x_i, one random point of the minority-labeled points of the neighbor set x_j is selected to generate the synthetic points g_i. The new data points are created as in Eq. 3. In which, $diff_j$ is the difference between x_i and x_j, and λ is a random number.

$$x_{new} = x_i + diff_j \times \lambda, \qquad \lambda \in [0, 1] \tag{3}$$

The density distribution of ADASYN defines non-uniform weights of the minority points, which lead to effectively decide the number of synthetic points to be generated for each minority-labeled point.

4) Borderline SMOTE: In 2005, Borderline-SMOTE was proposed [20], which is an extension of SMOTE with more powerful performance ability. Essentially, Borderline-SMOTE performs two stages; classifies the neighbors into three types of regions to identify the borderline instances, then synthesizes the new points. In the first version of Borderline-SMOTE (Borderline-SMOTE1), the new data is solely created from the determined borderline instances.

In Borderline-SMOTE1, the nearest neighbors of a minority data point are selected regardless of the type of the class (major or minor). Thus, the rate of the points of the major class of the selected neighbors decides whether the corresponding data point belongs to the noise, danger, or safe regions. If the neighbors of the minority point are all form the majority class, then it is classified as a noise point. If the neighbors contain majority points of ratio greater than $b^m/2$, then the respective data point is classified as a danger point. While having the number of majority-labeled points with a ratio of less than $b^m/2$, results in classifying the respective point as a safe point. As a result, all minority points classified as danger points called borderline instances. In order to create new points, for each point from the danger region p_i, a k number of neighbors is selected from the same minority class. Hence, a randomly s points are chosen from the neighbors. The difference is calculated between each point from s and the respective danger point, where the difference denoted by $(diff_j)$. Then the new synthetic data point is generated based on Eq. 4, given that r_j is a random number $\in [1, s]$.

$$p\_new = p_i + r_j \times diff_j \tag{4}$$

In version 2 of the Borderline-SMOTE, the neighbors of the points at the danger region are considered from the two classes; the minority and the majority.

B. XGBoost

Extreme Gradient Boosting (XGBoost) can be defined as an improved version of the Gradient boosting algorithm, and this algorithm considers one of the machine learning techniques\tools that applied for classification and regression problems. The idea behind its concept is to boost the weak learner to become stronger using the decision trees mechanism. This improved version utilized a more regularized model in order to reduce and control the overfitting of the model to improve its performance. Basically, the XGBoost adopted the three main techniques of the gradient boosting, which are Regularized, Gradient and Stochastic boosting to enhance and tune the model. Moreover, it has the ability to decrease the time consumption alongside using the optimal resources of memory, parallel execution and handling the missing values while generating the tree construction [21], [22].

The XGBoost as the tree algorithms implementation considers the features in the dataset as a conditional node, where it divides into various branches and splits until the leaf node that represents the selected detection of the problem. In addition, the XGBoost depends on its hyperparameter in order to perform well considering its number and characteristics.

IV. XGBoost with Oversampling

In this section, the framework for customer churn prediction is described, see Fig. 1. The main two components of this framework are the classification algorithm and the oversampling method. For classification, the powerful XGBoost classifier is used. While for oversampling, we try four different oversampling methods all of them are variations of the SMOTE algorithm which are the basic version of SMOTE, ADASYN, SMOTE borderline, and the simplest method which is random oversampling. The proposed framework have the following steps:

- First, the dataset will be split into two parts. The first part is used for parameter tuning, while the second part will be used for training and testing the developed models. A stratified sampling is used because the dataset is imblalanced and it important to have the same ration of class labels in both samples of parts.
- In the second step, the parameters of XGBoost are tuned using the GridSearch algorithm implemented in Python. This is very important step because XGBoost is vert sensitive to the initial values of its many parameters. This step will assure to maximize the performance of the classifier in the rest of the experiments.
- The second part of the dataset is used to trian and test the algorithm using the 10-folds cross validation technique. Using this way, 9 folds are used to train the model and one fold is used for testing the model. This process is repeated 10 times. Then the average of the results are calculated.
- The next step is the oversampling method. In this step four very popular oversampling methods are used: Random oversampler, SMOTE, ADASYN and Borderline SMOTE. It is very important to note that these oversampling methods are applied only in the training folds. All the oversampling methods are applied at different oevrsampling ratios to study the effect of this ratio on the classification results of the classifier.
- After the oversampling step, the XGBoost is trained using the oversampled data and tested on the testing data which is not oversampled.
- After applying the cross validation process, the performance of the XGBoost is evaluated using the common classification metrics which are : the accuracy rate, precision, recall, and F1-measure.

V. Description of Churn Dataset

The dataset used in this work which is used to build the retention prediction models consists of information of 5000 subscribers and includes independent 20 variables which are shown in Table I. Note that three features were removed from the dataset because they do not provide any information which are: state, area code, and phone number. The dependent variable in the dataset is whether the customer left the company or not, which is coded as 1 for “yes” and 0 for “no.”. In the dataset there 707 customers who left the company so the ratio of churn in the dataset is 14%.

TABLE I. List of Features of the Churn Dataset.

State	Total eve minutes
Account length	Total eve calls
Area code	Total eve charge
Phone number	Total night minutes
International plan	Total night calls
Voice mail plan	Total night charge
Number vmail messages	Total intl minutes
Total day minutes	Total intl calls
Total day calls	Total intl charge
Total day charge	Number customer service calls

VI. Evaluation Measures

In this paper, the precision, recall, accuracy and F-measure performance criteria are used to evaluate the XGBoost and the selected, well-known benchmark classifiers in churn prediction for the telecommunication sector. The four performance evaluation criteria are computed based on the confusion matrix presented in Table I. The false positive and true positive cases are denoted as FP and TP, respectively, while the false negative and true negative cases are abbreviated as FN and TN, respectively [9] (Table II).

On the other hand, the precision is the percentage of correctly-predicted positive cases. It is computed using the following equation [9]:

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

TABLE II. Confusion Matrix

		Predicted Class	
		Non-churners	Churners
Actual Class	Non-churners	TN	FP
	Churners	FN	TP

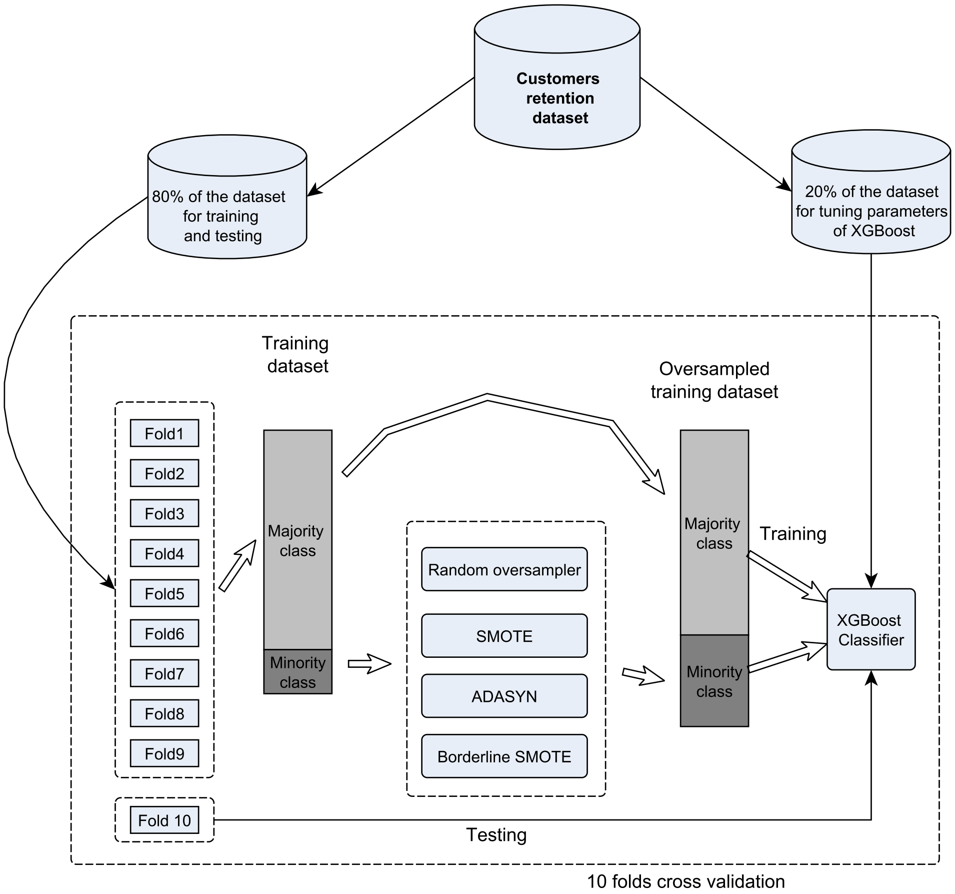

Fig. 1. Framework of XGBoost with Oversampling for Customer Churn Prediction.

In other respects, recall expresses the percentage of the correctly-predicted positive cases. It is calculated using the equation [9]:

$$Recall(Sensitivity) = \frac{TP}{TP + FN} \tag{6}$$

Meanwhile, the accuracy represents the percentage of the total correct predictions. It is given by the equation [9]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

Recall or precision alone can not describe efficiency of the classifier owing to that a good performance according to one of these two indices does not necessarily mean good performance according to the other. On account of this, the F-measure, which is common combination of these two metrics, is frequently employed as a single metric for evaluation of performance of the classifier. This measure is defined as the harmonic mean of recall and precision [9]:

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{8}$$

The closer the F-measure to one, the better. An F-measure value that is close to one means that a good combined recall and precision is provided by the classifier under evaluation [9].

VII. Experiments and Results

In this experiment we used python3 and the following libraries : `Scikit-learn` is a library in Python that provides many unsupervised and supervised machine learning algorithms. This library is built based on other popular libraries like `NumPy`, `Pandas`, and `Matplotlib`. For the oversampling algorithms, the `Imbalanced-learn` is used. `Imbalanced-learn` imbalanced-learn is a Python package that provides a set of resampling algorithms commonly used for imbalanced datasets.

A. Experiments Setup

For parameter tuning of the machine learning classifiers, one fifth of the dataset which is 1000 instances are used for this task. To perform this task, the `GridSearchCV` from `sklearn` library in Python is used. `GridSearchCV` function is applied with 3 folds cross validation to find the best parameters of Random Forest, SVM, XGBoost, Logistic Regression and SGD classifier. The ranges of the parameters to be searched by the `GridSearchCV` are specified as given in Table III. The best parameters of this experiment are listed in Table IV.

The rest of the dataset which are 1000 instances (customers) are used for training and testing the machine learning classifiers using 10 folds cross-validation.

TABLE III. RANGES OF PARAMETERS FOR GRID SEARCH

Classifier	Range of Parameters
RandomForest	N_estimators: start:200,end:2000
	Max_features : ['auto', 'sqrt']
	Max_depth:start:10,end:110
	Min_samples_split : [2, 5]
SVM	C: [0.1, 1, 10, 100, 1000]
	Gamma: [1, 0.1, 0.01, 0.001, 0.0001]
XGboost	Min_child_weight :[1, 5, 10]
	Gamma: [0.5, 1, 1.5]
	Subsample: [0.6, 0.8, 1.0]
	Colsample_bytree: [0.6, 0.8, 1.0]
	Max_depth: [3, 4, 5]
LogisticRegression	C: [1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 1e0]
SGDClassifier	Alpha: [1e-4, 1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2, 1e3]

TABLE IV. BEST PARAMETERS FOR MACHINE LEARNING CLASSIFIERS USING GRID SEARCH METHOD.

Classifier	Best parameters
Random Forest	Max_depth: 60
	Max_features: sqrt
	Min_samples_split: 2
	N_estimators: 1100
SVM	C: 0.1
	Gamma: 0.0001
	Kernel: rbf
XGboost	Colsample_bytree: 0.8
	Gamma: 1.5
	Max_depth: 5
	Min_child_weight: 1
	Subsample: 1.0
Logistic Regression	C: 1.0
	Penalty: l2
SGD	Alpha: 1.0
	Loss: log
	Penalty: l2

B. Comparison of XGBoost with Other Classifiers

In this experiment the XGBoost and the other machine learning classifiers are applied on the dataset to build customers retention models but without applying any balancing methods. The result of this experiment are given in Table V. From the result we can see that XGBoost and Random Forest performed much better than the other classifiers which are SVM, Logistic Regression, and SGDClassifier in all measures especially the F1 measure. On the other side there is small difference between the results of the XGBoost and Random Forest with the small advantage for XGBoost.

C. Comparison of XGBoost with Other Classifiers after using Weighting Method

In this experiment we study the effect of the class balancing method on the results of machine learning classifiers that were applied in the previous experiment. The results of this experiment are given in Table VI. We can see that performance of the SVM, Logistic Regression, and SGD classifier was improved. However that the performance of the XGBoost and Random Forest is still much more better than the performance of the other classifiers. On the other side the class balancing method has not improved the accuracy and F1 measure of Random Forest and XGBoost.

D. XGBoost Combined with Oversampling Methods

In this section, the performance of XGBoost combined with oversampling methods is experimented. The oversampling methods are: Random oversampler, SMOTE, ADASYN, and Borederline SMOTE. All of these oversampling are tested with XGBoost at different oversampling ratios starting from 20% until 100%.

Fig. 2 shows the results of Random oversampler combined with XGBoost. It can be seen that the recall increases by the increase of oversampling ratio until it reaches around 81% at oversampling ratio 100%. The best F1 measure was reached at 40%. This increase in F1 and recall decreased the ratio of precision from around 91% to around 84%.

Fig. 3 shows the results of SMOTE combined with XGBoost. It can be seen that the recall increases by the increase of oversampling ratio until it reaches around 78% at oversampling ratio 100%. The best F1 measure was reached at 20%. This increase in F1 and recall decreased the ratio of precision from around 93% to around 85%.

Fig. 4 shows the results of ADASYN combined with XGBoost. It can be seen that the recall is almost constant. The best F1 measure was reached at 40%. This increase in F1 and recall decreased the ratio of precision from around 91% to around 87%.

Fig. 5 shows the results of BorderLine combined with XGBoost. It can be seen that the recall increases by the increase of oversampling ratio until it reaches around 78% at oversampling ratio 100%. The best F1 measure was reached at 20%. This increase in F1 and recall decreased the ratio of precision from around 93% to around 86%.

VIII. CONCLUSIONS AND FUTURE WORKS

In this work, an approach based on Gradient Boosted Trees algorithm with oversampling methods is proposed for predicting customer retention in telecommunication companies. In this approach four common and well-regarded oversampling methods are used and compared which are: random oversampling, SMOTE, ADASYN, and Borderline SMOTE. The first part of the experiments showed that Gradient Boosted Trees without oversampling outperforms other popular classifies including SVM, Random Forests, Logestic Regression and SGD classifier. In the second part of the experiments, the oversampling methods were applied at different oversampling ratios. The experiments reveal that the oversampling methods improve the performance of Gradient Boosted Trees in predicting the churn class and the best F-measure value (which is around 84%) can be reached with SMOTE method at oversampling ratio of 20%.

TABLE V. COMPARISON OF XGBOOST WITH OTHER CLASSIFIERS.

	Accuracy	Precision	Recall	F1 measure
RandomForest	0.955 (0.008)	0.936 (0.044)	0.743 (0.053)	0.827 (0.033)
SVM	0.827 (0.021)	0.351 (0.065)	0.225 (0.056)	0.272 (0.058)
XGboost	0.956 (0.009)	0.924 (0.052)	0.752 (0.057)	0.829 (0.052)
LogisticRegression	0.864 (0.016)	0.618 (0.154)	0.204 (0.042)	0.302 (0.052)
SGD	0.801 (0.193)	0.552 (0.208)	0.225 (0.275)	0.223 (0.120)

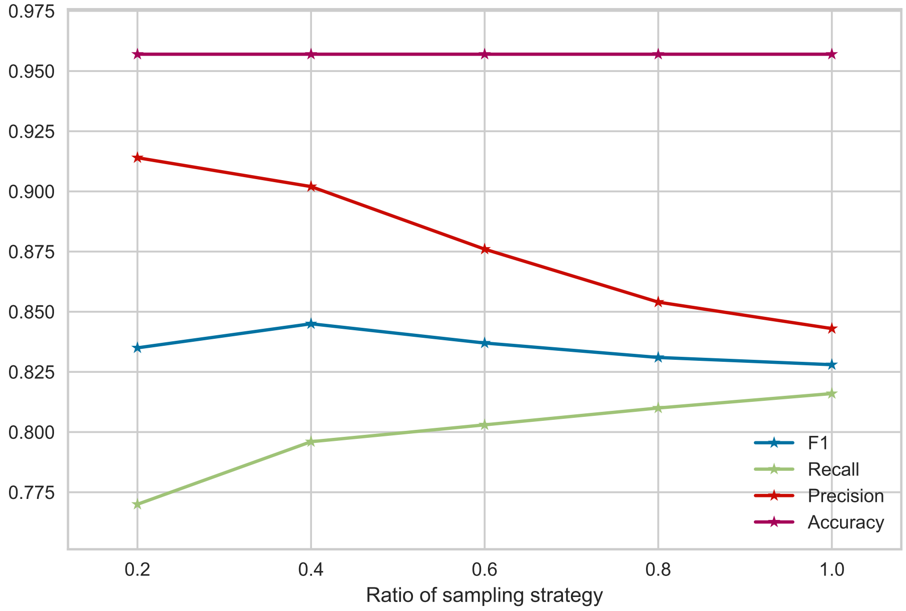

Fig. 2. Effect of Sampling Strategy Ratio of the Random Oversampler on Evaluation Measures of XGBoost.

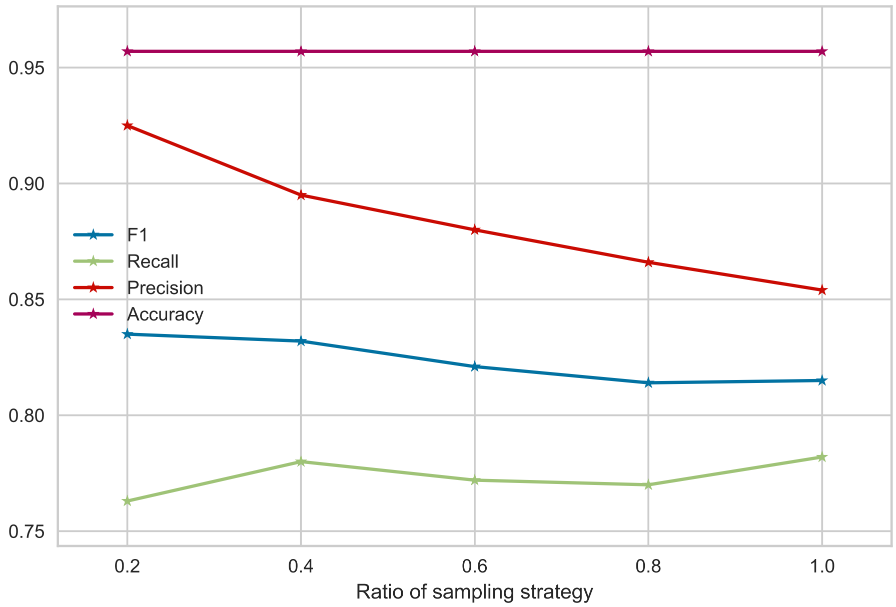

Fig. 3. Effect of Sampling Strategy Ratio of the SMOTE on Evaluation Measures of XGBoost.

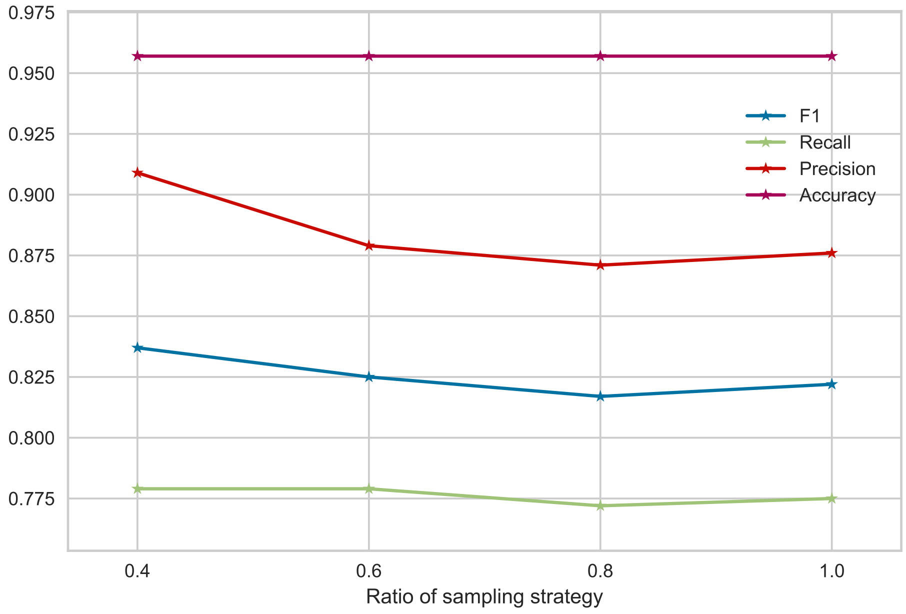

Fig. 4. Effect of Sampling Strategy Ratio of ADASYN on Evaluation Measures of XGBoost.

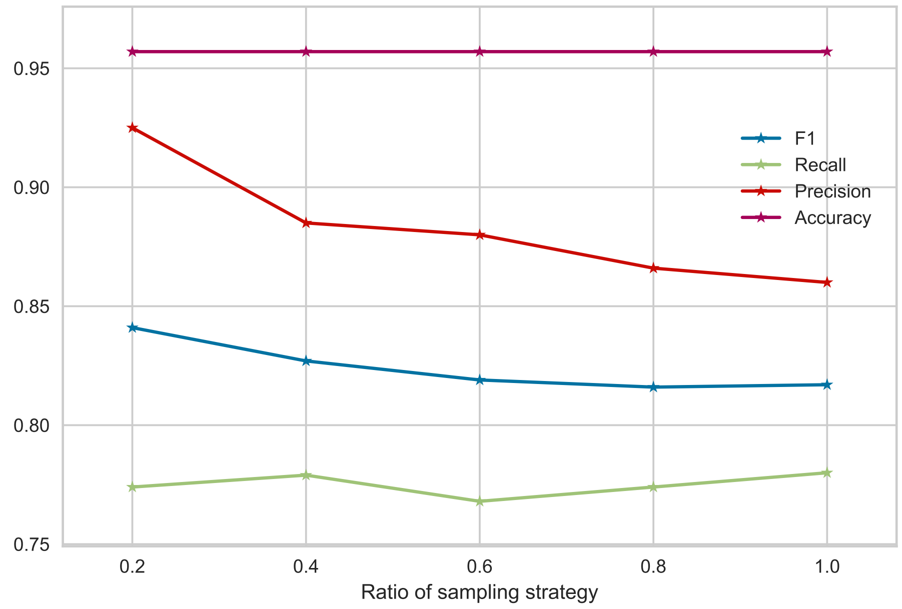

Fig. 5. Effect of Sampling Strategy Ratio of BorderLine SMOTE on Evaluation Measures of XGBoost.

TABLE VI. Comparison of XGBoost with Other Classifiers after applying Class Weighting Method.

	Accuracy	**Precision**	**Recall**	**F1 measure**
Random Forest	0.955 (0.008)	0.902 (0.042)	0.770 (0.057)	0.830 (0.044)
SVM	0.768 (0.023)	0.286 (0.036)	0.398 (0.078)	0.330 (0.043)
XGboost	0.948 (0.009)	0.822 (0.059)	0.818 (0.048)	0.819 (0.045)
LogisticRegression	0.773 (0.014)	0.366 (0.048)	0.770 (0.037)	0.495 (0.048)
SGD	0.519 (0.247)	0.230 (0.120)	0.608 (0.294)	0.273 (0.050)

REFERENCES

[1] A. K. Ahmad, A. Jafar, and K. Aljoumaa, “Customer churn prediction in telecom using machine learning and social network analysis in big data platform,” *arXiv preprint arXiv:1904.00690*, 2019.

[2] C.-P. Wei and I.-T. Chiu, “Turning telecommunications call details to churn prediction: a data mining approach,” *Expert systems with applications*, vol. 23, no. 2, pp. 103–112, 2002.

[3] H. Faris, “A hybrid swarm intelligent neural network model for customer churn prediction and identifying the influencing factors,” *Information*, vol. 9, no. 11, p. 288, 2018.

[4] V. Mahajan, M. Richa, and M. Renuka, “Review on factors affecting customer churn in telecom sector,” *International Journal of Data Analysis Techniques and Strategies*, vol. 9, no. 2, pp. 122–144, 2017.

[5] A. Rodan, A. Fayyoumi, H. Faris, J. Alsakran, and O. Al-Kadi, “Negative correlation learning for customer churn prediction: A comparison study,” *The Scientific World Journal*, vol. 2015, 2015.

[6] K. Coussement, D. F. Benoit, and D. Van den Poel, “Improved marketing decision making in a customer churn prediction context using generalized additive models,” *Expert Systems with Applications*, vol. 37, no. 3, pp. 2132–2143, 2010.

[7] A. Sharma and P. Prabin, “A neural network based approach for predicting customer churn in cellular network services,” *International Journal of Computer Applications*, vol. 27, no. 11, pp. 26–31, 2011.

[8] 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.

[9] 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.

[10] A. Idris, M. Rizwan, and A. Khan, “Churn prediction in telecom using random forest and pso based data balancing in combination with various feature selection strategies,” *Computers & Electrical Engineering*, vol. 38, no. 6, pp. 1808–1819, 2012.

[11] A. Idris, A. Iftikhar, and Z. ur Rehman, “Intelligent churn prediction for telecom using gp-adaboost learning and pso undersampling,” *Cluster Computing*, vol. 22, no. 3, pp. 7241–7255, 2019.

[12] B. Zhu, B. Baesens, and S. K. vanden Broucke, “An empirical comparison of techniques for the class imbalance problem in churn prediction,” *Information sciences*, vol. 408, pp. 84–99, 2017.

[13] A. Amin, S. Anwar, A. Adnan, M. Nawaz, N. Howard, J. Qadir, A. Hawalah, and A. Hussain, “Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study,” *IEEE Access*, vol. 4, pp. 7940–7957, 2016.

[14] A. Hanif and N. Azhar, “Resolving class imbalance and feature selection in customer churn dataset,” in *2017 International Conference on Frontiers of Information Technology (FIT)*. IEEE, 2017, pp. 82–86.

[15] H. Faris, “Neighborhood cleaning rules and particle swarm optimization for predicting customer churn behavior in telecom industry,” *International Journal of Advanced Science and Technology*, vol. 68, no. 1, pp. 11–22, 2014.

[16] A. Idris, A. Khan, and Y. S. Lee, “Intelligent churn prediction in telecom: employing mrmr feature selection and rotboost based ensemble classification,” *Applied intelligence*, vol. 39, no. 3, pp. 659–672, 2013.

[17] 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)*. IEEE, 2013, pp. 131–136.

[18] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[19] H. He, Y. Bai, E. A. Garcia, and S. Li, “Adasyn: Adaptive synthetic sampling approach for imbalanced learning,” in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, 2008, pp. 1322–1328.

[20] H. Han, W.-Y. Wang, and B.-H. Mao, “Borderline-smote: a new over-sampling method in imbalanced data sets learning,” in *International conference on intelligent computing*. Springer, 2005, pp. 878–887.

[21] T. Chen, T. He, M. Benesty, V. Khotilovich, and Y. Tang, “Xgboost: extreme gradient boosting,” *R package version 0.4-2*, pp. 1–4, 2015.

[22] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016, pp. 785–794.