Enhancing Customer Churn Prediction With Resampling: A Comparative Study

Jia-Xuan Ong [1] , Gee-Kok Tong [1] , Kok-Chin Khor [2] , Su-Cheng Haw [1]

[1] *Faculty of Computing & Informatics, Multimedia University, Persiaran Multimedia, 63100 Cyberjaya, Selangor, Malaysia*
[2] *Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul Rahman, Jalan Sungai Long, Bandar Sungai Long, 43200 Kajang, Selangor, Malaysia*

Abstract **– In this competitive business world, accurately predicting customer churn is crucial to maintaining and preventing revenue loss. However, due to the imbalanced nature of customer churn data, traditional machine learning algorithms often fail to identify churned customers accurately. This has led to exploring resampling techniques, demonstrating their efficacy in addressing this issue. However, current studies in the customer churn prediction field frequently overlook the untapped potential of comprehensive investigation and comparison of resampling techniques. Instead of exploring and comparing various resampling methods, many studies predominantly rely on a single resampling method, such as SMOTE. Hence, this paper aims to compare and evaluate the effectiveness of several resampling methods, including oversampling, undersampling, and hybrid techniques. We utilized the benchmark dataset, telecommunication customer churn, from IBM Watson, where approximately 26.5% of the customers have churned, indicating that the data is imbalanced. Our results demonstrate that the combination of random forest with a hybrid sampling method – SMOTE-ENN obtained the best result. The combination yields an F1 score of 95.3% and an accuracy of 96.0%, surpassing the studies that utilized the same dataset. This highlights the benefits of comparing resampling techniques in predicting customer churn, specifically in imbalanced datasets.**

Keywords **– Customer churn prediction, imbalance datasets, resampling, oversampling, undersampling.**

DOI: 10.18421/TEM133-20
https://doi.org/10.18421/TEM133-20

Corresponding author: Gee-Kok Tong ,
Faculty of Computing & Informatics, Multimedia University, Persiaran Multimedia, 63100 Cyberjaya, Selangor, Malaysia
Email: gktong@mmu.edu.my

Received: 17 January 2024.
Revised: 09 May 2024.
Accepted: 05 July 2024.
Published: 27 August 2024.

1. Introduction

Customer churn, commonly referred to as customer attrition, is defined as the tendency of customers to discontinue or terminate engaging with a company. Almost 1.5 million customers are churning in a year, and the number rises yearly [1]. Industries such as banking, telecommunications, airlines, and e-commerce, face challenges in retaining their customers as they can change their preferred services and providers anytime and anywhere due to the numerous choices available in the competitive market. For example, in the telecommunications industry, customers might shift to other competitors due to distinct reasons, such as lower cost of product plans, stronger internet connections in certain locations, higher service quality, etc.

Maintaining great customer connections is critical for businesses because consumers drive sales and profitability. The cost needed to gain new customers is about six times higher than retaining the ones the companies already have [2]. This is because companies need to spend a lot on marketing. For instance, most companies will advertise their products on television programs, websites, and social media to attract new customers, which is not at a small cost. So, customer churn is undesirable in the fluctuating and dynamic business industries. It might be a barrier for a company to expand their revenue.

On the other hand, in customer churn prediction, certain research works focused only on implementing predictive models when facing imbalanced datasets, with little attention paid to exploring dataset balancing approaches [3], [4], [5], [6].

This is noteworthy because resampling methods have demonstrated significant potential for enhancing the performance of these predictive models. Furthermore, among [7], [8], [9], [10] who have incorporated resampling into their studies, there is a common tendency to rely solely on a single method, such as SMOTE, thus limiting the scope of their investigation into alternative resampling strategies. Furthermore, it is worth highlighting that even in cases where studies, such as [11], [12], [13], [14] have compared various resampling methods, they have not conducted a comprehensive comparison encompassing oversampling, undersampling, and hybrid resampling techniques in the context of customer churn prediction. This lack of thorough analysis suggests a further research gap in the field. Thus, there is a need for more investigations that rigorously evaluate and compare different resampling strategies to enhance the accuracy and reliability of churn prediction models.

This research uses the IBM dataset to find a robust and accurate machine-learning model for predicting customer churn in the telecommunication industry. A comprehensive approach that includes six baseline machine learning algorithms for predicting customer churning was developed to achieve this. Multiple resampling methods, such as oversampling, undersampling, and hybrid methods, were used to deal with the imbalanced data.

2. Methodology

This section will describe the research methodology of this paper. Figure 1 provides an overview of the research framework employed in this study. The research process begins with data collection, followed by comprehensive data preprocessing and feature selection steps. To ensure the reliability of the experiments, we divided the dataset using the train-test split approach, allocating 80% of the data for training purposes and reserving the remaining 20% for testing. Subsequently, the training set underwent six resampling methods, creating six resampled datasets. Six different machine-learning models were then constructed based on these resampled datasets. These models were built to capture and analyze the patterns in the data to predict customer churn effectively. Utilizing different resampling methods enables the investigation of the impact of each method on the model performance. By comparing the prediction outcomes of using the models on the resampled datasets, we can evaluate the effectiveness of each resampling method in addressing the class imbalance issue commonly encountered in customer churn prediction tasks.

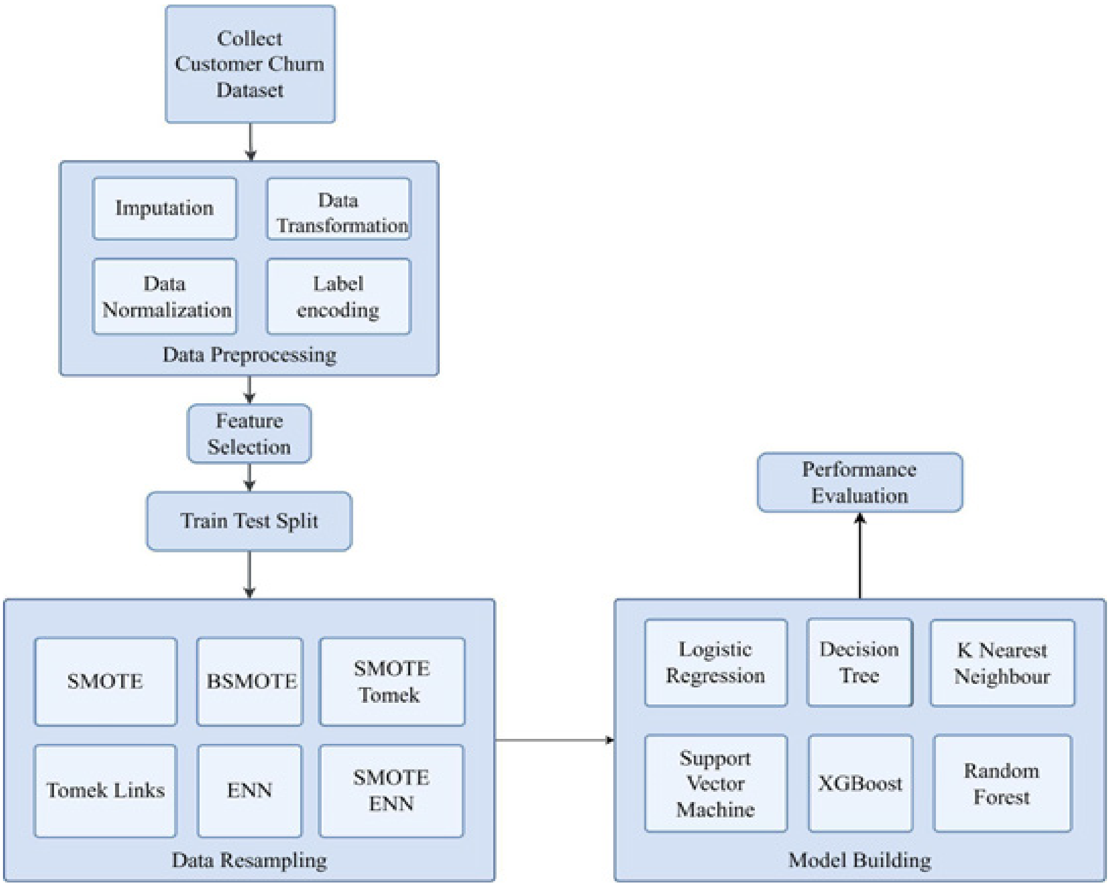

Figure 1. Overview of the research methodology

2.1. Dataset Used

This study uses the IBM dataset, a publicly available collection of customer churning data in the telecommunications industry. It was downloaded from Kaggle https://www.kaggle.com/datasets/yeanzc/telcocustomer-churn-ibm-dataset. Several researchers utilized this dataset for their studies on customer churn, such as [15], [16], [17]. This dataset is from a telecommunications provider that offered Internet and home phone service in California. The raw IBM dataset consists of 7,043 samples with 33 features. Each sample represents a customer. The features include demographic data, details about the customer's internet service, contract terms, billing information, and charges. The target label for analysis is the binary "Churn Label" variable, which indicates whether or not the customer churned (terminated their service).

2.2. Data Preprocessing

Data preprocessing plays a critical role in data analysis and machine learning projects. In this study, we carried out data transformation involving handling missing or damaged data and converting data into a suitable format for machine learning algorithms. Missing values were carefully imputed to avoid bias and maintain prediction accuracy, while categorical variables were label-encoded to convert them into numerical values. Additionally, continuous numerical features (Total Charges, Monthly Charges, Tenure Months) were normalized using Min-Max Scaler to fit within a predefined range, typically 0-1. These preprocessing steps ensure that the data is appropriately prepared for the machine learning algorithms used in this study.

2.3. Feature Selection

One important step is dropping unnecessary columns to reduce data dimensionality and improve performance. In this study, irrelevant features are removed manually instead of applying specific feature selection methods. For example, features like customer ID and geolocation information were dropped because they do not carry any meaningful information to the prediction. Besides, churn reason was also removed as it is a variable that is not known until the customer has already churned. It is, therefore, not a variable that can be used to predict future customer churn. Including these variables in the dataset for churn prediction can lead to overfitting, as the model will likely learn to predict churn based on the reason for the churn rather than learning more general patterns that can be used to predict churn for new customers.

Hence, by removing the irrelevant variables, we aim to streamline the dataset and focus on the important features.

2.4. Data Resampling

The IBM dataset included information on over 7,000 customers, with approximately 26.5% having churned, indicating the data is imbalanced. Figure 2 illustrates the distribution of the data points of the original training set. To solve the class imbalance problem, we investigated using various resampling methods such as oversampling, undersampling, and hybrid methods to the training set. Table 1 summarizes the resampling strategies and associated parameters. Both oversampling techniques were used with a sampling strategy of 0.7, which means that the minority class was oversampled in order to establish a 0.7 ratio between the minority and majority classes. The k_neighbors option was set to 5, indicating the number of nearest neighbors utilized to generate synthetic samples. Next, the sampling_strategy for the undersampling methods were set as 'not minority' which will preserve all samples from the majority class that are not involved in the algorithm. While for ENN, the parameter n_neighbors was set to 3 which indicate instances from the majority class that were misclassified by 3 nearest neighbors in the minority class will be eliminated, while n_jobs was set to -1, indicates that the algorithm should use as many cores as are available on the machine. Table 2 displays the overall distribution of churners and non-churners after resampling.

Table 1. Overview of resampling methods with the corresponding parameter

Resampling Method	Parameter
SMOTE	sampling_strategy=0.7, k_neighbors=5
BSMOTE	sampling_strategy=0.7, k_neighbors=5
Tomek Links	sampling_strategy='not minority'
ENN	n_neighbors=3, n_jobs=-1, sampling_strategy='not minority'
SMOTE Tomek	sampling_strategy=0.7, smote=K_{smote}5
SMOTE-ENN	sampling_strategy=0.7, smote=K_{smote}5

Table 2. The distribution of churners and non-churners in each dataset

Dataset	Non-Churners	Churners	Total
Imbalanced	4,139 (73.5%)	1,495 (26.5%)	5,634
SMOTE	4,139 (62.5%)	2,483 (37.5%)	6,622
BSMOTE	4,139 (62.5%)	2,483 (37.5%)	6,622
Tomek Links	3,719 (71.3%)	1,495 (28.7%)	5,214
ENN	2,374 (61.4%)	1,495 (38.6%)	3,869
SMOTE-ENN	2,266 (59.9%)	1,514 (40.1%)	3,780
SMOTE Tomek	3,838 (63.8%)	2,182 (36.2%)	6,018

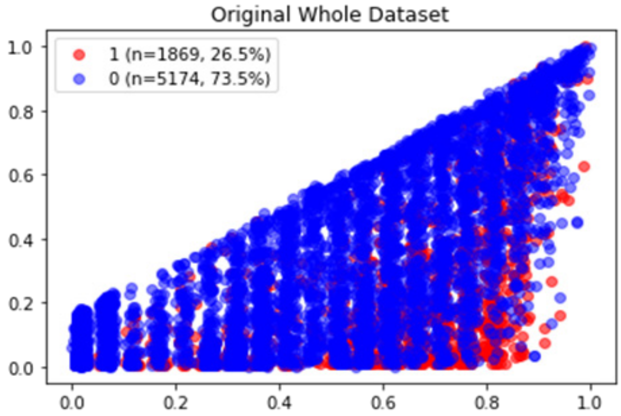

Figure 2. Data points distribution of the minority (churners, red) and majority (non-churners, blue) classes in the original dataset

2.4.1. Synthetic Minority Oversampling Technique

Synthetic Minority Oversampling Technique (SMOTE) is a method that develops synthetic minority class samples by generating new data points comparable to existing minority class samples [18]. To over-sample the minority class, SMOTE synthesizes new samples by selecting a minority class sample and its k nearest neighbors, and then generating new samples at random points along the lines connecting the original sample and its neighbors. Researchers can specify the specific number of neighbors to use and the level of over-sampling. This process is repeated for all minority class samples, and the resulting synthesized samples are added to the original dataset. Figure 3 displays the distribution of data points after performing SMOTE oversampling.

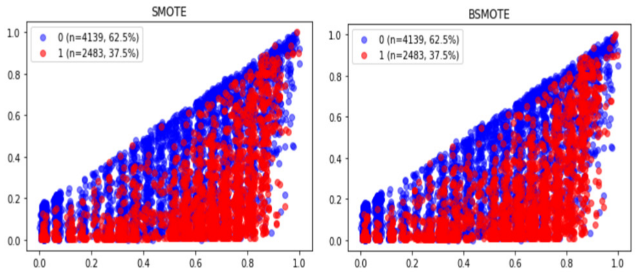

Figure 3. Data points distribution of the minority and majority class after performing oversampling with SMOTE and BSMOTE, respectively

2.4.2. Boderline SMOTE

Borderline SMOTE (BSMOTE) is a variant of SMOTE that was created to address the issue of overgeneralization in minority class samples by synthesizing new samples closer to the decision boundary between the minority and majority classes. This is performed by finding "borderline" samples, minority class samples at the decision boundary, and producing new samples in their surroundings [19]. Borderline samples are important because most classification algorithms strive to understand the boundaries of each class as precisely as possible during the training process to obtain better predictions. The instances on the borderline and those adjacent are more likely to be misclassified than those further away, making them more vital for classification purposes. Figure 3 visually represents the data points of the majority and minority classes after performing BSMOTE.

2.4.3. Tomek Links

Tomek links is an undersampling method that seeks to improve the performance of a classification model by removing noisy or borderline examples from the majority class [20]. Tomek links might be used as an undersampling or post-process cleaning step [21]. Only samples from the majority class are expelled when employed as an undersampling method. While it is employed as a post-process cleaning step, both samples are eliminated. Tomek links happen when two data points, where one from the majority class and one from the minority class, are the closest neighbors to each other. If two occurrences create a Tomek connection, either one is noise, or both are close to a boundary. The goal of Tomek link is to identify and remove the majority class that created a Tomek connection. This is because they are considered noisy, possibly contributing to misclassification. Hence, by deleting these data points, well-defined clusters may be established, leading to higher classification performance. Figure 4 shows the distribution of the data points after implementing Tomek Links.

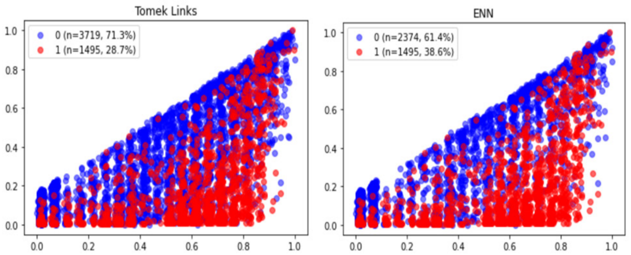

Figure 4. Data points distribution of the minority and majority class after performing undersampling with Tomek Links and ENN, respectively

2.4.4. Edited Nearest Neighbour

Edited Nearest Neighbour (ENN) is also an undersampling method used to address class imbalance in machine learning. It is a method that implements the K-Nearest Neigbours algorithm to decrease the data in the majority class [22]. Unlike Tomek links, ENN works by detecting possibly mislabeled examples inside the majority class by directly analyzing the class labels of instances and their nearby neighbours. ENN looks for cases where the class label does not match the majority class label of their neighbours, suggesting potential noise or mistakes. Once discovered, these possibly mislabeled examples are removed from the majority class. This elimination step removes occurrences that may lead to misclassification or confuse the classification model. ENN strives to increase overall performance, particularly for the minority class, by minimizing noise within the majority class. Figure 4 illustrates the undersampled data points using ENN.

2.4.5. SMOTE-ENN

SMOTE-ENN is a hybrid of oversampling and undersampling methods that combine the SMOTE algorithm with ENN. As mentioned earlier, SMOTE oversamples the minority class by synthesizing new samples, selecting a minority class sample and its k nearest neighbors, and then generating new samples at random points along the lines connecting the original sample and its neighbors. Then, ENN undersamples the minority class by detecting possibly mislabeled examples inside the majority class and removing them afterward. Hence, SMOTE-ENN leverages the strength of both oversampling and undersampling methods to combine the synthetic minority class instances generated by SMOTE with the reduced majority class instances obtained from ENN. Figure 5 visually represents the distribution of data points after performing the hybrid resampling using SMOTE-ENN.

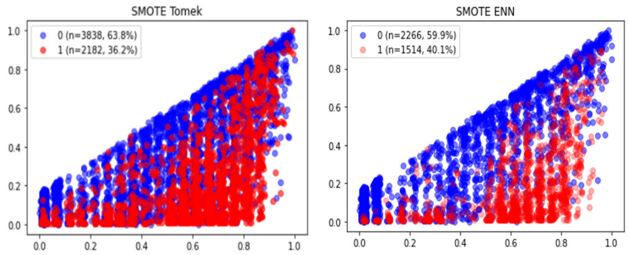

Figure 5. Data points distribution of the minority and majority class after performing hybrid sampling with SMOTE Tomek and SMOTE-ENN, respectively

2.4.6. SMOTE Tomek

SMOTE Tomek is a combination of oversampling and undersampling methods that was first introduced by [19]. It is a hybrid resampling method that combines Tomek links and SMOTE. As described earlier, SMOTE oversamples the minority class by synthesizing new samples, while Tomek links undersample by removing noisy or borderline examples from the majority class. In short, SMOTE-Tomek links approach benefits from the oversampling and undersampling, thus aids in reducing class distribution imbalance, making the dataset more acceptable for training classifiers. It can increase classifier performance by providing a balanced training set and increasing class separation. Figure 5 shows the distribution of the majority and minority class after implementing SMOTE-Tomek links.

2.5. Classifier

This section will delve into a detailed discussion of the six distinct classifiers that were employed in this research.

2.5.1. Decision Tree

Decision tree (DT) is one of the most popular and powerful classification algorithms [23]. The DT approach turns data into a tree that symbolizes the rules, which can be easily conveyed by natural language. It is a flowchart-like tree structure, with each internal node representing a feature (or attribute), each decision node representing a rule, and each leaf node representing the conclusion. The root node is the node at the top of a DT. Based on the values of the input features, it learns to partition the data into subsets.

2.5.2. K Nearest Neighbour

K nearest neighbour (KNN) is an algorithm that presumes related items are close to one another [24]. Besides, [25] proposed that when new unknown data occurs, which can be called a test sample, KNN sorts it into the class with the highest similarity by determining the k closest samples from an existing dataset.

In other words, the classification method assigns the test sample group to the class with the highest likelihood using the k training samples that are the test sample's closest neighbours.

2.5.3. Logistic Regression

Logistic regression (LR) is a technique for categorizing the relationship between numerous independent factors and dependent variables. It is a statistical approach for categorizing data. It is a generalized linear model (GLM) form that models a binary or multi-class dependent variable using a logistic function. It is frequently applied to the medical and social sciences, such as customer churn prediction [26]. There are several types of LR, such as binary, multinomial, and ordinal, but only binary LR will be discussed in the following paper. This is because the target variable, Churn, is a binary variable that consists of Yes (1) and No (0).

2.5.4. Support Vector Machine

Support vector machine (SVM) is a machine learning technique that creates predictions and generalizations about new data by performing learning on data when the distribution is uncertain. It is a linear model that can deal with regression and classification problems. SVM divides classes with the greatest feasible margin of error by employing a hyperplane created by a collection of support vectors to classify data points. In other words, the SVM algorithm draws a line dividing the data into classes [27]. For example, SVM will split all the data points in the n-dimensional plane into churner and non-churner groups based on the maximum marginal hyperplane.

2.5.5. Random Forest

Random forest (RF) is an ensemble learning approach that uses a forest of decision trees to produce predictions [28]. It is a supervised machine learning technique that can be used for classification and regression problems. Creating many decision trees and combining their forecasts into one big prediction is the underlying notion behind random forest. A random subset of the features and a separate subset of the training data referred to as a bootstrap sample, are used to generate each decision tree in the forest. "Bootstrap aggregating" or "bagging" refers to this procedure. RF eliminates overfitting, a typical issue with decision trees, by averaging the predictions of several decision trees.

2.5.6. eXtreme Gradient Boosting

XGBoost (eXtreme Gradient Boosting) is a widely used machine learning technique for handling supervised learning issues.

It is a gradient-boosting implementation, an ensemble approach combining numerous weak models to generate a stronger model. Gradient boosting generates a new model by sequentially adding weak models to an ensemble. Each new model is designed to remedy the mistakes committed by prior models. It is good at handling sparse data and is very efficient when working with huge datasets.

2.6. Model Evaluation

The authors examined several measures widely used in classification tasks to assess the performance of our models, including F1 score, recall, precision, accuracy, and AUC ROC. Given the imbalanced character of our dataset, the F1 score was chosen as primary evaluation measure. The F1 score considers precision and recall, making it an appropriate measure for evaluating model performance on imbalanced datasets. Rather than maximizing overall accuracy, we can guarantee that our models perform well across both the positive and negative classes by concentrating on the F1 score. The formula of the performance metrics used is as follows.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

3. Results and Discussion

This section presents the results obtained from a series of experiments and the discussions aimed at validating the resampling methods. As the dataset encountered an imbalance issue, F1 score will be the main metric to evaluate the results.

3.1. Individual Resampling Techniques

Table 3 summarizes the performance of the individual classifier in each resampling method. One of the significant observations from Table 3 is that Tomek links demonstrated the lowest average results out of all resampling methods. This suggests that while designed to solve class imbalance, Tomek links undersampling may not be as helpful as other resampling strategies in enhancing the performance of the models for customer churn prediction.

Besides, SMOTE-ENN consistently outperformed other resampling methods, achieving the highest average performance for all classifiers. Among the models that were trained within the SMOTE-ENN datasets, random forest (RF) obtained a remarkable F1 score of 95.3% and an accuracy of 96.0%.

The superior performance of RF can be attributed to its ensemble nature as it is a bagging algorithm that combines several decision trees, which enables it to learn effectively from the resampled data and make accurate predictions. Furthermore, upon evaluating the outcomes, it is worth noting that while RF had the greatest overall performance in the SMOTE-ENN dataset, XGBoost also performed well. XGBoost obtained an F1 score of 94.7%, an accuracy score of 95.7%, a recall score of 95.1%, and a precision score of 94.3%. This finding implies that combining oversampling (SMOTE) and undersampling (ENN) approaches successfully balanced the class distribution, enhancing prediction accuracy. While these results provide an overview, our subsequent analysis will delve deeper into the specific performance of the models in the minority and majority classes, shedding light on the strengths and limitations of the chosen approach in capturing churned and non-churned customers.

Table 3. The performance (%) of classifiers in each resampling method

			Original	SMOTE	BSMOTE	ENN	Tomek Links	SMOTE Tomek	SMOTE-ENN
Logistic Regression	Overall	F1	61.8	75.0	73.1	85.0	67.9	77.8	91.8
		Precision	66.9	73.0	70.6	86.6	71.3	75.7	91.3
		Recall	57.5	77.2	75.9	83.6	65.0	80.1	92.3
		Accuracy	81.2	78.9	77.1	88.6	82.4	81.5	93.4
	Majority Class	Precision	85.0	83.0	82.0	90.0	86.0	86.0	95.0
		Recall	90.0	80.0	78.0	92.0	89.0	82.0	94.0
	Minority Class	Precision	67.0	73.0	71.0	87.0	71.0	76.0	91.0
		Recall	58.0	77.0	76.0	84.0	65.0	80.0	92.0
Decision Tree	Overall	F1	51.3	69.8	69.0	81.8	59.9	73.7	91.4
		Precision	50.1	70.3	68.8	82.1	59.1	73.2	92.0
		Recall	51.0	70.0	70.7	81.5	59.6	74.7	92.3
		Accuracy	73.8	75.4	74.7	85.6	76.6	78.6	93.4
	Majority Class	Precision	82.0	79.0	79.0	89.0	84.0	82.0	95.0
		Recall	82.0	80.0	77.0	88.0	83.0	82.0	94.0
	Minority Class	Precision	51.0	71.0	68.0	81.0	59.0	73.0	91.0
		Recall	52.0	70.0	70.0	82.0	60.0	74.0	92.0
Support Vector Machine	Overall	F1	55.9	75.5	74.8	84.7	66.2	78.5	93.6
		Precision	65.6	73.0	70.9	88.1	71.6	76.0	93.9
		Recall	48.8	78.5	79.3	81.7	61.7	81.4	93.4
		Accuracy	79.6	79.2	78.0	88.7	82.0	82.1	94.9
	Majority Class	Precision	83.0	84.0	84.0	89.0	85.0	87.0	96.0
		Recall	91.0	80.0	77.0	93.0	90.0	83.0	96.0
	Minority Class	Precision	65.0	73.0	71.0	88.0	71.0	76.0	94.0
		Recall	49.0	78.0	79.0	82.0	62.0	81.0	93.0
K Nearest Neighbour	Overall	F1	53.8	74.7	75.2	86.4	62.4	77.6	94.4
		Precision	55.8	68.5	67.2	87.5	63.0	71.3	92.0
		Recall	52.1	82.3	85.3	85.4	61.9	85.3	97.0
		Accuracy	76.3	77.2	76.8	89.7	78.6	80.2	95.4
	Majority Class	Precision	83.0	86.0	87.0	91.0	85.0	89.0	98.0
		Recall	85.0	74.0	71.0	92.0	85.0	77.0	94.0
	Minority Class	Precision	56.0	69.0	67.0	88.0	63.0	71.0	92.0
		Recall	52.0	82.0	85.0	85.0	62.0	85.0	97.0
Random Forest	Overall	F1	58.0	78.4	77.6	88.0	67.7	81.3	95.3
		Precision	66.0	77.2	76.8	90.3	73.5	81.2	95.6
		Recall	52.0	79.7	79.9	86.2	62.2	82.1	94.9
		Accuracy	80.0	82.0	81.8	90.5	82.8	84.9	96.0
	Majority Class	Precision	84.0	85.0	86.0	91.0	86.0	88.0	97.0
		Recall	90.0	84.0	82.0	94.0	90.0	86.0	97.0
	Minority Class	Precision	66.0	77.0	76.0	90.0	72.0	80.0	95.0
		Recall	53.0	79.0	80.0	86.0	63.0	82.0	95.0
XGBoost	Overall	F1	58.1	76.1	76.2	85.9	65.8	79.4	94.7
		Precision	62.7	75.9	74.9	86.7	68.7	79.6	94.3
		Recall	54.3	77.3	78.3	85.2	63.3	80.0	95.1
		Accuracy	79.2	80.5	80.3	90.5	81.2	83.6	95.7
	Majority Class	Precision	84.0	84.0	84.0	91.0	86.0	86.0	97.0
		Recall	88.0	83.0	82.0	92.0	88.0	86.0	96.0
	Minority Class	Precision	63.0	76.0	75.0	87.0	69.0	79.0	94.0
		Recall	54.0	77.0	78.0	85.0	63.0	80.0	95.0

3.2. Majority and Minority Class Performance

Table 3 also summarizes the class performance which are the majority class (non-churners) and minority class (churners) of all the classifiers. Notably, recall in the minority class is a critical performance metric when it comes to imbalance issues. Lower recall implies that the models struggle to effectively identify customers at risk of churning. Since we already know that SMOTE-ENN is the best-performing resampling method, we need to investigate the minority and majority class performance of the models. We found an interesting observation.

RF achieved impressive results with 95% precision and recall in the minority class and an outstanding 97% precision and recall in the majority class. Although RF is the best-performing model, as we mentioned earlier, it is noteworthy that K Nearest Neighbour (KNN) slightly outperformed RF in the individual metrics for each class. To be more precise, KNN achieved a recall of 97% in the minority class, surpassing RF by 2%, while also achieving a precision of 98% in the majority class, which is 1% higher than RF. This indicates that KNN exhibits strength in identifying more correct churners than RF. Although it seems like KNN outperforms RF, a comprehensive comparison is still necessary to evaluate the performances thoroughly.

KNN demonstrates the highest precision of 98% in the majority class, indicating it can identify non-churned customers correctly. However, it has a lower recall of 94% in the majority class, suggesting that it may miss some non-churned customers. In contrast, RF obtains a precision score of 97% in the majority class, which is 1% lower compared to KNN, yet it can obtain a better recall, accurately identifying a huge proportion of non-churners. In the minority class, KNN achieves a higher recall of 97%, suggesting its ability in capturing churned customers. Nevertheless, it has a lower precision of 92%, implying a higher chance of misclassifying non-churned customers as churned. RF, in comparison, achieves a high precision of 95% in the minority class, ensuring a high level of accuracy in identifying churned customers, and maintaining a recall of 95%, capturing a significant proportion of churned customers. In addition, when XGBoost is compared to RF, it has the same recall in both the minority and majority classes but a lower precision. This means that RF may correctly identify more churned clients than XGBoost.

In short, these findings suggest that RF provides a more balanced performance across both classes, showcasing a reliable ability to identify churned and non-churned customers.

While KNN may excel in certain metrics for individual classes, the overall performance of RF remains more consistent and robust. Therefore, RF with SMOTE ENN is preferable when aiming for a balanced and accurate churn prediction model.

3.3. Comparison with Existing Studies

Table 4 compares the proposed work with other researchers' works on the same IBM telecommunication dataset. Our approach has yielded a good result, surpassing the same field's studies. We achieved a 97.7% F1 score and 98.1% accuracy in predicting customer churn. The selection of models and resampling has played a crucial role in enhancing the prediction accuracy. In short, the outstanding results can demonstrate the superiority of our study in tackling the customer churn prediction problem in the telecommunication industry.

Table 4. Comparison between the proposed work with other researchers' work on the same IBM dataset

Study	Approach	F1 (%)	Accuracy (%)
Our approach	Random Forest with SMOTE-ENN	95.3	96.0
[13]	Hybrid resampling methods with ensemble learning	63.4	80.7
[15]	Multiple classifiers	58.2	79.8
[14]	Multiple classifiers	80.6	81.7

4. Conclusion

Customer churn is a critical metric for businesses as it directly impacts their long-term success and profitability. Through this research, we have addressed the critical challenge of predicting customer churn in an imbalanced dataset, a crucial issue in the telecommunication industry.

By comparing several machine learning models with various resampling methods, we have effectively tackled the class imbalance problem and significantly improved the prediction performance. The result shows the best performance with a 95.3% F1 score that was obtained using the random forest with SMOTE-ENN. Our approach has yielded a good result that surpasses and outperforms the studies in the same field using the same dataset. The efficient implementation of our technique provides practitioners and researchers with significant insights into customer churn prediction, allowing them to make more accurate forecasts and informed decisions.

Moving forward, several future works can be explored.

First and foremost, advanced machine learning algorithms such as neural networks can be implemented to enhance prediction performance. Additionally, rather than relying on the existing resample methods, future work can be focused on proposing and developing a novel algorithm to tackle the class imbalance problem. By addressing these future research objectives, we may continue to enhance the field of customer churn prediction and contribute to creating more accurate and effective ways for businesses to prevent customer churn and build long-term customer relationships

References:

[1]. Kaur, I., & Kaur, J. (2020). Customer Churn Analysis and Prediction in Banking Industry using Machine Learning. In *2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC) 434–437.* IEEE. Doi: 10.1109/PDGC50313.2020.9315761.

[2]. Amin, A., et al. (2019). Cross-company customer churn prediction in telecommunication: A comparison of data transformation methods. *International Journal of Information Management, 46*, 304–319. Doi: 10.1016/j.ijinfomgt.2018.08.015.

[3]. Zhao, H., Zuo, X., & Xie, Y. (2022). Customer Churn Prediction by Classification Models in Machine Learning. In *2022 9th International Conference on Electrical and Electronics Engineering (ICEEE),* 399–407. IEEE. Doi: 10.1109/ICEEE55327.2022.9772553.

[4]. Tang, P. (2020). Telecom Customer Churn Prediction Model Combining K-means and XGBoost Algorithm. In *2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE),* 1128–1131. IEEE. Doi: 10.1109/ICMCCE51767.2020.00248.

[5]. De Caigny, A., Coussement, K., & De Bock, K. W. (2018). A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. *European Journal of Operational Research, 269*(2), 760–772. Doi: 10.1016/j.ejor.2018.02.009.

[6]. Khattak, A., Mehak, Z., Ahmad, H., Asghar, M. U., Asghar, M. Z., & Khan, A. (2023). Customer churn prediction using composite deep learning technique. *Scientific Reports, 13*(1), 17294. Doi: 10.1038/s41598-023-44396-w.

[7]. Singh, M., Singh, S., Seen, N., Kaushal, S., & Kumar, H. (2018). Comparison of learning techniques for prediction of customer churn in telecommunication. In *2018 28th International Telecommunication Networks and Applications Conference (ITNAC),* 1–5. IEEE. Doi: 10.1109/ATNAC.2018.8615326.

[8]. Latheef, J., & Vineetha, S. (2021). LSTM Model to Predict Customer Churn in Banking Sector with SMOTE Data Preprocessing. In *2021 2nd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS),* 86–90. IEEE. Doi: 10.1109/ACCESS51619.2021.9563347.

[9]. Nguyen, N. N., & Duong, A. T. (2021). Comparison of Two Main Approaches for Handling Imbalanced Data in Churn Prediction Problem. *Journal of Advanced Information Technology, 12*(1), 29–35. Doi: 10.12720/jait.12.1.29-35.

[10]. Hammoudeh, A., Fraihat, M., & Almomani, M. (2019). Selective Ensemble Model for Telecom Churn Prediction. In *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, 485–487. IEEE. Doi: 10.1109/JEEIT.2019.8717406.

[11]. Chowdhury, A., Kaisar, S., Rashid, M. M., Shafin, S. S., & Kamruzzaman, J. (2021). Churn Prediction in Telecom Industry using Machine Learning Ensembles with Class Balancing. In *2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE),* 1–6. IEEE. Doi: 10.1109/CSDE53843.2021.9718498.

[12]. De, S., & Prabu, P. (2022). A Sampling-Based Stack Framework for Imbalanced Learning in Churn Prediction. *IEEE Access, 10,* 68017–68028. Doi: 10.1109/ACCESS.2022.3185227.

[13]. Wagh, S. K., Andhale, A. A., Wagh, K. S., Pansare, J. R., Ambadekar, S. P., & Gawande, S. H. (2023). Customer churn prediction in telecom sector using machine learning techniques. *Results in Control and Optimization, 14,* 100342. Doi: 10.1016/j.rico.2023.100342.

[14]. Salunkhe, U. R., & Mali, S. N. (2018). A Hybrid Approach for Class Imbalance Problem in Customer Churn Prediction: A Novel Extension to Under-sampling. *International Journal of Intelligent Systems and Applications, 10*(5), 71–81. Doi: 10.5815/ijisa.2018.05.08.

[15]. Kimura, T. (2022). Customer Churn Prediction With Hybrid Resampling And Ensemble Learning. *Journal of Management Information & Decision Sciences*, *25*(1).

[16]. Lalwani, P., Mishra, M. K., Chadha, J. S., & Sethi, P. (2022). Customer churn prediction system: A machine learning approach. *Computing, 104*(2), 271–294. Doi: 10.1007/s00607-021-00908-y.

[17]. Pamina, J., Raja, B., SathyaBama, S., Sruthi, M. S., & VJ, A. (2019). An effective classifier for predicting churn in telecommunication. *Jour of Adv Research in Dynamical & Control Systems*, *11*.

[18]. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research, 16*, 321–357. Doi: 10.1613/jair.953.

[19]. Han, S. S., et al. (2020). Augmented Intelligence Dermatology: Deep Neural Networks Empower Medical Professionals in Diagnosing Skin Cancer and Predicting Treatment Options for 134 Skin Disorders. *Journal of Investigative Dermatology, 140*(9), 1753–1761. Doi: 10.1016/j.jid.2020.01.019.

[20]. Tomek, L. (1976). Two Modifications of CNN. *IEEE Transactions on Systems, Man, and Cybernetics, 6*(11), 769–772. Doi: 10.1109/TSMC.1976.4309452.

[21]. Batista, G. E. A. P. A. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter, 6*(1), 20–29.
Doi: 10.1145/1007730.1007735.

[22]. Wilson, D. L. (1972). Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. *IEEE Transactions on Systems, Man, and Cybernetics, 2*(3), 408–421. Doi: 10.1109/TSMC.1972.4309137.

[23]. Charbuty, B., & Abdulazeez, A. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. *Journal of Applied Science and Technology Trends, 2*(1), 20–28.
Doi: 10.38094/jastt20165.

[24]. Harrison, O. (2018). *Machine Learning Basics with the K-Nearest Neighbors Algorithm*. Medium. Retrieved from:
https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761 [accessed: 04 Januay 2024].

[25]. Rahman, M., & Kumar, V. (2020). Machine Learning Based Customer Churn Prediction In Banking. In *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA),* 1196–1201. IEEE.
Doi: 10.1109/ICECA49313.2020.9297529.

[26]. Çelik, O., & Osmanoglu, U. O. (2019). Comparing to techniques used in customer churn analysis. *Journal of Multidisciplinary Developments*, *4*(1), 30-38.

[27]. Reza, D. S. A. A., Ullah, M. A., Mallick, U. K., & Rony, M. A. T. (2021). A Machine Learning Approach to Identify Customer Attrition for a Long Time Business Planning. In *2021 5th International Conference on Electrical Information and Communication Technology (EICT),* 1–6. IEEE.
Doi: 10.1109/EICT54103.2021.9733713.

[28]. Breiman, L. (2001). Random Forests. *Machine Learning, 45*, 5–32.
Doi: https://doi.org/10.1023/A:1010933404324