ISSN:2731-538X | E-ISSN:2731-5398
Vol. 19 No. 01 (2025)

ACCURACY ENHANCEMENT OF MACHINE LEARNING MODEL BY HANDLING IMBALANCE DATA

Shaik Mohammed Imran1
Research Scholar, Department of CSE,
Hindustan Institute of Technology & Science, Chennai.
smimran.it@gmail.com

Dr. Angelina Geetha2
Professor,
Department of CSE,
Hindustan Institute of Technology and Science,
angelinag@hindustanuniv.ac.in

Abstract— As big data has growing rapidly, a new era of scientific research has emerged. Uneven values of response variable distributed, or class imbalance, is one of the most prevalent problems with raw data. This issue arises in many domains when the number of instances with negative labels is far higher than the total number of occurrences with positive labels. For example, it is used in fraud detection, medical diagnostics, and network intrusion detection. Machine Learning (ML) algorithms fail in dealing with imbalanced data because they focus on reducing error rates for the majority category while disregarding the minority. The research aims to propose an effective method to deal with the issue of data imbalance and improve the accuracy of ML models. We use a churn prediction dataset with imbalanced data obtained from Kaggle. The dataset initially contains missing values, irrelevant features, improper data formats, and imbalances. To address these challenges, preprocessing is conducted. For data imbalance, we introduce a novel ensemble margin-based algorithm along with custom methods such as Tomek Links, Synthetic Minority Over-sampling Technique (SMOTE), and NearMiss. The balanced data from each method is then fed into ML models like Support Vector Machine (SVM) and Naïve Bayes (NB). The performance of both models under various techniques is evaluated using positive metrics. Experimental findings indicate that the proposed algorithm achieves the highest accuracy of 97.29% and 98.17% for SVM and NB models, respectively.

Keywords—*Imbalance Data, Data Pre-processing, Features, Kaggle, Churn Prediction, Machine Learning, Accuracy, Ensemble Method*

Introduction

A class imbalance occurs if and only if the number of samples in each class in a dataset is unevenly distributed. Simply put, there is a large disparity in the total number of instances between classes [1]. When there are two classes in a binary classification problem, and there are far fewer samples of the positive class than the negative class, we say that there is an imbalance. Problems with imbalanced datasets can arise from a variety of sources and types of data. When some groups are favoured over others during the selection or data-collecting process, this is known as sampling bias. Another problem that might arise from privacy concerns or restricted resources is data scarcity, which shows up when certain classes have fewer examples than others [2]. Another possible cause of imbalanced datasets is idea drift, which can be attributed to changes in the environment or user behaviour over time [3]. The results and efficiency of ML models might be negatively impacted by imbalanced data [4].

Consider ML algorithms as an example. The data is skewed in favour of the predominant class, they reason, and the minority class is either ignored or incorrectly classified. As a result, even though the algorithm is generally very accurate, the minority class may have poor precision, recall, or F1-score. Due to their inappropriateness or unfairness for unbalanced datasets, traditional metrics and assessment methods for ML models may provide an inaccurate representation of the model's actual performance or value for the minority class or a specific issue area. If the imbalance is big or unconnected to what was seen in the training data, ML models trained on such datasets could have a hard time adjusting their findings to fresh or undiscovered data. Many real-world fields of study make use of imbalanced datasets, including medical imaging, identifying genetic sequences, estimating face ages, detecting anomalies, and the life sciences [5]. The purpose of this research is to examine the prediction of client churn.

Churn prediction is a critical component of marketing analytics [6]. As information becomes more accessible and business conditions become more volatile, clients may find it easier to switch from one service provider to another. Retaining existing clients is typically less expensive than obtaining new ones. According to research, loyal customers help to enhance revenue and encourage word-of-mouth marketing. Instead of indiscriminately pursuing new clients, firms should focus on retaining their existing customer base. An accurate churn prediction model is required for optimal customer retention to detect probable churners [7]. The challenges caused by a class imbalance in customer churn prediction algorithms are difficult to overcome, and customer churn prediction is commonly presented as a simple binary classification problem. Some established industries, notably telecommunications, experience a turnover rate of less than 25%, implying far fewer churners than non-churners. Consequently, churn prediction models are frequently trained on an uneven collection of customer data, with non-churners constituting the majority and churners the minority [8]. Many traditional learning algorithms face accuracy concerns when dealing with class imbalance scenarios due to their bias toward the majority class, neglecting the minority. Using traditional learning approaches to create churn prediction classifiers on an uneven dataset yields acceptable overall accuracy by classifying the majority of occurrences, including minority churners, as non-churners. However, this categorization is meaningless as it results in poor prediction accuracy for the minority class [9]. Consequently, several recent research areas have focused on the issue of imbalanced learning.

Several solutions to the challenge of class imbalance have been proposed; nevertheless, the question of accuracy persists. To enhance ML model performance on imbalanced data, we present an innovative bagging-based ensemble model for predicting customer churn, which can be optimized for a profit-based measure. The remainder of this work is organized as follows: Section 2 provides a brief survey of recent work on class imbalance. In Section 3, we describe the technique of predicting churn using imbalanced data. Following that, Sections 4 and 5 discuss balancing approaches and ML models, respectively, while Section 6 offers a comprehensive analysis of experimental data. Section 7 summarizes the study and provides resources for further research.

Literature Survey

Corrected: Some of the recent work done by researchers to handle imbalanced data in various applications and achieve better prediction results is detailed in this section. Inthe article [10], the author compares the efficiency of Deep Neural Network (DNN) and Convolutional Neural Network (CNN)-based algorithms on several well-known solutions for imbalanced data, including oversampling and undersampling. Next, present a CNN-based model that manages unequal data well by using SMOTE. To evaluate the techniques, they used the KEEL, Z-Alizadeh Sani, and breast cancer datasets, running the trials 100 times with varied data distributions to ensure reliable results. On all 24 imbalanced datasets, the hybrid SMOTE-Normalization-CNN achieved 99.08% accuracy, surpassing previous techniques. As a result, the proposed mixed model has the potential to address various real-world datasets with imbalanced binary classification challenges. The publication [11], describes the Credit Card Anomaly Detection model, employing meta-learning ensemble approaches and the base learners paradigm to enhance anomaly detection. To identify fraudulent transactions using the proposed stacked ensemble technique, they employ the XGBoost algorithm as the meta-learner and four outlier detection methods. To address data imbalance and overfitting, they utilize a k-fold cross-validation technique and a stratified sample approach. Calculating the discordance rate enhances the precision of ensemble learning. The research trains and assesses the model on two datasets: Credit Card Default Payment and Fraud. Experiments reveal that the method identifies more outliers than state-of-the-art techniques, especially for occurrences of minority classes in these datasets.

The study [12] examines the Credit Card Fraud Recognition Dataset, notable for its high-class imbalance and real-world transaction data. This study compares two ML classification algorithms, binary and one-class, based on their performance in identifying occurrences of credit card fraud. Binary classification proves superior in detecting credit card fraud, with CatBoost identified as the best binary classifier in the research findings. The research paper [13] proposes a system for identifying fraudulent transactions with credit cards employing fraud feature-boosting algorithms and a Spiral Oversampling Balancing Technique (SOBT). The objective is to enhance the dataset by eliminating redundant or closely similar attributes using a compound grouping elimination strategy. Additionally, improve each feature's decision-making capabilities for the target domain with a multifactor synchronous embedding technique. To improve the fraud detection model's capacity to distinguish between genuine and fraudulent transactions, they propose a SOBT with an equal proportion of the two samples. The solutions outperform cutting-edge algorithms, enabling effective fraud detection based on extensive research findings from two standard datasets.

Using data from the 2009 High School Longitudinal Study, the research [14] explores various sampling procedures to cope with both somewhat unbalanced and severely imbalanced classifications. Three techniques are examined: random undersampling (RUS), random oversampling (ROS), and a hybrid approach combining SMOTE for nominal and continuous with RUS. Findings indicate that hybrid resampling is the most effective strategy for extremely unbalanced data, while random oversampling is best for moderately imbalanced data. The discussion focuses on prospective future research areas and implications for educational data mining applications. The researchers [15] propose a unique technique called Neighbor-based Under-Sampling (N-US) to address class imbalance. The purpose of this study is to demonstrate that the recommended N-US approach can reliably predict modules with flaws. N-US undersamples the dataset to minimize the removal of majority data points while highlighting minority data points in order to prevent information loss. The utility of N-US is assessed by comparing it to three frequently used under-sampling approaches, investigating its virtues and disadvantages as a dependable partner to a Software Defect Prediction (SDP) classifier through numerous tests on benchmark data. The recommended model outperforms other probable SDP models in the evaluation of the N-US technique for SDP.

Materials and Methods

In this segment, we elaborate on the methodological approaches undertaken to address the imbalanced data problem and enhance the accuracy of our ML model. This study utilizes a dataset from Kaggle for churn forecasting. The initial steps of data pre-processing involved meticulous elimination of missing data, identification, and removal of unnecessary features, and modification of certain features to optimize their impact on the prediction model. We devised a novel ensemble method specifically for handling imbalanced data. Subsequently, this innovative method was compared to common balancing techniques such as Tomek link, SMOTE, and NearMiss. After applying these methods to the pre-processed data, it was fed into the SVM and NB models for churn prediction. Various metrics were employed to assess the effectiveness of each imbalance technique. The results were thoroughly analysed to establish the optimum approach for dealing with imbalanced data. Figure 1 illustrates the overall workflow of our investigation, demonstrating the progression from pre-processing the data to addressing imbalances and applying the model.

Data Collection and Processing

The research study utilizes a telecom churn dataset from Kaggle [16] that comprises nineteen columns (independent variables) describing the features of telecom firm customers. The Churn column (dependent variable) indicates whether a client left within the last month. In this column, the class "Yes" represents churn, while "No" represents non-churn. This study's primary goal is to establish the connection between customer characteristics and churn. The collected data comprises 7043 samples, consisting of 1869 churn and 5174 non-churn samples. The distribution of the data is depicted in Figure 2, illustrating a highly imbalanced dataset.

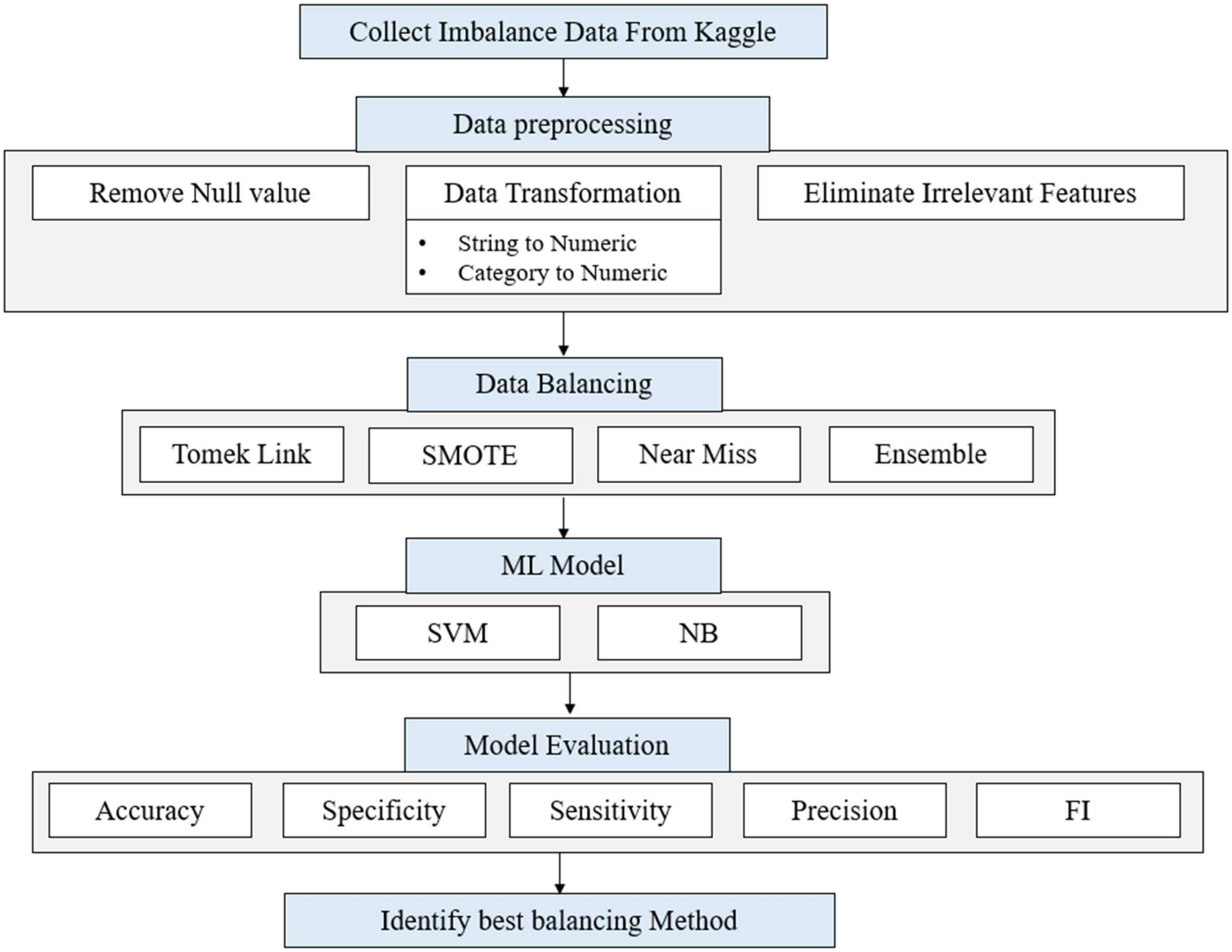

Fig. 1. Workflow of handling data imbalance to enhance ML model accuracy

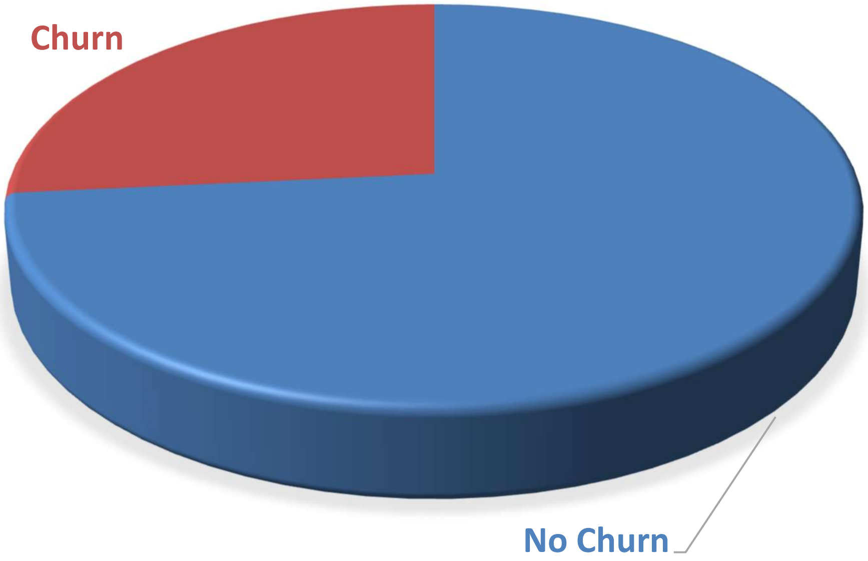

Fig. 2. Telecom churn data distribution.

The collected raw data from Kaggle is given in Table 1. The dataset is initially checked for missing values. Currently, eleven values are missing from the TotalCharges column. We can simply eliminate the samples with missing values from our whole dataset of 7043. It is important to note that the TotalCharges column was of the object type. This column is a numeric variable that displays the total amount charged to the consumer. We transform this column to a numerical data type so that we can perform additional analysis on it. The customerID field does not indicate whether a customer would churn. As a result, we have removed this column from the dataset. Many of the features are grouped into categories.

This study used a mix of label encoding [17], one-hot encoding [18], and feature calling to handle numerical variables, whereas label encoding was used to handle categorical variables with two values. To normalize numerical values for categorical variables, label encoding is utilized. To accommodate numerical values, we will use label encoding to convert two-value category variables such as Dependents, PaperlessBilling, Partner, Churn, PhoneService, and gender into new variables. Using one-hot encoding, categorical variables are transformed having more than two values into numerical ones. The variables are MultipleLines, TechSupport, OnlineSecurity, DeviceProtection, StreamingTV, Contract, InternetService, StreamingMovies, OnlineBackup, and PaymentMethod.For feature scaling, we use MinMaxScaler. We set the values of features such as MonthlyCharges, TotalCharges, and tenure from 0 to 1. After applying all the pre-processing techniques, the processed data is presented in Table 2. Now, the data contains no irrelevant features, missing values, or categorical values.

Table 1. Raw data of telecom churn prediction from Kaggle.

Customer ID	Gender	Senior Citizen	Partner	Dependents	**….**	Monthly Charges	Total Charges	Churn
7892-POOKP	Female	0	No	Yes	….	104.8	3046.05	Yes
4190-MFLUW	Female	0	No	No	….	55.2	528.35	No
1066-JKSGK	Male	0	Yes	No	….	20.15	20.15	No
7310-EGVHZ	Male	0	No	Yes	….	20.2	20.2	No
4080-IIARD	Female	0	No	No	….	76.2	981.45	No

Table 2. Data after several pre-processing techniques. Gender	Senior Citizen	Partner	Dependents	Payment Method	….	Monthly Charges	Total Charges	Churn
1	0	0	1	0	….	0.19	0.523	1
1	0	0	0	1	….	0.1	0.09	0
0	0	1	0	0	….	0.03	0.003	0
0	0	0	1	0	….	0.03	0.003	0
1	0	0	0	2	….	0.14	0.169	0

Data Balancing Techniques

To address imbalanced data, we introduce a novel ensemble method, which is then compared to traditional balancing strategies like Tomek Links, SMOTE, and NearMiss. This section elaborates on the theoretical concepts underlying all of these strategies.

Tomek Links

Tomek introduced the Tomek link, initially designed for two classes: one as the majority and the other as the minority, assuming no other class x_z exists such that $d(x_a, x_z) < d(x_a, x_b)$ or $d(x_b, x_z) < d(x_a, x_b)$ [19]. In this situation, the distance between the two classes is $d(x_a, x_b)$, where x_a and x_b represent the majority and minority, respectively. T-link selects instances from the majority class using the closest neighbor criteria and eliminates those that are too close to the minority class. T-link is another name for the nearest neighbor that is better-condensed. Since there are no clearly defined borders, this method can also be used for post-processing data cleaning to eliminate instances from both the majority and minority classes. Only by eliminating examples from the majority class can this strategy be deemed under-sampling. The results were significantly improved following the T-link treatment.

SMOTE

Over-sampling enhances the quantity of minority group samples by randomly reproducing them to the required extent to represent a balanced distribution. Compared to under-sampling, this method performs better without eliminating data. However, this method may also result in overfitting due to recurrent occurrences. The research [20] developed an upgraded oversampling approach called SMOTE. SMOTE uses the k-nearest neighbor technique to generate fresh synthetic samples in the feature space based on a predefined percentage of minority classes. SMOTE can use the current minority class data without duplicating it to create fresh synthetic data in order to solve the overfitting issue. Equation (1) provides a foundation for the synthetically created data:

$$S_{syn} = r\left(S_{kNN} - S_f\right) + S_f \qquad [1]$$

here r = a random number between 0 and 1, S_{syn} and S_f are generated synthetic and feature samples, respectively, and S_{kNN} represents feature samples considered as k-nearest neighbors. The classifier utilizes synthetic samples to construct targeted zones.

NearMiss

Since Near Miss creates a more stable and equal class distribution boundary, it improves classifier performance in large-scale imbalanced datasets, which is why we selected it. The experiment utilized the Near Miss technique to implement undersampling, invoking a class through an imbalance-learn package [21]. To achieve the required results, we modified the Near Miss technique by providing it with the appropriate parameters. Three versions of the Near Miss technique are available:

- NearMiss-1 determines the test data that is the furthest away by averaging the nearest samples from the negative class.
- NearMiss-2 picks positive samples that are geographically close to the farthest negative samples in the class.
- The NearMiss-3 procedure consists of two phases. The first order of business is to keep each negative sample's nearest neighbors. Following that, positive samples are chosen by averaging the distances between each person and their immediate neighbors.

NearMiss-1 is susceptible to noise when undersampling a certain class. This means that neighborhood samples will be drawn from the target class. Nonetheless, samples near the limits will often be chosen. NearMiss-2 will not have this effect because it prefers the farthest samples over the closest ones. When there are minor outliers, sampling can potentially change the noise level. The first-step sample selection improves NearMiss-3's noise resistance. After multiple iterations of all three variants, the NearMiss-2 variation was chosen for this study since it was demonstrated to be the best fit for the credit card dataset. To ensure fair cross-comparison, datasets were treated in a conventional experiment.

Ensemble

The proposed ensemble margin-based imbalance training technique has been motivated by the significant oversampling approach SMOTEBagging, which was described in the preceding section. It integrates the concepts of under-sampling, ensemble, and margins. Our strategy prioritizes low-margin cases and has the potential to overcome the faults in both SMOTEBagging [22] and UnderBagging [23]. When compared to SMOTEBagging and UnderBagging, this technique prioritizes the most relevant examples for classification tasks while utilizing fewer computational resources. The complete process consists of three phases:

- Using the training data, an ensemble classifier determines the ensemble margin values.
- By concentrating more on cases with a narrow margin, balanced training subsets are created.
- Using balanced training subsets to train base classifiers and creating an ensemble with improved imbalance learning capabilities.

The training samples are represented as $S = \{X, Y\} = \{x_i, y_i\}_{i=1}^{n}$. The first step in our technique is to create a robust ensemble classifier (bagging) from the complete training set. Then, for each training instance, we determine its margin. Phase two entails establishing fresh balanced training subsets from the most important training data for classification. Let L denote the total number of classes, and N_i represent the number of training samples in each class. We arrange the classes in decreasing order based on the total number of instances. This indicates that class 1 has the highest training size, N_1, while class L has the smallest, N_L. The margin-based importance assessment function is used to list the training instances for each class in descending order., with $1 \leq c \leq L$. If an instance $x_i \in c$ has a greater significance value $W(x_i)$ for each class c, it is deemed more important for the classification choice. Then, similar to SMOTEBagging, a rate of resampling is used to control how many instances are chosen for each class in order to produce a balanced dataset. Every instance belonging to the smallest class is kept.

The span of a was first defined as 10–100, creating $N_L$instances by bootstrapping from N_1. Each class $c \neq L$ is utilized to produce the subset S_{c1}using the $a\%$ of its significance-ordered samples of class c, with Lbeing the smallest class. There isn't a single outlier in any of the categories when N_1 is smaller than the number of class $c(2 \leq c \leq L - 1)$. Asin UnderBagging, the initial $N_c$samples of class c are used to bootstrap the $a\%, N_L$instances. The initial set of balanced data is created by combining the N_L minimal class samples with $S_{c1}(c = 1, \ldots., L - 1)$. Building the first base classifier is the next stage after acquiring the balanced training set. In mathematics, Amight represent a potential progression involving elements in the interval a. The resampling rate ranging from 10% to 100% will be used by each of the ten classifiers comprising an ensemble of $T = 100$, much like SMOTEBagging. In contrast to SMOTEBagging, which uses N_1, specifically, our technique uses the minority category L's training dimensions as an example of importance-based undersampling on other majority categories, and uses the majority category 1's training dimension as a baseline for oversampling (SMOTE) on other relative minority categories.

ML Models

After data processing and balancing for churn prediction, we employed two ML models: SVM and NB. The details of these two ML models are provided in this section.

SVM

SVM is a common classification model that seeks a suitable hyperplane to divide the acquired data samples. The classification process is based on the maximization of hard and soft intervals, each formulated as a unique programming quadratic equation. The primary models are outlined below [24]. If the training set is linearly separable, then the optimization of the hard interval should be carried out using the linear separable SVM. To obtain the optimal kernel function and optimize the soft interval in a training sample that is almost linearly separable, a linear SVM should be employed. A nonlinear SVM must be utilized in order to maximize the soft interval and identify the proper kernel function of the training sample in a non-time-sharing setting. Here is a summary of the most important SVMs.

To begin, we supply a set of training samples. The main principle of a linear separable SVM is to locate a suitable partitioning hyperplane in the sample space, and hyperplanes are employed to divide the samples into different categories. The term "linearly separable" refers to data samples that may be separated using a linear function. More precisely, a linear function can be defined as follows: in two dimensions, a straight line is commonly conceived of as a linear function, while in three dimensions, a plane is commonly referred to as a linear function. A hyperplane is a linear function that ignores spatial dimensions. Although samples can be separated linearly, examining the data graphically indicates that there are an endless number of lines that can be used to separate samples. Lines with the longest intervals and the ability to accurately separate data correspond to the linearly separable SVM. Determining the interval within the sample space is critical since the maximum interval is desired. To describe the division of the hyperplane in the sample space, we employ the linear equation below:

$$W^T x + b = 0 \qquad [3]$$

W is the normal vector that defines the hyperplane's orientation, and b is the displacement that determines the separation of the hyperplane from the origin. We make the following assumption on the training samples: the hyperplane can correctly identify them if and only if the following formula applies:

$$w^t x_i + b \geq 1, y = 1 \qquad [4]$$

$$w^t + b \leq -1, y = -1 \qquad [5]$$

The notion of the maximum interval hypothesis refers to the formula presented above. The values of 1 or −1 are provided for ease of calculation, although any constant can be used.

NB

The naïve Bayes classifier, a Bayesian theory-based probability categorization method, can be applied to challenges with classification requiring multiple classes [25]. The strength of this classifier lies in its ability to estimate data classes using only a restricted amount of features. Data mining and ML systems typically use this classifier when presented with a classification difficulty. Here's the classification model.

$$P\left(\frac{X}{Y}\right) = P\left(\frac{Y}{X}\right) * \frac{P(X)}{P(Y)} \qquad [6]$$

The $P(X)$ signifies the conditional probability of X, while $P(X|Y)$ denotes X's prior or marginal probability. $P(Y|X)$ represents the conditional probability of Y given X. $P(Y)$is the prior or marginal probability of Y, which serves as a normalization constant. There exists a linear and scalable probabilistic classifier. The multiclass classification challenge arises when an instance can be classified into more than three categories. This classifier, used with "N" classes in ML, can predict the class label of the test. To categorize things, a probabilistic method known as a naive Bayes classifier uses Bayes' theorem. This classifier considers the independence of data point attributes, whether naive or strong. Many applications use naive Bayes classifiers, including medical diagnosis, text analysis, signal segmentation, and spam filters. Its ease of implementation makes it a popular choice among ML techniques. A prediction model is described as below:

$$p(C_k|x_1 \ldots .x_n) = p(C_k)\prod_{i=1}^{n} p(x_i|C_k) \qquad [7]$$

In a dataset categorized by conditional probability, x_i represents a sample, and C_k represents the class.

Results and Discussion

The purpose of this research is to propose an effective method for handling imbalanced data and to validate this approach, we selected highly imbalanced telecom churn prediction data from Kaggle. The data underwent preprocessing, followed by balancing using four different techniques: Tomek Links, SMOTE, NearMiss, and the proposed Ensemble method. The balanced datasets from each technique were then employed to train ML models, namely SVM and NB, for churn prediction. The outcome of churn prediction by SVM and NB model using various balancing techniques is given in Table 3.

Table 3. Performance evaluation of ML models using balanced data from different techniques.Model	**SVM**				
Metrics	ACCURACY	SPECIFICITY	SENSITIVITY	PRECISION	F1
Tomek Links	94.19	87.71	96.48	95.69	96.09
SMOTE	96.55	93.62	97.49	97.95	97.72
NearMisss	95.56	89.77	97.66	96.33	96.99
Ensemble	97.29	94.10	98.35	98.04	98.19
Model	NB				
Metrics	ACCURACY	SPECIFICITY	SENSITIVITY	PRECISION	F1
Tomek Links	94.99	88.75	97.19	96.07	96.63
SMOTE	97.81	94.98	98.77	98.32	98.54
NearMisss	96.14	90.97	98.01	96.77	97.39
Ensemble	98.17	95.76	98.98	98.58	98.78

The outcomes of the SVM model on different balanced datasets were assessed using positive metrics like Accuracy, Specificity, Sensitivity, Precision, and F1 score. The proposed Ensemble method achieved the highest metrics, with an accuracy of 97.29%, specificity of 94.1%, sensitivity of 98.35%, precision of 98.04%, as well as F1 score of 98.19%. The second-highest performance was noted with the traditional method SMOTE, obtaining 96.55% accuracy, 93.62% specificity, 97.49% sensitivity, 97.95% precision, and 97.72% F1 score. The lowest performance was recorded with the Tomek Link method. Figure 3 illustrates the comparison of performance metrics achieved by the SVM model on various balancing techniques.

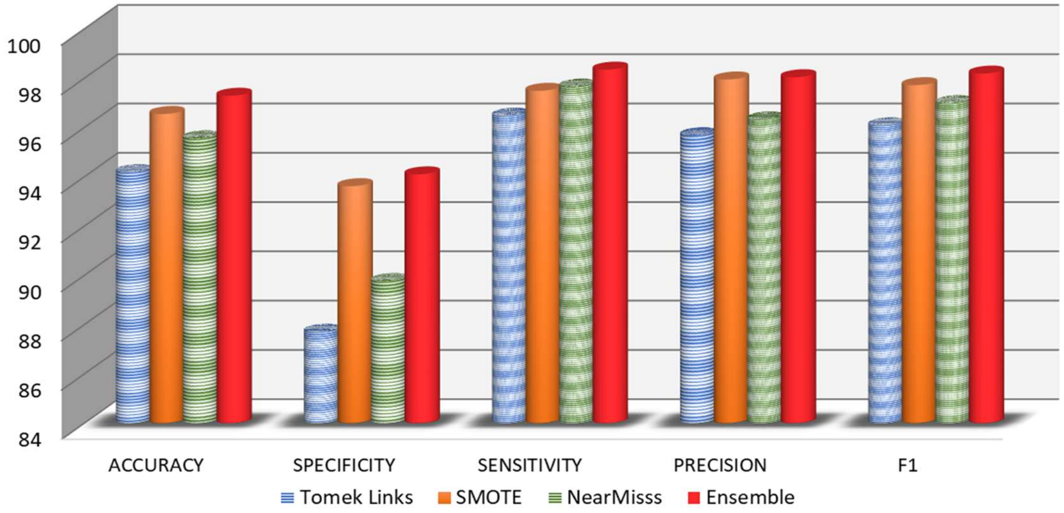

Fig. 3. Outcome of SVM model using data from various data balancing techniques.

Next, the outcomes of the NB model on different balanced datasets were evaluated using the same metrics. Similar to the SVM results, the Ensemble method demonstrated the highest metrics, with an accuracy of 98.17%, specificity of 95.76%, sensitivity of 98.98%, precision of 98.58%, and F1 score of 98.78%. The lowest metrics were attained by the Tomek method, with an accuracy of 94.99%, specificity of 88.75%, sensitivity of 97.19%, precision of 96.07%, and F1 score of 96.63%. Figure 4 provides a visual comparison of the performance metrics achieved by the NB model on various balancing techniques. The results highlight the efficiency of the suggested Ensemble method in handling imbalanced information compared to traditional techniques.

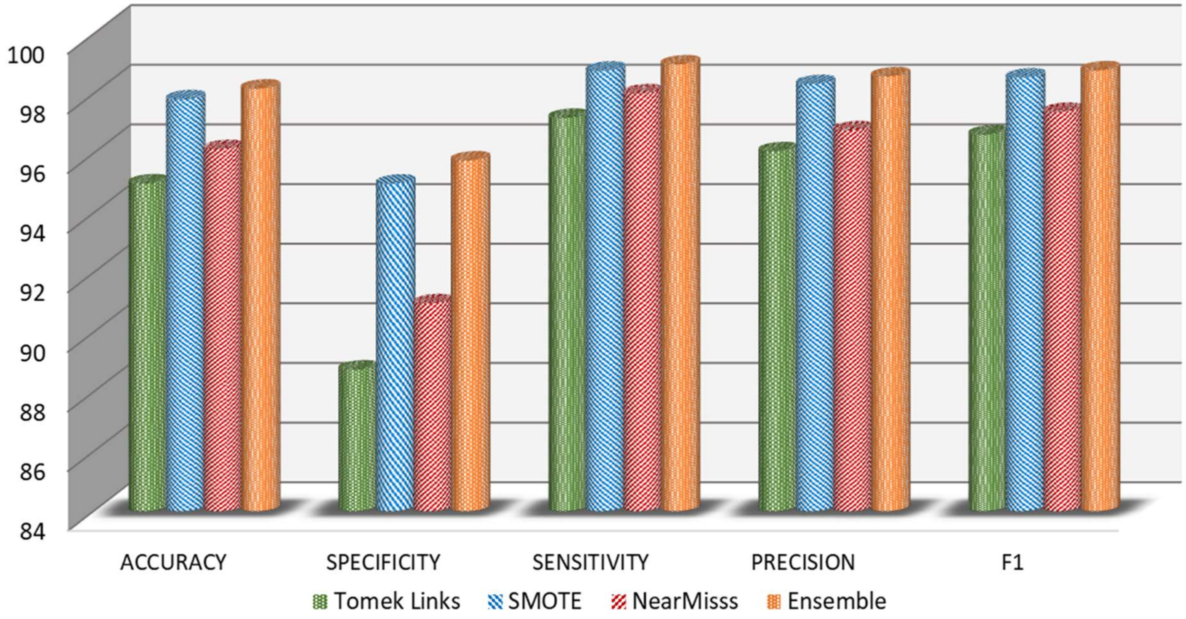

Fig. 4. Outcome of NB model using data from various data balancing techniques.

Conclusion

Due to the skewed nature of data, imbalanced data issues in data mining are prevalent these days and have a detrimental effect on the classification step in the machine learning workflow. This research successfully proposed an effective ensemble algorithm for handling imbalanced data, aiming to enhance the accuracy of ML model predictions. To evaluate the performance of our research, we chose the churn prediction dataset from Kaggle, which serves as an exemplary case of imbalanced data. The acquired data underwent analysis, processing, and imbalanced data handling using Tomek Links, SMOTE, NearMiss, and ensemble methods. The outcomes of these methods were then provided to ML models for churn prediction. As discussed earlier, imbalanced data significantly affects ML models, leading to increased errors in predictions. However, our proposed model successfully mitigates these errors and improves the overall accuracy of the ML model. Specifically, the proposed balancing method achieved 98.17% and 97.29% accuracy for churn prediction using NB and SVM models, respectively.In comparison, traditional methods such as Tomek Links, SMOTE, and NearMiss achieved accuracies of 94.99%, 97.81%, and 96.14% with the NB model. The SVM classifier, when using data processed by Tomek Links, SMOTE, and NearMiss methods, produced accuracies of 94.19%, 96.55%, and 95.56%, respectively. In the future, we aim to conduct a comprehensive comparative study on various types of unbalanced datasets in order to assess the effectiveness of the proposed method. In addition, we will keep improving and optimising the suggested ensemble approach while taking into account other factors like scalability and processing efficiency.

References

[1] Ali, Aida, Siti Mariyam Shamsuddin, and Anca L. Ralescu. "Classification with class imbalance problem." *Int. J. Advance Soft Compu. Appl* 5, no. 3 (2013): 176-204.

[2] Ali, Haseeb, MN Mohd Salleh, Rohmat Saedudin, Kashif Hussain, and Muhammad Faheem Mushtaq. "Imbalance class problems in data mining: A review." *Indonesian Journal of Electrical Engineering and Computer Science* 14, no. 3 (2019): 1560-1571.

[3] Priya, S., and R. Annie Uthra. "Comprehensive analysis for class imbalance data with concept drift using ensemble based classification." *Journal of Ambient Intelligence and Humanized Computing* 12 (2021): 4943-4956.

[4] Tran, Ngan, Haihua Chen, Janet Jiang, Jay Bhuyan, and Junhua Ding. "Effect of Class Imbalance on the Performance of Machine Learning-based Network Intrusion Detection." *International Journal of Performability Engineering* 17, no. 9 (2021).

[5] Shyalika, Chathurangi, Ruwan Wickramarachchi, and Amit Sheth. "A Comprehensive Survey on Rare Event Prediction." *arXiv preprint arXiv:2309.11356* (2023).

[6] Li, Weilong, and Chujin Zhou. "Customer churn prediction in telecom using big data analytics." In *IOP Conference Series: Materials Science and Engineering*, vol. 768, no. 5, p. 052070. IOP Publishing, 2020.

[7] Lemmens, Aurélie, and Sunil Gupta. "Managing churn to maximize profits." *Marketing Science* 39, no. 5 (2020): 956-973.

[8] Tamuka, Nyashadzashe, and Khulumani Sibanda. "Real time customer churn scoring model for the telecommunications industry." In *2020 2nd International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, pp. 1-9. IEEE, 2020.

[9] Kim, Jongchan, and Marco Comuzzi. "A diagnostic framework for imbalanced classification in business process predictive monitoring." *Expert Systems with Applications* 184 (2021): 115536.

[10] Joloudari, Javad Hassannataj, Abdolreza Marefat, Mohammad Ali Nematollahi, Solomon Sunday Oyelere, and Sadiq Hussain. "Effective Class-Imbalance Learning Based on SMOTE and Convolutional Neural Networks." *Applied Sciences* 13, no. 6 (2023): 4006.

[11] Islam, Md Amirul, Md Ashraf Uddin, Sunil Aryal, and Giovanni Stea. "An ensemble learning approach for anomaly detection in credit card data with imbalanced and overlapped classes." *Journal of Information Security and Applications* 78 (2023): 103618.

ISSN:2731-538X | E-ISSN:2731-5398
Vol. 19 No. 01 (2025)

[12] Leevy, Joffrey L., John Hancock, and Taghi M. Khoshgoftaar. "Comparative analysis of binary and one-class classification techniques for credit card fraud data." *Journal of Big Data* 10, no. 1 (2023): 118.

[13] Ni, Lina, Jufeng Li, Huixin Xu, Xiangbo Wang, and Jinquan Zhang. "Fraud feature boosting mechanism and spiral oversampling balancing technique for credit card fraud detection." *IEEE Transactions on Computational Social Systems* (2023).

[14] Wongvorachan, Tarid, Surina He, and Okan Bulut. "A comparison of undersampling, oversampling, and SMOTE methods for dealing with imbalanced classification in educational data mining." *Information* 14, no. 1 (2023): 54.

[15] Goyal, Somya. "Handling class-imbalance with KNN (neighbourhood) under-sampling for software defect prediction." *Artificial Intelligence Review* 55, no. 3 (2022): 2023-2064.

[16] https://www.kaggle.com/datasets/blastchar/telco-customer-churn

[17] Wei, Jinxin, and Zhe Hou. "Binary Encoding for Label." *Authorea Preprints* (2023).

[18] Rodríguez, Pau, Miguel A. Bautista, Jordi Gonzalez, and Sergio Escalera. "Beyond one-hot encoding: Lower dimensional target embedding." *Image and Vision Computing* 75 (2018): 21-31.

[19] Pereira, Rodolfo M., Yandre MG Costa, and Carlos N. Silla Jr. "MLTL: A multi-label approach for the Tomek Link undersampling algorithm." *Neurocomputing* 383 (2020): 95-105.

[20] Chawla, Nitesh V., Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. "SMOTE: synthetic minority over-sampling technique." *Journal of artificial intelligence research* 16 (2002): 321-357.

[21] Tanimoto, Akira, So Yamada, Takashi Takenouchi, Masashi Sugiyama, and Hisashi Kashima. "Improving imbalanced classification using near-miss instances." *Expert Systems with Applications* 201 (2022): 117130.

[22] Wang, Shuo, and Xin Yao. "Diversity analysis on imbalanced data sets by using ensemble models." In *2009 IEEE symposium on computational intelligence and data mining*, pp. 324-331. IEEE, 2009.

[23] Barandela, Ricardo, Rosa Maria Valdovinos, and José Salvador Sánchez. "New applications of ensembles of classifiers." *Pattern Analysis & Applications* 6 (2003): 245-256.

[24] Pisner, Derek A., and David M. Schnyer. "Support vector machine." In *Machine learning*, pp. 101-121. Academic Press, 2020.

[25] Berrar, Daniel. "Bayes' theorem and naive Bayes classifier." *Encyclopedia of bioinformatics and computational biology: ABC of bioinformatics* 403 (2018): 412.