Results in Engineering 26 (2025) 104629

Contents lists available at ScienceDirect

Results in Engineering

journal homepage: www.sciencedirect.com/journal/results-in-engineering

Research paper

A data-driven approach with explainable artificial intelligence for customer churn prediction in the telecommunications industry

Daniyal Asif [a], Muhammad Shoaib Arif [b,*], Aiman Mukheimer [b]

[a] Skolkovo Institute of Science and Technology (Skoltech), Moscow, 121205, Russia
[b] Department of Mathematics and Sciences, College of Humanities and Sciences, Prince Sultan University, Riyadh, 11586, Saudi Arabia

ARTICLE INFO

Keywords:
Churn prediction
Telecom industry
Retention strategies
Decision-making
Business intelligence
Data science
Machine learning
Predictive analytics
Explainable artificial intelligence

ABSTRACT

In the competitive telecommunications industry (TCI), retaining clients is crucial for profitability, as customer churn remains a significant challenge. Traditional machine learning (ML) models often lack the predictive power needed for complex telecom data, while black-box models provide limited transparency, reducing trust and actionable insights. This study introduces XAI-Churn TriBoost, an interpretable and explainable data-driven model developed using a dataset of over 2 million records. The model combines extreme gradient boosting (XGBoost), categorical boosting (CatBoost), and light gradient boosting machine (LightGBM) in a soft voting ensemble to enhance churn prediction. Data preprocessing included handling missing values through iterative imputation with a Bayesian ridge. Sequential data scaling was implemented by combining robust, standard, and min-max scaling methods to ensure feature consistency. Feature selection was conducted using the Boruta technique with a random forest (RF), and class imbalance in the training data was addressed using the synthetic minority oversampling technique (SMOTE). XAI-Churn TriBoost achieved high predictive performance, with an accuracy of 96.44%, precision of 92.82%, recall of 87.82%, and F1 score of 90.25%. To enhance model transparency, we incorporated explainable artificial intelligence (AI) techniques, specifically local interpretable model-agnostic explanations (LIME) and Shapley additive explanations (SHAP), to interpret individual predictions and identify critical features affecting churn. Key factors impacting churn include regularity and montant, offering TCI valuable insights for targeted retention strategies. XAI-Churn TriBoost thus provides both robust performance and interpretability, highlighting its potential to support customer retention efforts in the TCI.

1. Introduction

Modern connectivity relies heavily on the TCI, which drives innovation, global business, and communication. As markets become saturated and competition intensifies, customer retention has become as crucial as customer acquisition. Price wars, evolving consumer expectations, and frequent technological disruptions are among the challenges businesses face in today's highly competitive environment. For telecommunication carriers, customer retention is a strategic priority, especially as maintaining high-quality service becomes increasingly difficult amidst rising operational costs and growing regulatory constraints. Competitiveness, the number of operators, service offerings, and other metrics have all contributed to the TCI remarkable growth and development over the past several decades. However, intense competition, saturated markets, and a constantly evolving environment have also led to significant challenges, particularly in managing customer churn [60]. The saturated market landscape allows customers to easily switch between services and providers, and the wide variety of highly competitive service offerings further exacerbates the issue. From the perspective of service providers, churned customers either discontinue using their services or switch to competitors, resulting in a substantial impact on the company's financial performance [56].

Customer churn, where customers stop using a service, directly impacts profitability by reducing brand loyalty, increasing customer acquisition costs, and eroding revenue. High churn rates also disrupt resource allocation and network planning, compromising service continuity. Research shows that it might cost five to twenty-five times more to acquire a new customer than to maintain an existing one, hence reducing churn is economically vital [10]. Globalization and technical improvements put further strain on TCI since they give customers more alternatives [15]. Revenue and profitability have been hit even harder by rising competition, pricing regulation, market consolidation, and the popularity of

* Corresponding author.
E-mail addresses: Daniyal.Asif@skoltech.ru (D. Asif), marif@psu.edu.sa (M.S. Arif).

https://doi.org/10.1016/j.rineng.2025.104629
Received 11 January 2025; Received in revised form 7 March 2025; Accepted 12 March 2025

Available online 17 March 2025

over-the-top services [45,51]. Recognizing critical customer behaviors, improving customer engagement, and maintaining competitive advantage all depend on precise churn prediction in this setting.

Customer churn is a big concern for TCI because keeping current customers is much cheaper than finding new ones [52]. Keeping consumers around for the long run is crucial for maintaining a constant flow of revenue and remaining ahead of the competition. The most common reason for clients to abandon a firm is when they are unhappy with the services they receive for a long time. Most of the time, people decide to cease using a service after becoming increasingly unhappy with it over time. Service providers can address this by proactively measuring client satisfaction and identifying areas for service improvement. By separating loyal customers from those at risk, telecom companies can employ predictive algorithms to develop tailored retention tactics. Predicting customer turnover is an important emphasis for TCI as it has a direct impact on brand value and revenue output. It is common practice for telecoms to meticulously document client data and service usage in order to identify patterns and refine retention strategies. Maintaining TCI's growth depends on accurately predicting customer churn, since client retention is of utmost importance [32].

In various domains, ML has proven its adaptability [25,2]. ML has emerged as a powerful tool in TCI for predicting consumer behavior and identifying patterns in large datasets, therefore lowering customer attrition. Customer risk assessment and churn prediction have both seen extensive use of ensemble techniques and other sophisticated ML models. The models allow TCI to conduct targeted retention initiatives, allocate resources efficiently, and fine-tune their marketing strategies. By shifting from reactive to proactive operations, ML alters the manner in which firms engage with customers and aids in client retention [29,22]. Even though conventional ML models have promise, their predictions are often inaccurate and difficult to understand [40]. Stakeholders struggle to trust complex ensemble models and other opaque algorithms, despite the fact that they generate accurate forecasts [50]. In industries like TCI, where practical findings form the basis of strategic decisions, the absence of transparency can hinder the adoption of ML solutions [42]. To compensate for this shortcoming, explainable AI offers a way to understand model output, understand the importance of various attributes, and generate accurate predictions. Use explainable AI, which balances accuracy with interpretability, to make customer churn management decisions that are better and more transparent [8,61].

In order to combat client turnover in the telecom sector, this study adheres to essential concepts. We anticipate that a strong preprocessing pipeline will provide consistent and high-quality data, which will set the stage for precise modeling. The XAI-Churn TriBoost model is expected to outperform state of the art models in terms of predictive performance. It combines XGBoost, LightGBM, and CatBoost in a voting ensemble. We hope to improve clarity and understanding by using explainable AI methods like LIME and SHAP to shed light on the causes of customer attrition and offer solutions. Trained and tested on a large dataset of over 2 million records—with 60% allocated for training and 40% for testing—the model is designed to ensure robustness and scalability. Through the combination of advanced modeling and explainability, this study seeks to deliver a data-driven solution that supports TCI in identifying at-risk customers and implementing targeted retention strategies, ensuring both high accuracy and actionable insights for strategic decision-making.

The rest of the paper is structured as follows. Section 2 reviews related work, highlights gaps in previous research and explains the importance of addressing these issues. Section 3 describes the methodology, including data processing, ML techniques, and the use of explainable AI. Section 4 presents experimental results, evaluates model performance and discusses key findings. Section 5 provides a detailed discussion of the work, compares different models, analyzes previous studies, and outlines directions for future research. Finally, Section 6 concludes with a summary of contributions and suggestions for future work.

2. Related work

Various studies have explored customer churn prediction in the TCI, employing diverse ML techniques and methodologies. Ullah et al. [54] presented a customer attrition model for data analytics, which was validated using common assessment measures. Their proposed model demonstrated improved performance. RF achieved an 88% accuracy. The authors also performed cluster profiling to identify the major variables influencing churn based on churn risk. They concluded by offering actionable advice for telecom decision-makers on retaining their existing customers.

Ammar et al. [1] introduced a hybridized algorithm approach to effectively forecast churners. Although the accuracy of the suggested hybrid firefly algorithm and the traditional firefly algorithm were similar, the hybrid algorithm performed far better when dealing with very brief delays in time. F1 score, accuracy, precision, and recall were some of the metrics used to assess the algorithms by the writers. They emphasized that forecast models for the telecom industry must strike a compromise between precision and efficiency [19]. Creating smaller, more dimensional training models can be a straightforward way to improve performance. Methods for feature extraction and selection were especially effective, enabling accurate predictions with less computing load. Data mining has demonstrated its worth in churn prediction within the TCI in several studies [28]. A lot of research has looked into the effectiveness of classifiers such as k-nearest neighbors (KNNs), decision trees (DTs), and RF [46].

Many models that use ML approaches to forecast client churn have been developed [21,12]. Using these algorithms, the TCI can figure out when customers are about to depart and give them better service to entice them to stay. The implementation of DT- and RF-based strategies has increased the profitability of telecom services. Instead of simply removing features or disregarding data with missing values, predictive mean matching has been used to handle missing data [57]. To deal with huge datasets, shifting feature labels, and uneven data distributions, the TCI has included ensemble classifiers into consumer speculation models. According to Saran Kumar et al. [36], who conducted a comprehensive analysis of methods for predicting customer attrition, many existing models fail to accurately anticipate future churn. In order to maintain client loyalty, they emphasized the importance of precise prediction models.

An improved technique for constructing data mining applications was proposed by [34], providing the foundation for their subsequent development. Their approach was validated through the application of data mining technologies to predict prepaid subscriber retention. These studies underscore the importance of predictive modeling, feature engineering, and performance evaluation for effective customer churn analysis in the TCI. Examples of customers who consciously decided to analyze the traits and behaviors that lead to customer churn and profits were provided by [27]. The results demonstrated that enhanced models significantly outperformed their plain, non-optimized counterparts. Among the models evaluated, support vector machine (SVM) with polynomial kernel emerged as the best performer, achieving an F1 score of over 84% and an accuracy of more than 97%. Alzubaidi et al. [4] employed call detail record features to measure the strength of user-to-user social relationships within identified communities. Using a churn propagation model on the call graph, they assessed the net aggregated churn influence from churner nodes. Logistic regression (LR) was utilized to incorporate the churn influence effect and predict the churn tendency of specific users. This study highlighted that examining user connections enhances churn prediction models by leveraging social relationship attributes, such as call and SMS details.

Khalid et al. [33] compared various prediction models, and found that DTs outperformed other methods. Their findings confirmed that DTs are among the top methods for churn prediction. Similarly, [58] aimed to predict high-value customer turnover in the TCI using LR and big data analytics. By analyzing historical customer data, the study identified probable churners in the customer library and supported targeted win-back efforts based on churn behavior characteristics. Lalwani et al. [38] compared several ML methods for churn prediction, including LR, naïve bayes (NB), SVM, DTs, RF, XGBoost, CatBoost, adaptive boosting, and extra tree classifier. Their experimental results showed that ensemble algorithms, such as adaptive boosting and XGBoost, were the most accurate models, achieving an area under the curve of 84% and excelling across metrics like accuracy, precision, recall, and F1 score.

After reviewing these studies, several limitations become evident. Many rely on small datasets, limiting the generalizability and robustness of their findings. The ML techniques used are often outdated or lack sophistication, with minimal focus on advanced ensemble methods or hybrid approaches. Data preprocessing is another critical shortfall, as most studies rely on simple imputation methods for handling missing values instead of employing advanced feature engineering or selection techniques to improve model performance. Furthermore, class imbalance problems are common, which frequently results in biassed predictions, and testing datasets that are too small further reduce these models' reliability when applied to new data. The incorporation of explainable AI approaches, which are crucial for comprehending and making sense of model predictions, is also severely lacking. Strategic decision-making in the telecom sector relies heavily on actionable insights provided by explainable AI. Research into churn prediction needs better models that are more resilient, scalable, and interpretable due to these limitations. To address these gaps, the following innovative steps were taken:

1. A robust preprocessing pipeline was implemented, incorporating iterative imputation using Bayesian Ridge to effectively handle missing values.
2. The approach combines robust scaling, standard scaling, and min-max scaling sequentially to scale the data while preserving its distributional integrity.
3. To enhance model accuracy and interpretability, Boruta feature selection with RF was used to identify important features.
4. A voting classifier that combines XGBoost, LightGBM, and CatBoost was suggested as a way to improve the XAI-Churn TriBoost model's prediction accuracy and reliability.
5. Using 60% for training and 40% for testing, the model was trained and evaluated on a massive dataset with over 2 million records, proving its effectiveness and robustness over a big dataset.
6. In order to interpret the model's forecasts and offer practical understanding of the variables impacting churn, explainability methods like LIME and SHAP were employed.

3. Methodology

As depicted in Fig. 1, our suggested XAI-Churn TriBoost model for forecasting customer churn adheres to a systematic process. The churn dataset is first collected. The data is then cleaned, encoded, and scaled as part of the preprocessing steps that use Bayesian ridge for iterative imputation. After that, we employ feature engineering to generate new features and select important using the Boruta technique with RF. The data is split into training and testing sets, and we balance the training set using SMOTE. We train our model using a soft voting ensemble of three boosting algorithms: XGBoost, CatBoost, and LightGBM. For interpretability, we apply explainable AI techniques, including LIME and SHAP, to understand the predictions and provide actionable insights.

3.1. Dataset

In this study, we utilized a dataset provided through the Expresso Churn Prediction Challenge on Zindi [59]. The train.csv dataset is used in this study and is accessible to participants who join the competition on the Zindi platform. The dataset includes both numerical and categorical features, consisting of 17 variables and 2,154,048 samples. These variables capture various details about customer actions, purchases, and interactions within the network. The target variable, "churn," indicates whether a customer becomes inactive and makes no transactions for 90 consecutive days. We use 60% of the data to train the model and 40% to test our proposed model.

3.2. Data preprocessing

Data cleaning, encoding, missing value imputation, and scaling are all part of the dataset preprocessing described in this section.

3.2.1. Data cleaning

Data cleaning is necessary since the dataset includes a large number of missing values (as illustrated in Fig. 2) and features that are not useful. We removed the following features: zone1 and zone2, which represent calls to specific zones within the telecom network, as they had over 90% missing values, making them unreliable for analysis. The moving to rivaval group (MRG) feature, indicating a client who is leaving, was also dropped because it had only one unique value. The user_id, an anonymized identifier, was removed since it does not contribute to predictive analysis. The region feature, which identifies the client's location, was excluded to make the model applicable across different countries and to avoid introducing location-based bias. top_pack, indicating the most popular package used by the customer, and freq_top_pack, recording the number of activations for the top package, were removed due to high cardinality—containing 140 categorical values—and because package types vary widely among telecom companies, limiting their general applicability. After dropping these features, we retained 11 features that are common across telecom companies for further analysis.

3.2.2. Data encoding

Data encoding is a critical step in ML, as models generally work with numerical data. Categorical features, which consist of non-numeric values, need to be transformed into a format the model can interpret. For this, we use an ordinal encoder to handle our categorical features. The ordinal encoder converts categorical values into integer labels. The model can easily process each unique category in a feature because each one is assigned a particular number. When the categories are inherently ranked, this strategy really shines. In order for the model to properly learn from categorical data, the ordinal encoder converts the values into numeric form.

3.2.3. Data imputation

Because missing values can significantly affect the quality of analysis, proper approaches are required for handling missing data. When a significant amount of data is missing, it might be quite impractical to just delete the records that have that missing information. In some instances, erasing data could result in the substantial loss of important information [20]. With little missing data, traditional methods like filling in the gaps with the mean, median, or mode are effective [18]. Nevertheless, we encounter significant missing data in this instance, since certain characteristics have missing values above 50%. In order to tackle this, iterative imputation is employed.

In statistics, iterative imputation is a technique that estimates missing values by repeatedly modeling each feature with missing data as a function of other features. By repeatedly improving the imputed values and taking advantage of interdependence, this method improves precision [39]. We mainly use a Bayesian ridge model as an estimator throughout our repeated imputation process. The coefficients of this linear regression model include prior distributions, which allows for regularization. More consistent estimations are the result, and overfitting is better controlled. By estimating uncertainty features, imputation is robust against data noise and unpredictability and is a piece of cake to implement.

Since our dataset contains outliers, we start the iterative imputation process by utilizing the median value to fill in the gaps. This aids in reducing the impact of outliers since the median is less impacted by them

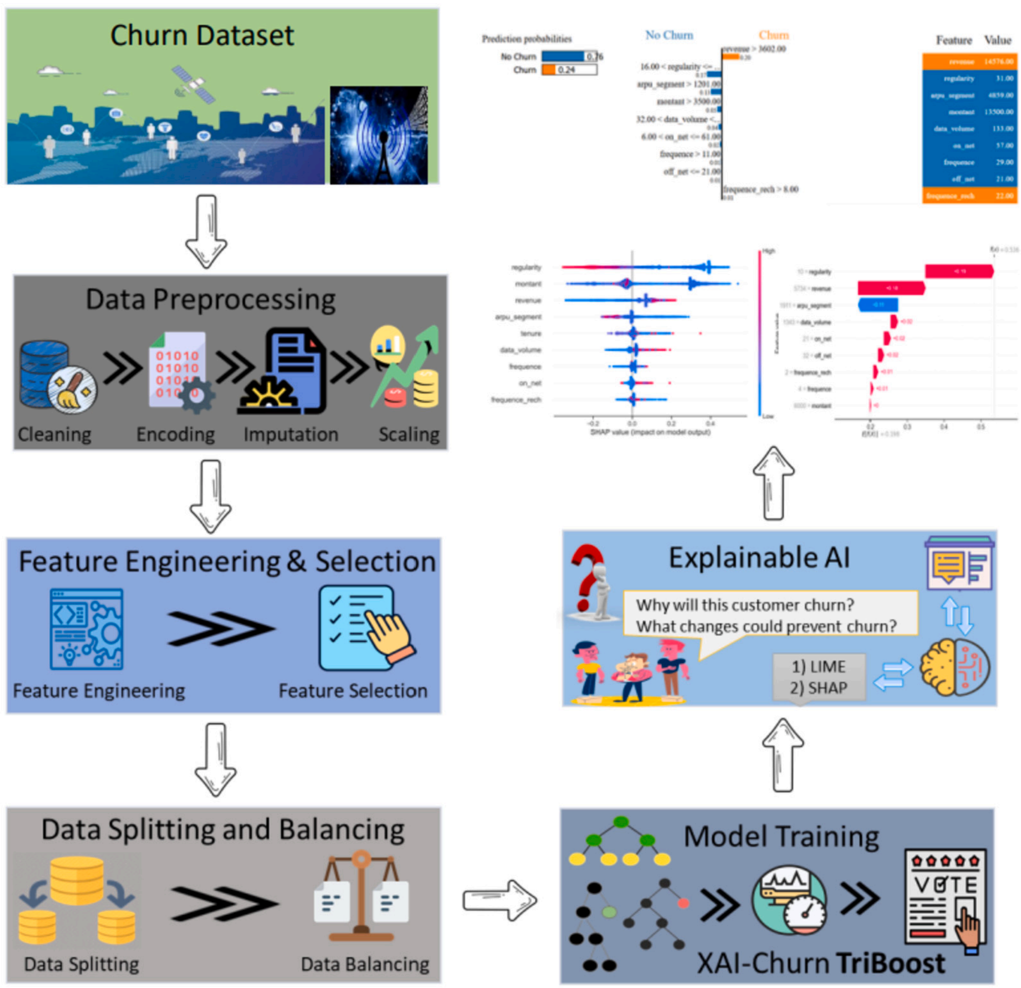

Fig. 1. Workflow of XAI-Churn TriBoost model for the customer churn prediction.

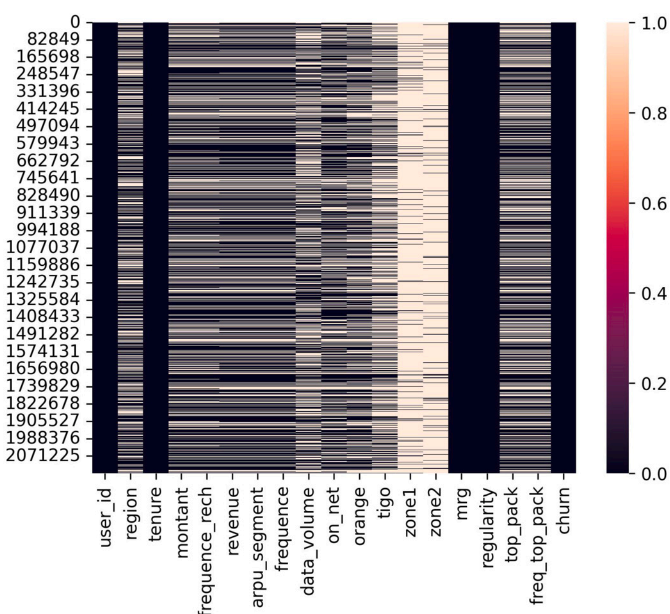

Fig. 2. The missing values in the dataset.

than the mean. Iterative imputation continues for up to twenty iterations after this first step. We guided convergence by setting an absolute tolerance of 0.00001. The method takes this tolerance level and the data scale into account to determine a scaled tolerance. As a convergence threshold, the algorithm in our situation utilized a scaled tolerance of around 18.24. After 15 iterations of iterative imputation, the imputed values stabilized within the scaled tolerance level, signifying convergence. The imputation of missing values was thorough and dependable as a consequence. In Algorithm 1, the algorithm for handling missing data is presented.

Algorithm 1 Iterative Imputation with Bayesian Ridge.

Initialize X_{imputed} with median values for missing entries
for each feature f with missing values **do**
  Initialize model $M_f \leftarrow$ Bayesian Ridge()
  Initialize iterations $\leftarrow 0$
  while mean absolute change > scaled tolerance **and** iterations < 20 **do**
    Fit M_f on X_{imputed}
    Predict missing values for f
    Update X_{imputed} with predicted values
    Compute mean absolute change
    Increment iterations
  end while
end for
Round X_{imputed} and convert to integer type
return X_{imputed}

3.2.4. *Data scaling*

Data scaling is an essential step in ML, as it improves model accuracy and learning efficiency. Our dataset contains many outliers, so we need a careful scaling approach to handle these effectively without distorting the data distribution.

To achieve this, we apply a sequential data scaling approach [6]. First, we use robust scaling to reduce the influence of outliers. Instead of deleting these outliers, we scale them based on the median and interquartile range (IQR). This makes the scaling less sensitive to extreme values. The formula for robust scaling is:

$$\text{Robust Scaling} = \frac{X - \text{median}(X)}{\text{IQR}(X)} \tag{1}$$

In this method, values are centered around the median, which minimizes the impact of outliers compared to mean-based methods.

After robust scaling, we apply standard scaling to standardize features. This scaling adjusts the data to have a mean of 0 and a standard deviation of 1. Standard scaling brings all features to a common scale, improving model performance. The formula for standard scaling is:

$$\text{Standard Scaling} = \frac{X - \mu}{\sigma} \tag{2}$$

Here, μ represents the mean of the feature, and σ is its standard deviation. This approach is useful for models that are sensitive to data magnitude, as it aligns each feature around a normalized range.

Finally, we apply min-max scaling to normalize the features within a fixed range, typically between 0 and 1. This ensures that all features contribute equally during model training, especially in algorithms where feature ranges are significant. The formula for min-max scaling is:

$$\text{Min-max scaling} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{3}$$

where $X_{\min}$ and $X_{\max}$ are the minimum and maximum values of the feature X, respectively. This normalization process retains the relationships within the data while compressing it to a standard range.

We successfully handle outliers and get the dataset ready for the best model performance by using our sequential scaling strategy.

3.3. Feature engineering

ML relies on feature engineering. Incorporating new features or modifying old ones is vital for improving the model. Effective feature engineering can reveal hidden relationships within the dataset. This improves the model's predictive power and makes it easier to understand. To make the model better at pattern-learning, we take useful features out of the input data and employ them.

Our database contains two distinct features: tigo and orange. The orange feature displays calls made to the orange network, whereas the tigo feature displays calls made to the tigo network. Both characteristics record the same data, which is reflective of outbound calls to different networks. We combined them into a single feature representing the overall amount of outgoing calls to external networks in order to simplify the dataset. We define this new feature as:

$$\text{off\_net} = \text{tigo} + \text{orange} \tag{4}$$

By combining call details with this new off_net feature, we might potentially streamline the dataset. Because of this, the model is much better able to spot significant patterns. By incorporating all off-network interactions into the model, instead of just one network at a time, we make it more comprehensive.

3.4. Feature selection

Feature selection is a crucial part of ML that helps improve model performance and interpretability by extracting useful properties from the dataset. Overfitting can be prevented by enhancing model accuracy, streamlining the model, and decreasing dimensionality in tandem. The model is made easier to understand and interpret through feature selection, which simplifies it by concentrating on relevant variables. The reference is [16].

Table 1
Hyperparameters for Boruta Feature Selection.

Hyperparameter	Value
Base Model	Random Forest
Number of Estimators (n_estimators)	100
Percentile (perc)	90
Significance Level (alpha)	0.05
Maximum Iterations (max_iter)	100
Random State (random_state)	42

Table 2
Description of Selected Features for Churn Prediction.

Feature	Description
montant	Top-up amount
frequence_rech	Number of times the customer refilled their account
revenue	Monthly income generated by each client
arpu_segment	Average income over the last three months
frequence	Number of times the client generated income
data_volume	Number of connections
on_net	Number of calls made within the same network
regularity	Number of activities over the last three months
off_net	Number of calls made to other networks
churn	The target variable indicating whether a client becomes inactive and makes no transactions for 90 days.

Feature selection can be done using a variety of methods; however, these methods frequently rely on thresholds or domain knowledge for manual selection, which can lead to bias. While more conventional approaches aid in feature identification, they run the risk of ignoring crucial interactions or depending on subjective judgments. The Boruta feature selection method, which compares original features with randomly produced "shadow" features, is an automated and effective strategy that we utilized to remedy issue. Boruta enhances the objectivity and reliability of the selection process by repeatedly assessing the value of traits, thereby retaining only those that are actually relevant [37]. Table 1 presents the hyperparameters and their values used in the Boruta feature selection strategy.

During each iteration, the Boruta algorithm relies on measures like the mean reduction in accuracy to determine feature relevance. It uses a tree-based classifier as its base model. We built Boruta using RF classifier because of how well its ensemble learning method assesses the relative value of features. To increase stability and decrease variance, the model was set up with 100 estimators, which means it consists of 100 DTs. The percentile of 90 ensures that only features with importance scores higher than at least 90% of the shadow features are retained. We set the significance level to 0.05, establishing a rigorous threshold for filtering out irrelevant features. To balance performance and efficiency, the maximum number of iterations was set to 100, preventing unnecessary computations. Additionally, setting the random state to 42 ensured consistent and reproducible results. This method systematically eliminates irrelevant features, retaining only the most meaningful ones. By leveraging Boruta with the RF model, we selected a reliable subset of features, enhancing the overall performance and interpretability of our model.

Table 2 describes the selected features used for churn prediction modeling in this study. These features were selected using the Boruta feature selection method and are used as input for our model. The statistical description of these features is provided in Table 3. This statistical analysis is crucial for understanding the distribution, variability, and scale of the data, helping to identify patterns, and ensure the dataset is well-prepared for modeling.

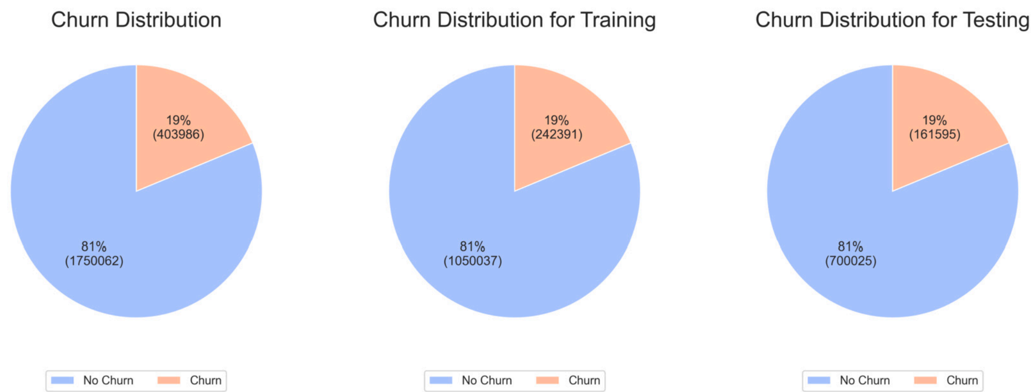

Fig. 3. Churn distribution across the entire dataset, training set, and testing set.

Table 3
Statistical description of selected features.

Feature	Count	Mean	Std	Min	Max
montant	2154048	4483.09	5926.81	0	470000
frequence_rech	2154048	8.87	11.34	0	133
revenue	2154048	4664.51	5970.97	1	532177
arpu_segment	2154048	1554.84	1990.32	0	177392
frequence	2154048	10.99	12.73	0	91
data_volume	2154048	2614.38	9767.18	0	1823866
on_net	2154048	186.90	707.41	0	50899
regularity	2154048	28.04	22.29	1	62
off_net	2154048	87.58	181.74	0	22161

Table 4
Statistical description of the training dataset.

Feature	Count	Mean	Std	Min	Max
montant	1292428	4,487.10	5,926.43	0	365357
frequence_rech	1292428	8.88	11.35	0	131
revenue	1292428	4,668.98	5,977.22	1	397968
apru_segment	1292428	1,556.33	1,992.40	0	132656
frequence	1292428	11.00	12.75	0	91
data_volume	1292428	2,621.68	9,916.46	0	1823866
on_net	1292428	187.18	709.27	0	50899
regularity	1292428	28.05	22.29	1	62
off_net	1292428	87.55	181.13	0	13913

3.5. Data splitting

Proper model evaluation and generalization depend on data splitting. The following equation represents the division of the dataset into two parts:

$$D = D_{training} + D_{testing} \tag{5}$$

In this study, we split the data with 60% allocated for training ($D_{training}$) and 40% for testing ($D_{testing}$). We chose this ratio primarily to reduce the risk of overfitting. When a model is trained on an excessively large portion of the data, it may become too specialized to the training set and fail to generalize well to new data. By allocating 40% of the data for testing, we ensure a more robust evaluation of the model's performance. Since our approach combines multiple boosting models, a larger test set helps assess generalization more effectively. Boosting models are powerful but prone to overfitting if trained on too much data without proper validation. Additionally, our dataset is highly imbalanced, with only 19% of samples representing churn cases. Using a larger test set ensures that we have enough churn samples to accurately measure model performance on minority classes.

To maintain consistency and ensure reproducibility, we set the random state to 42. We also applied stratified sampling to preserve the original class distribution in both the training and testing sets. Fig. 3 illustrates the churn distribution in the original dataset, as well as in the training and testing subsets. In Table 4, we present the statistical description of the training data, and in Table 5, we present the statistical description of the testing data. These tables provide insights into the distribution, variability, and scale of the data.

Table 5
Statistical description of the testing dataset.

Feature	Count	Mean	Std	Min	Max
montant	861620	4477.09	5927.37	0	470000
frequence_rech	861620	8.85	11.31	0	133
revenue	861620	4657.80	5961.58	1	532177
arpu_segment	861620	1552.60	1987.19	0	177392
frequence	861620	10.97	12.70	0	91
data_volume	861620	2603.43	9538.88	0	1702309
on_net	861620	186.47	704.61	0	38648
regularity	861620	28.03	22.28	1	62
off_net	861620	87.62	182.64	0	22161

3.6. Data balancing

Imbalanced datasets are common in ML applications. When a dataset is imbalanced, the minority class represents a small fraction of the overall data. This imbalance can skew model predictions towards the majority class, thus affecting overall performance. To address this issue, several resampling techniques have been developed [30]. Traditional methods involve either duplicating samples from the minority class or reducing the number of samples in the majority class to balance the dataset. However, these approaches can introduce problems such as overfitting or loss of valuable information, making them less effective in many scenarios. Importantly, resampling should be applied only to the training dataset. It might bias the model if applied to the whole dataset, as it would alter the real-world distribution.

The SMOTE method [13] is a popular strategy for dealing with class imbalance. By creating synthetic samples for the minority class, SMOTE avoids the overfitting danger associated with basic oversampling, which involves replicating existing samples. This method improves the model's performance on both classes by creating a training dataset that is balanced or almost balanced [11]. SMOTE creates synthetic samples by combining two close samples in the minority class, denoted as x and x_R. The nearest neighbor x_R is randomly selected from the neighbors of x within the minority class. The synthetic sample is computed as follows:

$$x_{\text{synthetic}} = x + u \times (x_R - x), \tag{6}$$

where $0 \leq u \leq 1$ is a random value.

To address the class imbalance in our dataset, we applied SMOTE using the imbalanced-learn Python library. SMOTE was implemented only on training data using a random state of 42 to ensure reproducibility. The algorithm used 5 nearest neighbors, meaning synthetic samples

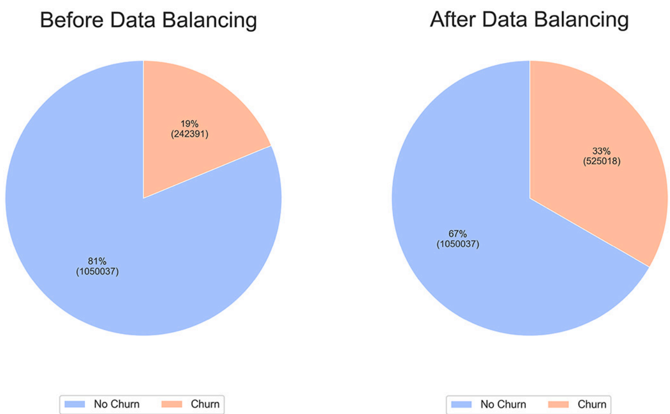

Fig. 4. Churn distribution in the training data before and after balancing with SMOTE.

Table 6
Statistical description of the training data after balancing with SMOTE.

Feature	Count	Mean	Std	Min	Max
montant	1575055	4118.37	5474.63	0	365357
frequence_rech	1575055	7.93	10.55	0	131
revenue	1575055	4337.25	5511.15	1	397968
arpu_segment	1575055	1445.75	1837.05	0	132656
frequence	1575055	9.86	11.89	0	91
data_volume	1575055	2428.81	9345.42	0	1823866
on_net	1575055	156.70	648.11	0	50809
regularity	1575055	24.03	22.22	1	62
off_net	1575055	80.15	165.54	0	13913

were generated based on the 5 closest minority class samples. The sampling strategy was set to 0.5, increasing the minority class size to 50% of the majority class. Initially, the minority class accounted for 19% of the training data, and after applying SMOTE, this proportion increased to 33%, as shown in Fig. 4. Table 6 presents the statistical summary of the training data after balancing with SMOTE. It illustrates how the distribution, variability, and scale of the data changed after the synthetic samples were generated.

3.7. Proposed XAI-Churn TriBoost

Our proposed model, XAI-Churn TriBoost, combines three boosting algorithms: XGBoost, CatBoost, and LightGBM. These algorithms are integrated into a unified soft voting ensemble to predict customer churn. Boosting algorithms are effective in handling complex patterns. Using an ensemble leverages each model's strengths to improve performance. Below is a step-by-step explanation of our methodology, shown alongside the TriBoost algorithm in Algorithm 2.

Algorithm 2 The algorithm of XAI-Churn TriBoost.

Define base models: XGBoost, CatBoost, LightGBM
for each base model **do**
  Set parameter grid
  Perform hyperparameter tuning with RandomizedSearchCV
  Select best parameters
  Train model using best parameters
end for
Apply soft voting with weights $w_1 = 2$, $w_2 = 1$, $w_3 = 3$:

$$\mathbb{P}_{\text{TriBoost}}(c) = \frac{w_1 \cdot \mathbb{P}_{\text{XGBoost}}(c) + w_2 \cdot \mathbb{P}_{\text{CatBoost}}(c) + w_3 \cdot \mathbb{P}_{\text{LightGBM}}(c)}{w_1 + w_2 + w_3}$$

where $\mathbb{P}(c)$ denotes the probability of class c
return Class with highest $\mathbb{P}_{\text{TriBoost}}(c)$

The XAI-Churn TriBoost ensemble consists of three base models. The first model, XGBoost is a popular gradient boosting algorithm known for its scalability and robust handling of overfitting through advanced regularization techniques [14]. It can handle large datasets and non-linear relationships [44]. XGBoost uses decision trees as weak learners, optimizing them through both L1 and L2 regularization. This regularization helps prevent overfitting, making the model generalize better on unseen data. The objective function for XGBoost is defined as:

$$Obj = \sum_{i=1}^{m} l(y_i, \hat{y}_i^{(t-1)} + f_i(x_i)) + \Omega(f_k) \tag{7}$$

Here, l represents the loss function, y_i is the true label, and $\hat{y}_i^{(t-1)}$ is the prediction from the $(t-1)$-th iteration. The function output for sample x_i is denoted by $f_i(x_i)$. The regularization term, $\Omega(f_k) = \gamma T + \frac{1}{2}\lambda w^2$, controls complexity.

The second base model, CatBoost is specifically designed to handle categorical data efficiently without extensive preprocessing [48]. With its recurrent training technique, CatBoost enhances accuracy and stability while decreasing prediction bias through the reduction of error in each succeeding tree [24].

Furthermore, LightGBM is perfect for big datasets due to its quick training and minimal memory consumption. As opposed to traditional level-based tree-growing methods, LightGBM prioritizes the growth of the most informative leaves and grows trees leaf by leaf. Using this approach with LightGBM improves accuracy and speeds convergence while reducing the number of trees needed. In order to decrease memory usage and increase processing performance, LightGBM discretizes continuous variables into bins using a histogram-based method. With high-dimensional data in particular, this enhances performance even further [31].

We employ randomized search CV for hyperparameter tuning to guarantee that all models are operating at their optimal performance. Our extensive dataset enhances the effectiveness of our strategy. By selecting hyperparameters at random, it efficiently explores enormous parameter spaces without having to analyze every conceivable combination, as stated by [9]. We give randomized search CV a range of possible values to each model's hyperparameters and allow it to try out a certain number of combinations. This approach finds the sweet spot between exploration and efficiency, making it ideal for complicated, high-dimensional parameter fields. The final hyperparameters selected for each model are shown in Table 7.

We use a soft voting technique to integrate the predictions of each model after they have been trained. With the use of weighted probabilities, each class's model output can be determined by soft voting. In our XAI-Churn TriBoost model, the final prediction probability for class c is denoted by $\mathbb{P}_{\text{TriBoost}}(c)$. This probability is computed by combining the prediction probabilities from the base models using specific weights. The calculation is defined as follows:

$$\mathbb{P}_{\text{TriBoost}}(c) = \frac{w_1 \cdot \mathbb{P}_{\text{XGBoost}}(c) + w_2 \cdot \mathbb{P}_{\text{CatBoost}}(c) + w_3 \cdot \mathbb{P}_{\text{LightGBM}}(c)}{w_1 + w_2 + w_3} \tag{8}$$

In this equation, $\mathbb{P}_{\text{XGBoost}}(c)$, $\mathbb{P}_{\text{CatBoost}}(c)$, and $\mathbb{P}_{\text{LightGBM}}(c)$ represent the probabilities of class c predicted by the XGBoost, CatBoost, and LightGBM models, respectively. The weights w_1, w_2, and w_3 indicate the relative importance of each model in the ensemble. We assign weights of $w_1 = 2$, $w_2 = 1$, and $w_3 = 3$ to XGBoost, CatBoost, and LightGBM, respectively. These weights reflect the performance characteristics of each model.

The final predicted class for each sample is selected based on the highest probability from $\mathbb{P}_{\text{TriBoost}}(c)$. This method ensures that the class with the strongest consensus across the models is chosen. This approach enhances prediction reliability and improves the overall robustness of the model.

Table 7
Optimal Parameters Selected by RandomizedSearchCV for Each Model.

Model	Optimal Parameters
XGBoost	{'subsample': 0.8, 'reg_lambda': 0, 'reg_alpha': 0.1, 'n_estimators': 300, 'max_depth': 6, 'learning_rate': 0.1, 'gamma': 1, 'colsample_bytree': 1.0}
CatBoost	{'random_strength': 1, 'learning_rate': 0.1, 'l2_leaf_reg': 5, 'iterations': 300, 'depth': 3, 'colsample_bylevel': 1.0, 'bagging_temperature': 1}
LightGBM	{'subsample': 0.8, 'reg_lambda': 0, 'reg_alpha': 0, 'n_estimators': 200, 'min_split_gain': 1, 'max_depth': 6, 'learning_rate': 0.1, 'colsample_bytree': 1.0}

3.8. Explainable artificial intelligence

Methods that simplify complex AI models and reveal the logic behind their predictions are part of the explainable AI area [26]. This is especially important in domains where comprehending model decisions promotes confidence and practical insights, such as customer churn prediction. We accomplish this by utilizing LIME and SHAP.

3.8.1. Local interpretable model-agnostic explanations

Complex ML models can be better understood with the help of LIME. Based on a single prediction, it builds a more straightforward and understandable model. The main concept is to make a local approximation to the model, concentrating just on the region surrounding a specific instance. This clarifies the reasoning for the prediction made by the initial model. LIME generates perturbed samples around the instance, then uses these samples to train an interpretable model such as linear model that reflects the behavior of original model near that instance [49].

LIME works by minimizing a loss function that measures the difference between the original model h and the interpretable model q around the target instance x. The approximation is weighted by proximity $\pi(x')$. The objective function is given by:

$$\delta(x) = \arg\min_{q \in Q} \left(M(h, q, \pi) + \Lambda(q) \right) \tag{9}$$

Here, $M(h, q, \pi)$ quantifies how well the simpler model q approximates h locally. The term $\Lambda(q)$ penalizes complexity to maintain interpretability. LIME assigns higher weights to samples closer to the target instance, ensuring locally faithful explanations. This approach provides actionable insights across text, images, and tabular data [7].

3.8.2. Shapley additive exPlanations

SHAP is a popular explainable AI technique. It provides insights into the contribution of each feature in a model's prediction. In accordance with the principles of cooperative game theory, SHAP gives every attribute a Shapley value. The feature's contribution to the prediction is represented by this value. When it comes to sophisticated ML models, SHAP really shines since transparency is key. It treats the model like a mystery and explains both individual forecasts and the general value of features, the reference [41].

The Shapley value $\phi(j)$ for a feature j is calculated by averaging its marginal contributions across all possible feature combinations. For a set of features F and a model v, the Shapley value $\phi(j)$ is given by:

$$\phi(j) = \frac{1}{|F|!} \sum_{S \subseteq F \setminus \{j\}} \frac{|S|!\,(|F| - |S| - 1)!}{|F|!} \left(v(S \cup \{j\}) - v(S) \right) \tag{10}$$

In this formula, S represents each subset of F that excludes j. The term $v(S \cup \{j\}) - v(S)$ calculates the effect of adding j to subset S. A statistic for the influence of each feature is provided by SHAP by adding contributions across all combinations. This method provides interpretability on a local and global scale [3].

4. Results

4.1. Environment and setup

4.1.1. Programming language and libraries

Python was selected as the primary language for this research because to its adaptability, ease of use, and rich library support. For data analysis and manipulation, Pandas was used, while NumPy facilitated efficient numerical operations and array structures. ML models were built, data was preprocessed and imputed, and model performance was evaluated using Scikit-learn. Seaborn was used to create the data visualizations, guaranteeing that they are both aesthetically pleasing and informative. For feature selection, the Boruta library was employed to identify and retain the most relevant features. To address data imbalance, the imblearn library was utilized, specifically using SMOTE to achieve balanced class distributions.

Various boosting libraries, including LightGBM, CatBoost, and XGBoost, enabled robust model development through advanced gradient boosting techniques. A voting classifier was created using Scikit-learn to combine predictions from multiple models. For model interpretability, explainability libraries such as LIME and SHAP were utilized, providing both local and global insights into feature importance and enabling a deeper understanding of model predictions.

4.1.2. Hardware specifications

All experiments and model training were conducted on an HP ZBook 15 G6 Mobile Workstation. The laptop is equipped with an Intel Core i9-9880H processor, an NVIDIA Quadro T2000 GPU, 32GB of DDR4 RAM, and a 512GB SSD.

4.2. Performance evaluation measures

The performance measures used in this research include accuracy, precision, recall, F1-score, and curve analysis metrics, such as area under the receiver operating characteristic curve (AUC-ROC) and the precision-recall (PR) curve. Additionally, terminologies from the confusion matrix are referenced in calculating these measures, including true positive (TP), true negative (TN), false positive (FP), and false negative (FN). For the task of churn prediction, TP represent correctly predicted churn instances, TN are correctly predicted non-churn instances, FP indicate non-churn instances incorrectly predicted as churn, and FN are churn instances incorrectly predicted as non-churn. Fig. 5 displays the confusion matrix, presenting the prediction results of the XAI-Churn TriBoost model.

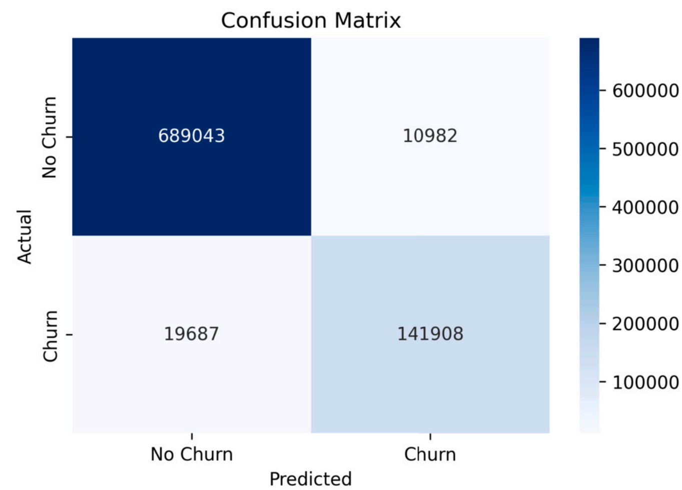

Fig. 5. The confusion matrix of XAI-Churn TriBoost.

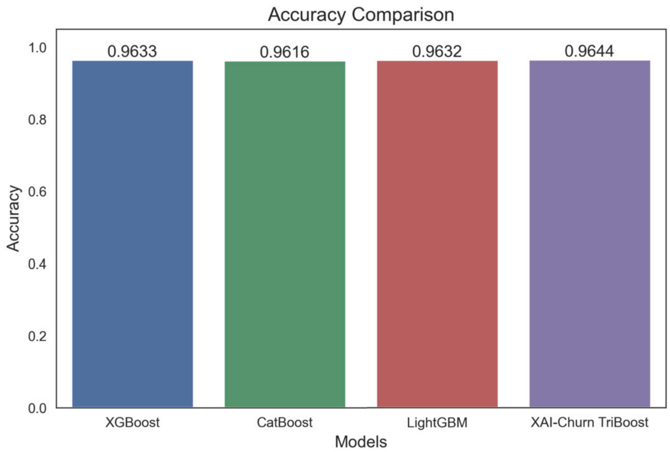

Fig. 6. Comparison of accuracy between XAI-Churn TriBoost and individual base models.

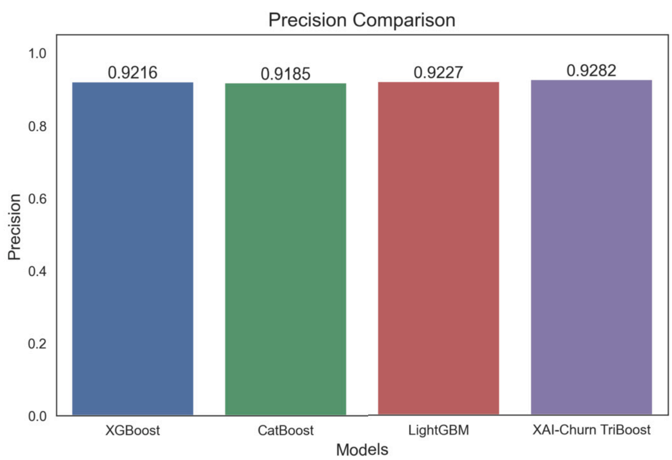

Fig. 7. Comparison of precision between XAI-Churn TriBoost and individual base models.

4.2.1. *Accuracy*

Accuracy is the ratio of correctly predicted instances to the total instances. It reflects the overall prediction accuracy of the XAI-Churn TriBoost model. Accuracy is calculated using Equation:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

Fig. 6 compares the accuracy of our model with other base models. The XAI-Churn TriBoost demonstrates a higher accuracy, indicating superior performance.

4.2.2. *Precision*

Precision is the ratio of correctly identified churn instances to all instances predicted as churn. It measures the ability of model to avoid false positives. Precision is calculated by Equation:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{12}$$

Fig. 7 shows the precision scores of our model and the base models. The XAI-Churn TriBoost model demonstrates a higher precision, reflecting its effectiveness in minimizing false churn predictions.

4.2.3. *Recall*

Recall is the ratio of correctly identified churn cases to all actual churn cases. It measures the ability of model to detect all churn instances. Recall is calculated as shown in Equation:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{13}$$

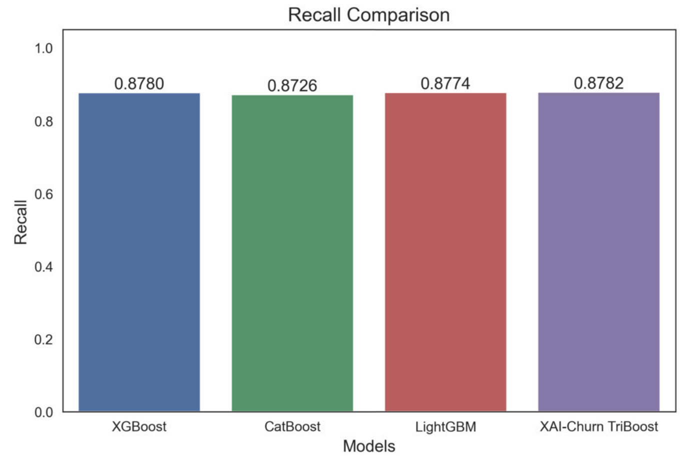

Fig. 8. Comparison of recall between XAI-Churn TriBoost and individual base models.

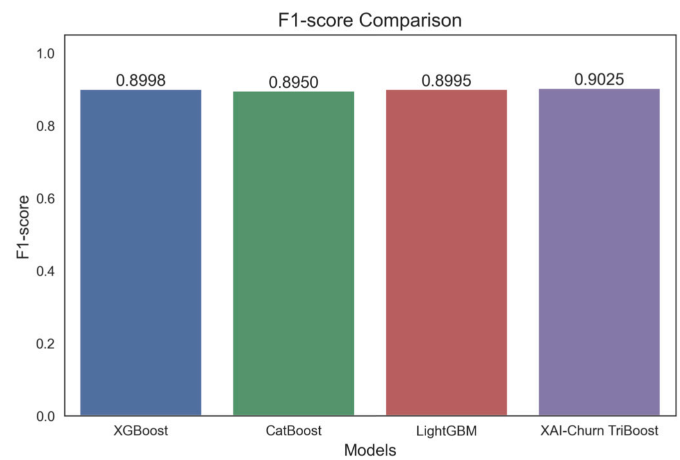

Fig. 9. Comparison of F1-score between XAI-Churn TriBoost and individual base models.

Fig. 8 illustrates the recall scores of the XAI-Churn TriBoost model compared to the base models. The XAI-Churn TriBoost model shows superior recall, indicating its effectiveness in capturing most churn cases.

4.2.4. *F1 score*

The F1 score is the harmonic mean of precision and recall. It balances the ability of model to make precise and complete predictions. The F1 score is calculated using Equation:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{14}$$

Fig. 9 shows the F1-scores for both the base models and the XAI-Churn TriBoost model. The XAI-Churn TriBoost model achieved a high F1 score, which indicates that it effectively handles both false positives and false negatives.

4.2.5. *Curve analysis*

We examine the XAI-Churn TriBoost model's performance utilizing two crucial metrics from curve analysis: AUC-ROC and PR curve. The model's performance when tested against various sets of classification criteria can be better understood with the help of these indicators.

A measure of the model's performance across all thresholds is the AUC-ROC. A one-to-one change in the classification threshold has an impact on the true positive and false positive rates, as seen by the ROC curve. The model performs optimally at 1; at 0.5, it essentially makes an random estimate. Churn prediction and other imbalanced datasets benefit greatly from the PR curve, which shows the correlation between

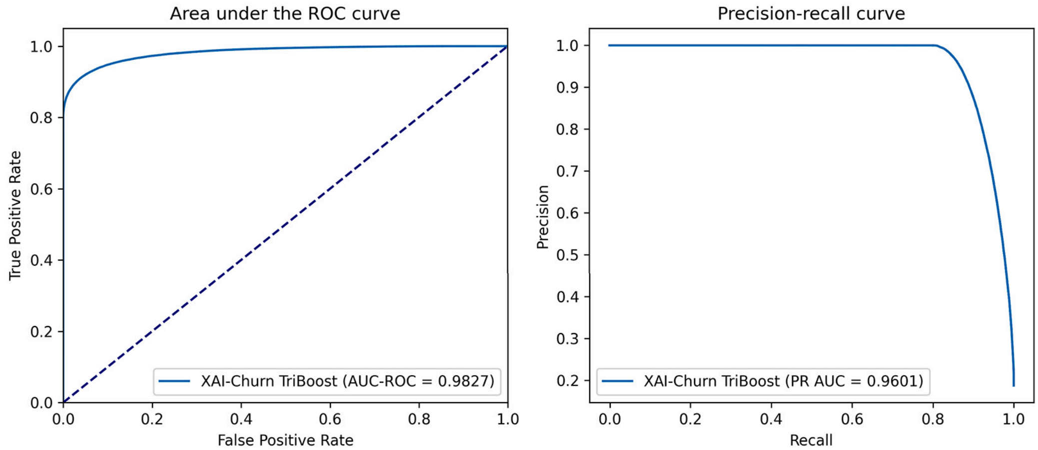

Fig. 10. The area under the ROC and precision-recall curves for XAI-Churn TriBoost.

accuracy and recall at different categorization thresholds. This graph shows that the model can maintain good recall and precision even when the classes are imbalanced.

The AUC-ROC score for the XAI-Churn TriBoost model is 0.9828, and the PR AUC score is 0.9601, as illustrated in Fig. 10. The model's near-perfect results show that it can distinguish between churn and non-churn events and that the balance between recall and precision is well-maintained. This verifies that the model successfully identifies churn cases with few false positives and has good classification performance.

4.3. *Explainability analysis with explainable AI techniques*

Explainable AI has become crucial in understanding complex models. In applications like customer churn prediction, it is important to know why models make specific predictions. This transparency helps build trust and confidence in the model's decisions. Traditional ML models are interpretable but often lack the predictive power needed for complex telecom data. To address this, advanced explainable AI methods like LIME and SHAP are used. These techniques explain individual predictions and highlight feature importance. They make churn prediction models more interpretable, giving stakeholders clear and actionable insights.

4.3.1. *LIME analysis*

Fig. 11 shows a LIME explanation for a churn prediction made by our model. On the right side, it displays the feature values for a specific instance. In the middle, it illustrates the decision-making boundary of the model. Features highlighted in blue contribute to a "no churn" prediction, while those in orange contribute to a "churn" prediction. On the left side, the model's prediction is displayed, showing a 53% probability of churn, with the actual class also being churn. Key features influencing this prediction include revenue, regularity, and arpu segment. Revenue and regularity strongly contribute towards churn, pushing the prediction in that direction. In contrast, features like arpu segment and montant slightly support a "no churn" prediction. However, the overall influence of the orange features results in the final prediction of churn.

Fig. 12 presents a LIME explanation where the model predicts no churn with a 78% probability, which matches the actual class of no churn. In this case, revenue contributes towards churn, but the combined influence of features like regularity, arpu segment, and data_volume favors no churn. These blue contributions strengthen the prediction of no churn.

4.4. *SHAP analysis*

Fig. 13 shows a force plot that provides an initial overview of how each feature contributes to the churn prediction for this instance. The model predicts a churn probability of 0.53. In this plot, features that increase the likelihood of churn are shown in red, while features reducing the likelihood of churn are shown in blue. For this instance, key features like 'regularity', and 'revenue' (in red) push the prediction toward churn, whereas 'apru_segment' (in blue) reduces the churn likelihood.

To gain a more detailed breakdown of each feature's influence, the waterfall plot in Fig. 14 is used. This plot displays the exact numerical impact of each feature on the final prediction score. Here, 'regularity' (+0.21) and 'revenue' (+0.06) are the strongest contributors to the churn prediction, while 'apru_segment' (-0.03) reduces it.

Together, these plots offer a comprehensive view of why the model predicts churn for this specific instance, with the force plot providing an overview and the waterfall plot giving detailed insights into feature contributions.

On a broader level, the SHAP beeswarm plot in Fig. 15 displays the distribution of SHAP values for each feature, illustrating their impact across multiple instances. Each dot represents an instance, with its position along the x-axis indicating the SHAP value. The color gradient from blue to red reflects feature values, where red represents higher values and blue represents lower values. For example, high values of 'regularity' (in red) are associated with a strong positive impact on churn likelihood, pushing predictions toward churn. This plot reveals patterns and the direction of feature influence, with features like 'regularity', and 'montant', and having the most significant impact.

The SHAP feature importance plot in Fig. 16 ranks features based on their average absolute SHAP values, showing their overall contribution to the model's predictions. 'Regularity' and 'montant' are the top features influencing churn prediction, followed by 'arpu segment' and 'revenue'. This plot provides a clear overview of the key drivers of churn, identifying the most critical factors for model interpretation.

Taken as a whole, these SHAP summary charts shed light on the features that have a consistent impact on the model's predictions and the direction of that influence, which helps to comprehend the value of features on a local and global scale.

4.5. *Ablation study*

Here we show the outcomes of an ablation study that used the XAI-Churn TriBoost model to assess the effects of various weight combinations for XGBoost, CatBoost, and LightGBM. A single soft voting ensemble is created by merging these three boosting algorithms in the XAI-Churn TriBoost model. Soft voting uses a weighted probability output from each class's model to determine the final prediction. To ensure optimal performance, it is essential to evaluate the contribution of each model by varying their weights. We conducted experiments by varying the weights of XGBoost, CatBoost, and LightGBM from 1 to 3 and recorded the performance metrics for each combination. Table 8

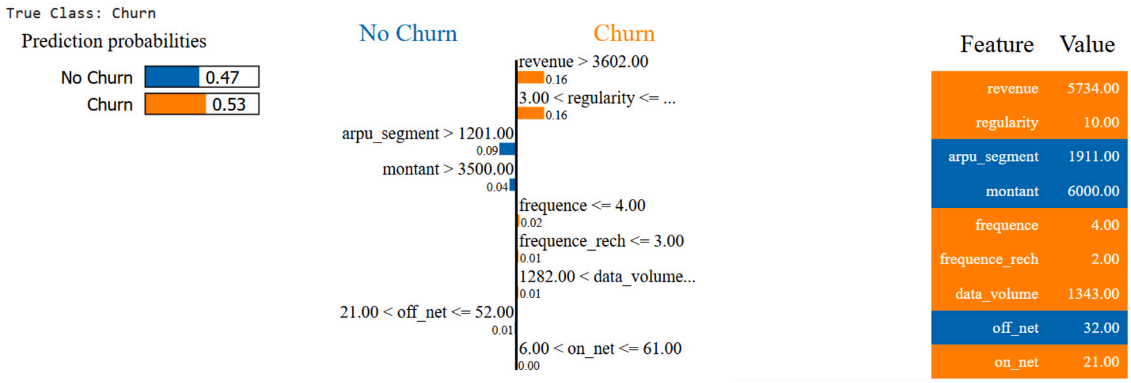

Fig. 11. An example of a LIME explanation for a customer predicted as "Churn" by our proposed model.

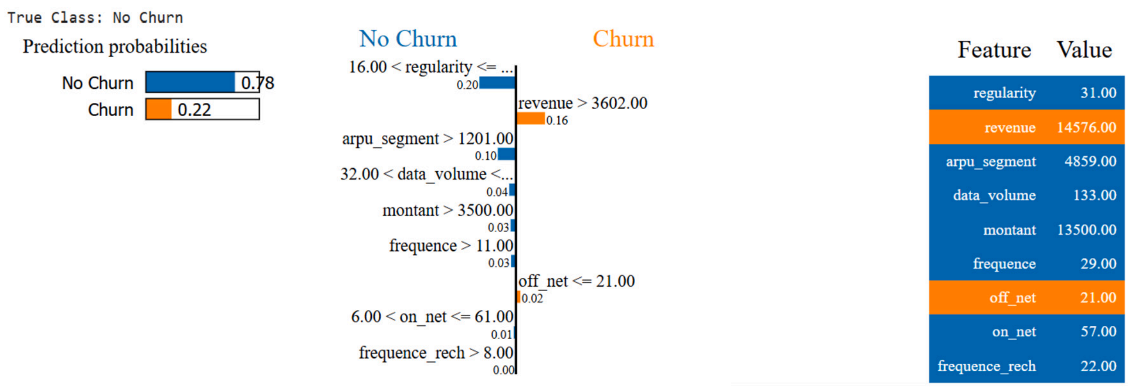

Fig. 12. An example of a LIME explanation for a customer predicted as "No Churn" by our proposed model.

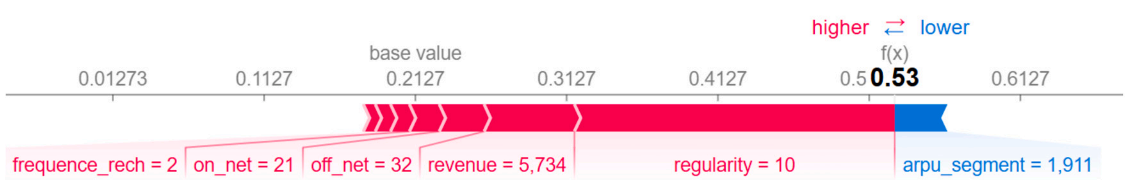

Fig. 13. SHAP force plot for XAI-Churn TriBoost model.

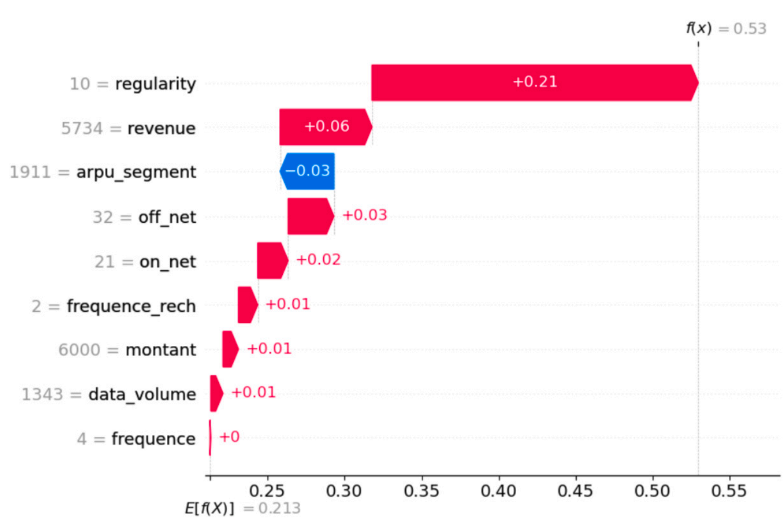

Fig. 14. SHAP waterfall plot for XAI-Churn TriBoost model.

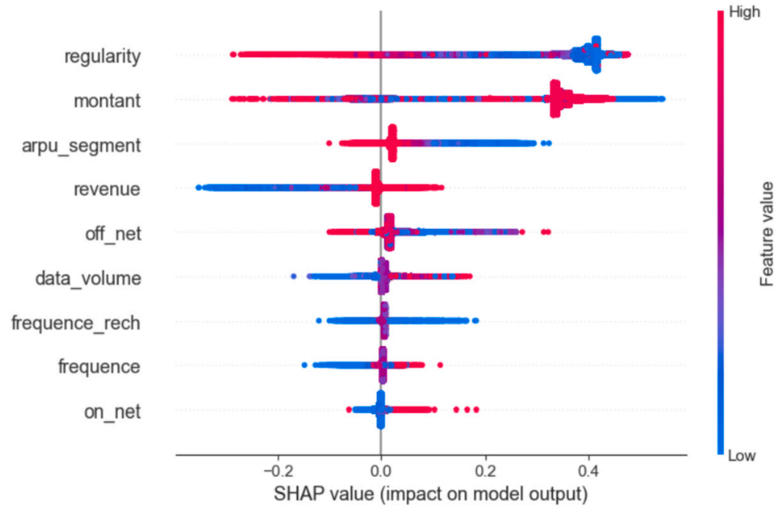

Fig. 15. SHAP beeswarm plot for XAI-Churn TriBoost model.

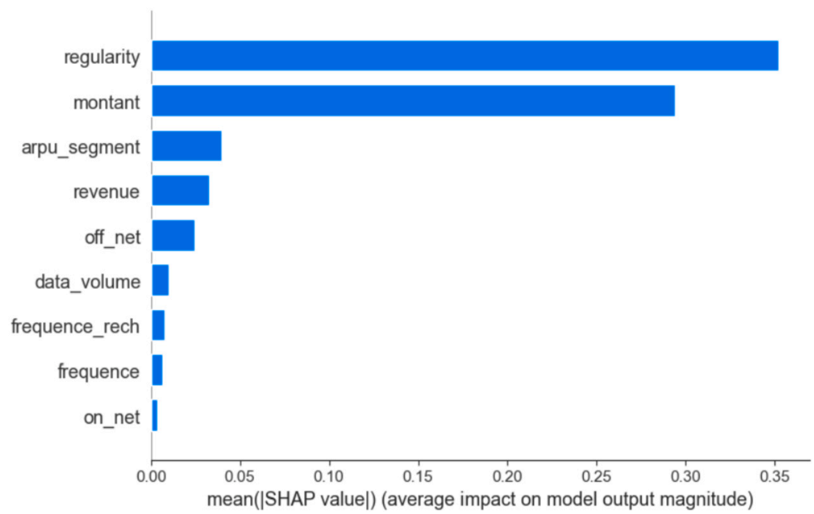

Fig. 16. SHAP feature importance plot for XAI-Churn TriBoost model.

summarizes the results of this ablation study. After analyzing the performance across all weight combinations, we selected the configuration where XGBoost: 2, CatBoost: 1, LightGBM: 3 demonstrated higher performance. This combination was chosen for the final XAI-Churn TriBoost model, as it consistently delivered superior results across multiple evaluation criteria.

5. Discussion

Customer churn is a significant problem in the TCI since retaining clients is crucial for profitability. ML has the potential to identify patterns in large datasets, enabling companies to anticipate customer churn early. This allows businesses to take strategic actions to retain customers before they leave. ML has proven to be highly effective in making businesses more profitable in such industries. However, one of the key challenges is that ML models require large amounts of data and often function as “black boxes,” making it difficult for industry professionals to fully understand their predictions. This lack of interpretability creates trust issues, making explainable AI essential.

In this research, we propose a model named XAI-Churn TriBoost, which integrates XGBoost, CatBoost, and LightGBM in a soft voting ensemble. The study is conducted on a dataset containing over two million records, with 60% of the data used for training and 40% for testing. The dataset contains a significant number of missing values, which we address using iterative imputation with Bayesian Ridge regression, instead of traditional imputation techniques. Furthermore, the dataset exhibits a high number of outliers and varying feature ranges. To handle this, we apply sequential scaling, where we first use Robust Scaling to minimize the impact of outliers, followed by Standardization, and finally, Min-Max Scaling to normalize the data within the range of 0 to 1.

Initially, the dataset consisted of 17 features, but after thorough data cleaning, removal of irrelevant features, and feature engineering, we reduced the number to 10 features. To ensure that only the most relevant and important features are retained, we employ the Boruta feature selection approach, an automated and effective method that compares original features with randomly generated “shadow” features. This process eliminates one feature, leaving us with nine selected features, which are then used for modeling. For modeling, we utilize tree-based algorithms, including XGBoost, CatBoost, and LightGBM, as they are well-suited for handling large and complex datasets. To optimize model performance, we apply Randomized Search CV for hyperparameter tuning. The predictions from these three models are then combined using a soft voting ensemble, which enhances overall accuracy and robustness.

Our proposed model outperforms individual base models, achieving an accuracy of 0.9644, precision of 0.9282, recall of 0.8782, and an F1-score of 0.9025. Additionally, we compare our approach with several state-of-the-art models, including DTs, RF, LR, Gaussian NB, Bernoulli NB, Ridge Classifier, and Stochastic Gradient Descent. The

Table 8
Ablation study results for different weight combinations in XAI-Churn TriBoost.

XGBoost	CatBoost	LightGBM	Accuracy	Precision	Recall	F1-Score
1	1	1	0.9642	0.9281	0.8774	0.9020
1	1	2	0.9643	0.9280	0.8777	0.9022
1	1	3	0.9643	0.9278	0.8780	0.9022
1	2	1	0.9639	0.9270	0.8764	0.9010
1	2	2	0.9641	0.9274	0.8772	0.9016
1	2	3	0.9641	0.9275	0.8774	0.9018
1	3	1	0.9635	0.9259	0.8753	0.8999
1	3	2	0.9638	0.9264	0.8764	0.9007
1	3	3	0.9640	0.9271	0.8770	0.9013
2	1	1	0.9644	0.9283	0.8778	0.9024
2	1	2	0.9644	0.9280	0.8781	0.9024
2	1	3	0.9644	0.9282	0.8782	0.9025
2	2	1	0.9642	0.9282	0.8770	0.9019
2	2	2	0.9642	0.9281	0.8774	0.9020
2	2	3	0.9643	0.9280	0.8776	0.9021
2	3	1	0.9640	0.9275	0.8763	0.9012
2	3	2	0.9641	0.9278	0.8770	0.9017
2	3	3	0.9642	0.9277	0.8772	0.9018
3	1	1	0.9643	0.9278	0.8782	0.9023
3	1	2	0.9644	0.9280	0.8782	0.9024
3	1	3	0.9644	0.9281	0.8783	0.9025
3	2	1	0.9643	0.9280	0.8776	0.9021
3	2	2	0.9644	0.9284	0.8777	0.9023
3	2	3	0.9644	0.9283	0.8778	0.9024
3	3	1	0.9641	0.9280	0.8768	0.9017
3	3	2	0.9642	0.9282	0.8771	0.9019
3	3	3	0.9642	0.9281	0.8774	0.9020

Table 9
Performance Metrics of Different Models.

Model	Accuracy	Precision	Recall	F1 Score
Decision Tree	0.9395	0.8070	0.8907	0.8468
Logistic Regression	0.8200	0.5125	0.8304	0.6338
Gaussian Naive Bayes	0.5682	0.2988	**0.9672**	0.4565
Bernoulli Naive Bayes	0.7952	0.4680	0.6726	0.5519
Ridge Classifier	0.8011	0.4828	0.8522	0.6164
Stochastic Gradient Descent	0.7993	0.4803	0.8532	0.6146
XAI-Churn TriBoost	**0.9644**	**0.9282**	0.8782	**0.9025**

same preprocessing steps were applied to all these models to ensure a fair comparison. The results, presented in Table 9, show that Gaussian NB achieves the highest recall (0.9672) due to its probabilistic classification approach and assumption of feature independence. This characteristic makes it more likely to classify borderline cases as positive, thereby reducing false negatives and increasing recall. However, this improvement comes at the cost of lower precision and overall accuracy.

In contrast, our proposed XAI-Churn TriBoost model delivers the best overall performance, achieving the highest accuracy (0.9644), precision (0.9282), and F1-score (0.9025). While models such as DT, Ridge Classifier, and Stochastic Gradient Descent perform well in certain aspects, they do not maintain the same level of balance across all evaluation metrics. The strength of XAI-Churn TriBoost lies in its ability to effectively identify churners while minimizing false positives, making it the most reliable and practical choice for businesses focused on customer retention strategies.

Since explainability is crucial in this domain, we incorporate LIME and SHAP. These techniques provide interpretability at both the individual prediction level and the overall model level, helping stakeholders understand the reasoning behind predictions. The SHAP analysis indicates that regularity and montant are the most important features influencing customer churn. Regularity refers to the number of activities in the last three months, while montant represents the top-up amount. Based on these findings, it appears that churn risk is highly dependent on consumer behavior patterns and transaction quantities.

In this research, our focus is solely on churn predictive modeling for TCI. However, in recent years, numerous studies have explored churn prediction across various industries, including banking and financial services, e-commerce and retail, and streaming and entertainment. These studies have applied different ML techniques to identify factors influencing customer churn and improve retention strategies.

An example of this is the churn prediction model that [5] created for a Software-as-a-Service inventory management company in Thailand. After implementing various ML models, they discovered that RFs outperformed the rest with a recall of 0.916 and an F1-score of 0.926 when utilizing 10-fold CV. Similarly, [17] constructed a predictive model to discern possible employee churners at a software company based in Portugal. According to their findings, XGBoost outperformed the other methods with ROC AUC of 0.86 and recall of 0.85.

In Bengaluru, India, [23] created a model to forecast client attrition in the online food delivery industry. They found that the NB and RFs classifiers had the best AUC score of 0.952 out of all the ML models they evaluated. According to [53], a churn prediction was developed in the banking industry using XGBoost, which resulted in an accuracy of 0.839, sensitivity of 0.601, specificity of 0.903, and F1-score of 0.613. Similarly, [55] used LightGBM to apply several ML models and achieved accuracy of 0.91, precision of 0.94, recall of 0.87, and F1-score of 0.90.

The gaming business has also investigated churn prediction. A model for predicting player attrition in a mobile video game that is free to play was created by [43]. Their results demonstrated that the RFs model attained accuracies ranging from 0.66 to 0.95, with the range being 1-7 days for the projected churn time.

Churn prediction in the TCI sector has been the subject of multiple research, each using a unique dataset and ML method. To illustrate, Stochastic Gradient Boosting was used by [47]on 7,043 samples containing 20 features, yielding 0.981 accuracy, 0.928 precision, and 0.927 recall. In a similar vein, [35] achieved 0.981 accuracy, 0.928 precision, and 0.927 recall using an RFs model on 2,000 samples with 15 characteristics.

The discussed studies highlight the diverse applications of churn prediction models across multiple industries, each employing different ML techniques and datasets. To provide a concise comparison, Table 10

Table 10
Churn prediction studies across different industries.

Study	Industry	Model	Results
[5]	Software	Random Forest	0.916 recall, 0.926 F1-score
[17]	Software	XGBoost	0.85 recall, 0.86 ROC AUC
[23]	Food Delivery	Naïve Bayes	0.952 AUC score
[53]	Banking	XGBoost	0.839 accuracy, 0.601 sensitivity, 0.903 specificity, 0.613 F1-score
[55]	Banking	LightGBM	0.91 accuracy, 0.94 precision, 0.87 recall, 0.90 F1-score
[43]	Gaming	Random Forest	0.66-0.95 accuracy (varied by churn period)
[47]	Telecom	Stochastic Gradient Boosting	0.79 accuracy, 0.84 AUC
[35]	Telecom	Random Forest	0.981 accuracy, 0.928 precision, 0.927 recall
This Study	Telecom	Soft Voting (XGBoost + CatBoost + LightGBM)	0.9644 accuracy, 0.9282 precision, 0.8782 recall, 0.9025 f1-score

summarizes these studies, outlining their respective industries, methodologies, and key findings.

In this research, our primary focus is on churn prediction within the TCI. We utilize a single, large-scale dataset containing over 2 million records to develop and evaluate our proposed model. However, we intend to expand our analysis in subsequent work by evaluating the model's effectiveness across several industries and by integrating additional datasets. This will guarantee that our method is applicable outside of the TCI industry and assist determine its relative merits and shortcomings.

Our goal is to evaluate the model using data from other locations so that we can account for regional variances in customer behavior and further enhance its generalizability. In addition, we intend to build a web app that can link to real-time data sources and incorporate monitoring capabilities. Businesses will be able to monitor important indicators in real time, evaluate client behavior, and measure customer attrition rates with this application. This will allow them to make proactive decisions.

The application's comparative analysis features will enable businesses to categorise their clients into various categories based on demographics, product usage, and behavioral trends. This tool will assist in identifying the consumer segments most prone to subscription cancellation, enabling targeted retention strategies. We plan to implement a feedback loop enabling users to provide insights on the model's predictions to ensure ongoing improvement. The model will be periodically retrained utilising this accumulated feedback to adapt to evolving customer behaviors and enhance its predictive accuracy.

The model can stay relevant by constantly adding new data and identifying trends and patterns, so it can handle situations when external influences, such economic turmoil or changes in the market, affect consumer preferences. Additionally, automating this process will boost adaptability by improving predicted accuracy and minimising dependency on human intervention. Ultimately, this approach enhances model accuracy, which benefits decision-making, retention strategy customisation, and long-term industry reliability.

6. Conclusion

The purpose of this research was to propose a model for predicting telecom customer churn called XAI-Churn TriBoost. This ensemble model leverages soft voting and combines XGBoost, CatBoost, and LightGBM. Data quality was ensured by extensive preparation methods using a dataset of over 2 million records. With an F1-score of 0.9025, an accuracy of 0.9644, precision of 0.9282, recall of 0.8782. It was trained on 1.58 million records and tested on over 0.86 million records. Integrating explainable AI approaches improved transparency. LIME and SHAP were used to explain individual predictions and highlight feature importance, improving interpretability and building stakeholder trust. SHAP analysis revealed that regularity and montant were the most influential features affecting customer churn, providing actionable insights for retention strategies. The strong performance of XAI-Churn TriBoost underscores the importance of interpretable models in addressing churn prediction challenges.

In future work, we aim to extend the study by incorporating multiple datasets from different industries to assess the model's generalizability. Additionally, we plan to develop a web-based application with real-time monitoring to track churn rates and customer behavior, facilitating proactive decision-making. A feedback loop will also be implemented to refine predictions by periodically retraining the model with new user insights, ensuring adaptability to evolving trends and sustained predictive accuracy.

Abbreviations

The following abbreviations are used in this manuscript:

Acronym	Description
TCI	Telecommunication Industry
ML	Machine Learning
AI	Artificial Intelligence
CV	Cross Validation
RF	Random Forest
DTs	Decision Trees
LR	Logistic Regression
NB	Naïve Bayes
XGBoost	Extreme Gradient Boosting
CatBoost	Categorical Boosting
SVM	Support Vector Machine
KNN	k-Nearest Neighbor
LightGBM	Light Gradient Boosting Machine
MRG	Moving to Rival Group
IQR	Interquartile Range
SMOTE	Synthetic Minority Oversampling Technique
LIME	Local Interpretable Model-Agnostic Explanations
SHAP	SHapley Additive exPlanations
AUC	Area Under the Curve
ROC	Receiver Operating Characteristic
PR	Precision-Recall
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

CRediT authorship contribution statement

Daniyal Asif: Writing – original draft, Data curation, Visualization, Methodology, Conceptualization, Validation, Software, Formal analysis. **Muhammad Shoaib Arif:** Writing – review & editing, Methodology, Funding acquisition, Validation, Conceptualization, Formal analysis. **Aiman Mukheimer:** Writing – review & editing, Project administration, Supervision, Resources.

Funding

This research did not receive any specific grant from public, commercial, or not-for-profit funding agencies.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to acknowledge the support of Prince Sultan University for paying the Article Processing Charges (APC) of this publication.

Data availability

The manuscript included all required data and implementing information.

References

[1] A. Ahmed, D. Maheswari, Churn prediction on huge telecom data using hybrid firefly-based classification, Egypt. Inform. J. 18 (2017) 215–220, https://doi.org/10.1016/j.eij.2017.02.002.
[2] A. Al-shawafi, H. Zhu, S.A. Laqsum, S.I. Haruna, Y.E. Ibrahim, Improved static and impact properties of uhpfrc retrofitted with pu grout materials: experiments and ml algorithms, Results Eng. 23 (2024) 102655.
[3] H. Alkadhim, M. Amin, W. Ahmad, K. Khan, S. Nazar, M. Faraz, M. Imran, Evaluating the strength and impact of raw ingredients of cement mortar incorporating waste glass powder using machine learning and shapley additive explanations (shap) methods, Materials 15 (2022) 7344, https://doi.org/10.3390/ma15207344.
[4] A. Alzubaidi, E. Al-Shamery, Predicting customer churn in telecom sector based on penalization techniques and ensemble machine learning, Int. J. Eng. Technol. 7 (2018) 657–664, https://doi.org/10.14419/ijet.v7i4.19.27977.
[5] P. Amornvetchayakul, N. Phumchusri, Customer churn prediction for a software-as-a-service inventory management software company: a case study in Thailand, in: 2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA), 2020, pp. 514–518.
[6] M. Arif, A. Mukheimer, D. Asif, Enhancing the early detection of chronic kidney disease: a robust machine learning model, Big Data Cogn. Comput. 7 (2023) 144, https://doi.org/10.3390/bdcc7030144.
[7] M. Arif, A. Ur Rehman, D. Asif, Explainable machine learning model for chronic kidney disease prediction, Algorithms 17 (2024) 443, https://doi.org/10.3390/a17100443.
[8] A. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, R. Chatila, Explainable artificial intelligence (xai): concepts, taxonomies, opportunities and challenges toward responsible ai, Inf. Fusion 58 (2020) 82–115, https://doi.org/10.1016/j.inffus.2019.12.012.
[9] D. Asif, M. Bibi, M. Arif, A. Mukheimer, Enhancing heart disease prediction through ensemble learning techniques with hyperparameter optimization, Algorithms 16 (2023) 308, https://doi.org/10.3390/a16060308.
[10] J. Bhattacharyya, M. Dash, What do we know about customer churn behaviour in the telecommunication industry? A bibliometric analysis of research trends, 1985–2019, FIIB Bus. Rev. 11 (2022) 280–302, https://doi.org/10.1177/23197145221116855.
[11] R. Blagus, L. Lusa, Smote for high-dimensional class-imbalanced data, BMC Bioinform. 14 (2013) 106, https://doi.org/10.1186/1471-2105-14-106.
[12] B. Borja, C. Bernardino, C. Alex, G. Ricard, M. David, The architecture of a churn prediction system based on stream mining, in: Frontiers in Artificial Intelligence and Applications, 2013, pp. 157–166.
[13] N. Chawla, K. Bowyer, L. Hall, W. Kegelmeyer, Smote: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357, https://doi.org/10.1613/jair.953.
[14] T. Chen, C. Guestrin, Xgboost: a scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 2016, pp. 785–794.
[15] D. Cumming, S. Johan, R. Reardon, Global fintech trends and their impact on international business: a review, Multinat. Bus. Rev. 31 (2023) 413–436, https://doi.org/10.1108/MBR-2023-0071.
[16] P. Dhal, C. Azad, A comprehensive survey on feature selection in the various fields of machine learning, Appl. Intell. 52 (2022) 4543–4581, https://doi.org/10.1007/s10489-021-02550-9.
[17] J.R. Dias, N. Antonio, Predicting customer churn using machine learning: a case study in the software industry, J. Market. Anal. (2023) 1–17.
[18] Y. Dong, C. Peng, Principled missing data methods for researchers, SpringerPlus 2 (2013) 222, https://doi.org/10.1186/2193-1801-2-222.
[19] V. E, P. Ravikumar, An efficient technique for feature selection to predict customer churn in telecom industry, in: Proceedings of the 1st International Conference on Advances in Information Technology (ICAIT), IEEE, 2019, pp. 174–179.
[20] S. García, S. Ramírez-Gallego, J. Luengo, Big data preprocessing: methods and prospects, Big Data Anal. 1 (2016) 9, https://doi.org/10.1186/s41044-016-0014-0.
[21] A. Gaur, R. Dubey, Predicting customer churn in telecom sector using various machine learning techniques, in: Proceedings of the International Conference on Advanced Computation and Telecommunication (ICACAT), IEEE, 2018, pp. 1–5.
[22] L. Geiler, S. Affeldt, M. Nadif, A survey on machine learning methods for churn prediction, Int. J. Data Sci. Anal. 14 (2022) 217–242, https://doi.org/10.1007/s41060-021-00266-x.
[23] J. Gerald Manju, A. Dharini, B. Kiruthika, A. Malini, Online food delivery customer churn prediction: a quantitative analysis on the performance of machine learning classifiers, in: International Conference on Data Analytics and Management, 2023, pp. 95–104.
[24] J. Hancock, T. Khoshgoftaar, Catboost for big data: an interdisciplinary review, J. Big Data 7 (2020) 94, https://doi.org/10.1186/s40537-020-00369-8.
[25] M.A. Haque, K.H. Nahin, J.H. Nirob, M.K. Ahmed, N.S.S. Singh, L.C. Paul, A.D. Algarni, M. ElAffendi, A.A. Ateya, Machine learning-based technique for directivity prediction of a compact and highly efficient 4-port mimo antenna for 5g millimeter wave applications, Results Eng. 24 (2024) 103106.
[26] V. Hassija, V. Chamola, A. Mahapatra, Interpreting black-box models: a review on explainable artificial intelligence, Cogn. Comput. 16 (2024) 45–74, https://doi.org/10.1007/s12559-023-10179-8.
[27] S. Hung, D. Yen, H. Wang, Applying data mining to telecom churn management, Expert Syst. Appl. 31 (2006) 515–524, https://doi.org/10.1016/j.eswa.2005.09.080.
[28] H. Jain, A. Khunteta, S. Srivastava, Churn prediction in telecommunication using logistic regression and logit boost, Proc. Comput. Sci. 167 (2020) 101–112, https://doi.org/10.1016/j.procs.2020.03.187.
[29] H. Jain, A. Khunteta, S. Srivastava, Telecom churn prediction and used techniques, datasets and performance measures: a review, Telecommun. Syst. 76 (2021) 613–630, https://doi.org/10.1007/s11235-021-00762-y.
[30] G. James, D. Witten, T. Hastie, R. Tibshirani, J. Taylor, Resampling methods, in: An Introduction to Statistical Learning, Springer, Cham, 2023.
[31] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T. Liu, Lightgbm: a highly efficient gradient boosting decision tree, in: Advances in Neural Information Processing Systems, 2017.
[32] A. Keramati, R. Jafari-Marandi, M. Aliannejadi, I. Ahmadian, M. Mozaffari, U. Abbasi, Improved churn prediction in telecommunication industry using data mining techniques, Appl. Soft Comput. 24 (2014) 994–1012, https://doi.org/10.1016/j.asoc.2014.07.001.
[33] L. Khalid, A. Mohsin Abdulazeez, D. Zeebaree, F. Ahmed, D. Zebari, Customer churn prediction in telecommunications industry based on data mining, in: Proceedings of the IEEE Symposium on Industrial Electronics & Applications (ISIEA), IEEE, 2021, pp. 1–6.
[34] G. Kraljević, S. Gotovac, Modeling data mining applications for prediction of prepaid churn in telecommunication services, Automatika 51 (2010) 275–283, https://doi.org/10.1080/00051144.2010.11828381.
[35] R. Krishna, D. Jayanthi, D.S. Sam, K. Kavitha, N.K. Maurya, T. Benil, Application of machine learning techniques for churn prediction in the telecom business, Results Eng. 24 (2024) 103165.
[36] A. Kumar, D. Chandrakala, A survey on customer churn prediction using machine learning techniques, Int. J. Comput. Appl. 154 (2016) 13–16, https://doi.org/10.5120/ijca2016912237.
[37] M. Kursa, W. Rudnicki, Feature selection with the boruta package, J. Stat. Softw. 36 (2010) 1–13, https://doi.org/10.18637/jss.v036.i11.
[38] P. Lalwani, M. Mishra, J. Chadha, Customer churn prediction system: a machine learning approach, Computing 104 (2022) 271–294, https://doi.org/10.1007/s00607-021-00908-y.
[39] W. Lin, C. Tsai, Missing value imputation: a review and analysis of the literature (2006–2017), Artif. Intell. Rev. 53 (2020) 1487–1509, https://doi.org/10.1007/s10462-019-09709-4.
[40] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, Z. Lin, When machine learning meets privacy: a survey and outlook, ACM Comput. Surv. 54 (2021) 1–36, https://doi.org/10.1145/3446376.
[41] S. Lundberg, S.I. Lee, A unified approach to interpreting model predictions, in: Neural Information Processing Systems, 2017.
[42] L. Lwakatare, A. Raj, I. Crnkovic, J. Bosch, H. Holmström Olsson, Large-scale machine learning systems in real-world industrial settings: a review of challenges and solutions, Inf. Softw. Technol. 127 (2020) 106368, https://doi.org/10.1016/j.infsof.2020.106368.
[43] K. Mustač, K. Bačić, L. Skorin-Kapov, M. Sužnjević, Predicting player churn of a free-to-play mobile video game using supervised machine learning, Appl. Sci. 12 (2022) 2795.
[44] T.V. Nagaraju, S. Mantena, M. Azab, S.S. Alisha, C. El Hachem, M. Adamu, P.S.R. Murthy, Prediction of high strength ternary blended concrete containing different silica proportions using machine learning approaches, Results Eng. 17 (2023) 100973.

[45] E. Ogidiaka, F. Ogwueleka, Over-the-top services (ott) on telecommunication operators in Nigeria: exploring consumers' behaviour, Int. J. Inf. Technol. 12 (2020) 437–446, https://doi.org/10.1007/s41870-020-00442-y.

[46] J. Pamina, R. Beschi, S. SathyaBama, S. Soundarya, M. Sruthi, S. Kiruthika, V. Aiswaryadevi, G. Priyanka, An effective classifier for predicting churn in telecommunication, J. Adv. Res. Dyn. Control Syst. 11 (Special Isssue) (2019). Available online, https://ssrn.com/abstract=3399937.

[47] B. Prabadevi, R. Shalini, B.R. Kavitha, Customer churning analysis using machine learning algorithms, Int. J. Intel. Netw. 4 (2023) 145–154.

[48] L. Prokhorenkova, G. Gusev, A. Vorobev, A. Dorogush, A. Gulin, Catboost: unbiased boosting with categorical features, in: Advances in Neural Information Processing Systems, 2018.

[49] M. Ribeiro, S. Singh, C. Guestrin, Why should I trust you? explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 2016, pp. 1135–1144.

[50] C. Rudin, Why black box machine learning should be avoided for high-stakes decisions, in brief, Nat. Rev. Methods Primers 2 (2022) 81, https://doi.org/10.1038/s43586-022-00112-8.

[51] F. Sacco, The evolution of the telecom infrastructure business: unchartered waters ahead of great opportunities, in: F. Sacco (Ed.), Disruption in the Infrastructure Sector: Challenges and Opportunities for Developers, Investors and Asset Managers, Springer, Cham, Switzerland, 2020, pp. 87–148.

[52] S. Saleh, S. Saha, Customer retention and churn prediction in the telecommunication industry: a case study on a Danish University, Soc. Netw. Anal. Appl. Sci. 5 (2023) 173, https://doi.org/10.1007/s13278-023-01045-7.

[53] P.P. Singh, F.I. Anik, R. Senapati, A. Sinha, N. Sakib, E. Hossain, Investigating customer churn in banking: a machine learning approach and visualization app for data science and management, Data Sci. Manag. 7 (2024) 7–16.

[54] I. Ullah, B. Raza, A. Malik, M. Imran, S. Islam, S. Kim, A churn prediction model using random forest: analysis of machine learning techniques for churn prediction and factor identification in telecom sector, IEEE Access 7 (2019) 60134–60149, https://doi.org/10.1109/ACCESS.2019.2914999.

[55] A.G. Văduva, S.V. Oprea, A.M. Niculae, A. Bâra, A.I. Andreescu, Improving churn detection in the banking sector: a machine learning approach with probability calibration techniques, Electronics 13 (2024) 4527.

[56] C. Wei, I. Chiu, Turning telecommunications call details to churn prediction: a data mining approach, Expert Syst. Appl. 23 (2002) 103–112, https://doi.org/10.1016/S0957-4174(02)00030-1.

[57] M. Yildiz, S. Varlı, Customer churn prediction in telecommunication, in: Proceedings of the 23rd Signal Processing and Communications Applications Conference (SIU), IEEE, 2015, pp. 256–259.

[58] M. Zhao, Q. Zeng, M. Chang, Q. Tong, J. Su, A prediction model of customer churn considering customer value: an empirical research of telecom industry in China, Discrete Dyn. Nat. Soc. 2021 (2021) 7160527, https://doi.org/10.1155/2021/7160527.

[59] Zindi, Expresso churn prediction challenge, https://zindi.africa/competitions/expresso-churn-prediction/data. (Accessed 20 October 2024).

[60] M. Óskarsdóttir, C. Bravo, W. Verbeke, C. Sarraute, B. Baesens, J. Vanthienen, Social network analytics for churn prediction in telco: model building, evaluation and network architecture, Expert Syst. Appl. 85 (2017) 204–220, https://doi.org/10.1016/j.eswa.2017.04.024.

[61] J. Černevičienė, A. Kabašinskas, Explainable artificial intelligence (xai) in finance: a systematic literature review, Artif. Intell. Rev. 57 (2024) 216, https://doi.org/10.1007/s10462-024-10276-9.