IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 12, No. 12, December, 2025, pp. 4373-4386
Journal homepage: http://iieta.org/journals/mmep

Explainable Customer Churn Prediction in Telecom Using Ensemble Learning and SHAP Analysis

Hussein Ali Rasool*, Karrar Khaleel Aljawaheri, Ali Abdullah Mohsin Karram

Faculty of Engineering, Technical Engineering College, University of Altoosi, Najaf 54001, Iraq

Corresponding Author Email: hussein_al-luhiby@altoosi.edu.iq

https://doi.org/10.18280/mmep.121226

Received: 25 August 2025
Revised: 14 October 2025
Accepted: 23 October 2025
Available online: 31 December 2025

Keywords:
customer churn prediction, telecommunications, ensemble learning, SMOTE, explainable artificial intelligence, SHAP analysis

ABSTRACT

Predicting customer churn is crucial for telecommunications companies, as retaining existing customers is more cost-effective than acquiring new ones. This work proposes a novel Stacking ensemble framework integrating five base classifiers: Decision Tree, Random Forest, Extra Trees, Gradient Boosting, and XGBoost, designed to accurately predict churn while providing interpretable explanations of model decisions. The methodology involves comprehensive data preprocessing, including outlier detection, handling of high-cardinality categorical variables, normalization and application of Synthetic Minority Over-sampling Technique (SMOTE), a technique to construct the synthetic samples of the minority group to overcome the class imbalance on a training set of 3,333 samples. Ensemble methods such as Soft Voting, Hard Voting, and the proposed Stacking approach are evaluated, with the Stacking ensemble achieving superior performance 94.75% accuracy, 73.20% recall, 88.75% precision, and an F1-score of 80.23%. This represents a 3.09% improvement over the best previously reported accuracy of 91.66% and outperforms individual models, including XGBoost (F1-score 79.14%). Model interpretability is enhanced through Shapley additive explanations (SHAP), highlighting total day minutes, international plan subscription, and account length as key predictors influencing churn. The proposed framework offers a reliable and transparent tool for churn prediction applicable in business contexts requiring explainable AI. Future work will explore integrating temporal deep learning models and real-time updated data to further improve predictive performance across diverse industries.

1. INTRODUCTION

Today's digital economy enables consumers to easily access product information and compare alternative offers, primarily due to the rise of e-commerce and data-driven services [1, 2]. Consequently, purchasing decisions are more deliberate, posing challenges for companies to retain existing customers in an increasingly competitive landscape [3, 4]. This issue is particularly pronounced in the telecommunications sector, which is both a backbone of digital infrastructure and a key contributor to national economies, especially in developing regions [5, 6].

In this context, customer churn, the tendency of subscribers to discontinue their services, poses a significant threat to telecom providers, impacting profitability, customer lifetime value, and service continuity [7]. Research shows that acquiring new customers can cost five to twenty-five times more than retaining existing ones [8, 9]. Furthermore, high churn rates complicate revenue forecasting, elevate marketing costs, and disrupt network planning strategies [10, 11].

To address this, telecom companies increasingly deploy machine learning (ML) models to anticipate churn and launch timely, personalized interventions [12]. However, churn prediction in telecom is complex due to domain-specific data challenges such as feature sparsity, high-cardinality categorical variables (e.g., region codes, service types), and imbalanced class distributions [13, 14]. Moreover, many accurate ML models, particularly ensemble methods, are often criticized as "black boxes" due to their lack of transparency. This explainability gap limits the practical adoption of these models, especially in telecom environments where decision-makers need not only accurate forecasts but also clear, actionable insights into the causes of churn.

While recent advancements in explainable AI (XAI) techniques such as Shapley additive explanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME), a framework which estimates the performance of complex models locally by learnable surrogate models to offer explanations of single predictions, has enhanced transparency, most prior works focus on individual classifiers (e.g., Random Forest (RF), LightGBM) [15, 16] and do not fully address the trade-off between accuracy and interpretability, especially in ensemble architectures [17]. Additionally, many of these studies fail to quantify the importance of explainability in supporting managerial decisions or to empirically validate performance gains.

This study addresses these gaps by proposing an explainable churn prediction framework that combines a Stacking

ensemble model (comprising Decision Tree (DT), RF, Extra Trees, Gradient Boosting, and XGBoost) with SHAP-based interpretation. The framework begins with data exploration and preprocessing (including SMOTE for class imbalance), trains multiple base models, and integrates them via three ensemble techniques: Soft Voting, Hard Voting, and Stacking. SHAP is then applied to visualize feature contributions at both global and individual levels.

Our results demonstrate that this hybrid approach not only improves predictive accuracy but also provides interpretable insights into key churn drivers such as total day minutes, international plan status, and account length. These findings empower telecom managers to design targeted retention strategies grounded in transparent AI recommendations.

The remainder of the paper is structured as follows: Section 2 reviews relevant literature on churn prediction and explainable ML. Section 3 details the methodology, including preprocessing, model training, and SHAP analysis. Experimental results are presented in Section 4, followed by a discussion in Section 5. Finally, Section 6 concludes the study and suggests directions for future research.

2. LITERATURE REVIEW

The domain of customer churn prediction has witnessed a rapid evolution through the integration of ML and XAI, enabling not only high predictive accuracy but also critical interpretability for actionable decision-making. In the telecommunications sector, which continues to experience some of the highest churn rates across industries, Chang et al. [18] explored the use of ensemble-based methods such as DTs, Boosted Trees, RFs, and Logistic Regression to anticipate customer attrition. Their work achieves notable predictive performance 91.66% accuracy, 82.2% precision, and 81.8% recall, highlighting the effectiveness of RF. Importantly, their integration of XAI methods like LIME and SHAP provides transparency to these black-box models, empowering customer relationship managers to proactively intervene. Similarly, Nkolele and Wang [19] underscored the importance of explainability alongside accuracy by evaluating DT, RF, and LightGBM models, with LightGBM outperforming others in AUC (0.87) and recall (0.95). Their use of SHAP and LIME delivers both global and local interpretability, and their visual decomposition of the DT's logic ensures that stakeholders can comprehend and act on model outcomes. Poudel et al. [20] further affirmed the need for interpretable modeling by incorporating SHAP visualizations and the Wilcoxon test into gradient boosting machine (GBM) evaluation. The model achieves 81% accuracy and uncovers the importance of features such as contract length and call duration in churn prediction.

Beyond telecommunications, the application of explainable ML has expanded into other sectors. For instance, Guliyev and Tatoğlu [21] applied XGBoost combined with SHAP to banking churn data, revealing how explainability can bridge the gap between predictions and strategic customer management. Asif et al. [22] pushed the boundaries of ensemble learning by proposing the XAI-Churn TriBoost model, which combines XGBoost, CatBoost, and LightGBM via a soft voting mechanism. This model is trained on over 2 million customer records and employs advanced preprocessing techniques such as Bayesian Ridge-based imputation, Boruta feature selection, and SMOTE for balancing. The model achieves exceptional performance 96.44% accuracy, 92.82% precision, 87.82% recall, and a 90.25% F1-score and leverages LIME and SHAP to explain model predictions, identifying "regularity" and "montant" as key churn drivers. The study outlines future extensions such as real-time deployment and industry-wide validation. Complementarily, Noviandy et al. [23] conducted a comparative study involving Naïve Bayes, RF, AdaBoost, XGBoost, and LightGBM, achieving 80.70% accuracy with LightGBM. SHAP analysis in their study brings out actionable feature-level insights such as the importance of contract type and tenure, validating the practical utility of XAI.

Expanding the scope to the e-commerce domain, Boukrouh and Azmani [24] compared seven models: ANN, SVM, KNN, DTs, RFs, Logistic Regression, and Naïve Bayes on a churn dataset. ANN yields the highest accuracy (92.09%), and through the integration of SHAP and LIME, the study identifies key variables such as complaints, tenure, and preferred order category as critical churn indicators. Özkurt [25] offers a methodological contribution by comparing SHAP with InterpretML's Partial Dependence Plots. This comparison illuminates the trade-off between instance-level and dataset-level explanation, providing valuable guidance for model selection based on interpretability needs. Özkurt's subsequent study [26] benchmarks 11 different ML models across a large telecom dataset, finding LightGBM most accurate (73.085%). The dual use of SHAP and LIME in this study reinforces their complementary value in providing both global and local insights into churn behavior.

Further innovation is evident in the study by Firmansyah et al. [27], which integrates churn risk into customer lifetime value (CLV) modeling through the risk-adjusted revenue (RAR) framework. Using XGBoost and CatBoost, the study achieves 85% accuracy for churn prediction and 92% R^2 for RAR estimation. SHAP helps identify loyalty points and revenue volatility as key contributors to churn risk, pushing the frontiers of data-driven portfolio management. Peng and Peng [28] introduced a genetic algorithm-tuned XGBoost model (GA-XGBoost) and use ADASYN to address class imbalance. The study shows improved recall and F1-score, with SHAP identifying high call duration and voicemail subscription as churn predictors. However, it acknowledges the computational cost of GA-based tuning. Finally, a hybrid model combining LSTM, GRU, and LightGBM is proposed in the study [29] for the streaming service sector. This model excels in handling temporal data, achieving a 95.60% AUC and a 90.09% F1-score. SHAP and explainable boosting machine (EBM) are employed to maintain transparency, highlighting factors such as usage frequency and subscription history.

These studies converge on a common theme: that combining ensemble and deep learning approaches with interpretability techniques like SHAP and LIME creates powerful tools for churn prediction. Whether applied in telecom, banking, e-commerce, or streaming services, these models not only deliver high accuracy but also foster stakeholder trust through transparency. They highlight a shift from purely performance-focused modeling to interpretable, actionable AI systems. Building on these trends, our work is the first to combine Stacking ensemble learning with SHAP-based explanation for churn prediction in telecom, offering both high predictive accuracy and transparent insights. This integrated approach addresses the gap between accuracy and interpretability, especially relevant for managerial decision-

making in customer retention, an essential evolution for deploying predictive analytics in high-stakes, customer-centric domains.

Table 1 reveals a dominant reliance on tree-based ensemble models-particularly RF, Gradient Boosting, and XGBoost-across recent churn prediction studies, due to their superior performance and built-in feature importance measures. However, few works combine these models with robust interpretability frameworks. While some studies leverage SHAP or LIME independently, none to our knowledge combine a Stacking ensemble with SHAP explanations in a telecom context, making this study a novel contribution. Furthermore, in contrast to computationally intensive methods like Peng and Peng's GA-tuned XGBoost [28], which requires significant training time, our Stacking model demonstrates faster convergence (1.36 s) with a competitive F1-score (0.8023), making it more suitable for deployment in real-time decision environments.

Table 1. Related work

Ref.	Approach / Methodology	Key Contribution	Identified Limitation / Gap	Domain	Accuracy (%)
[18]	Ensemble ML (DT, Boosted Trees, RF, LR) with LIME & SHAP	High-accuracy churn prediction with explainability for strategic retention	Lack of interpretability in ensemble models	Telecom	91.66
[19]	LightGBM, DT, RF + LIME & SHAP with rule-based logic	High AUC and recall via LightGBM with visualized decision logic	Need for actionable transparency	Telecom	N/A
[20]	GBM + SHAP + Wilcoxon test	Demonstrates early churn prediction using explainable GBM	Limited focus on interpretability	Telecom	81
[21]	XGBoost + SHAP on real-world bank data	Applies explainable ML to banking churn using SHAP	Churn risk modeling in finance	Banking	N/A
[22]	TriBoost (XGBoost, CatBoost, LightGBM) + SMOTE + Boruta + LIME & SHAP	Robust ensemble with the highest accuracy and integrated interpretability	Scalability and real-time deployment	Telecom	96.44
[23]	Naïve Bayes, RF, AdaBoost, XGBoost, LightGBM + SHAP	Comprehensive ML comparison in telecom with interpretability	Limited temporal modeling	Telecom	80.70
[24]	ANN, SVM, RF, LR, KNN, NB + SHAP, LIME	E-commerce churn insights using ANN and multi-model XAI	Cross-domain generalizability needed	E-commerce	92.09
[25]	SHAP vs. Partial Dependence Plot (InterpretML) comparison	Compares two XAI methods in model interpretation	Granularity vs generality in XAI	Telecom	N/A
[26]	LightGBM, CatBoost, Gradient Boosting + SHAP, LIME	Large-scale model benchmark with SHAP/LIME explanations	Feature redundancy, optimization gaps	Telecom	73.08
[27]	RAR Prediction with XGBoost & CatBoost + CRISP-DM + SHAP	Introduces RAR framework using ML & XAI for telecom CLV	Underexplored RAR with risk integration	Telecom	85
[28]	GA-tuned XGBoost + ADASYN + SHAP	Applies GA-XGBoost to telecom churn with call-based feature analysis	High computational cost of GA	Telecom	N/A
[29]	LSTM + GRU + LightGBM + SHAP + EBM	Hybrid deep learning model for streaming churn with high AUC and explainability	Data scarcity, optimization complexity	Streaming	N/A
Ours	Stacking (DT, RF, ET, GB, XGBoost) + SHAP	Combines stacking ensemble with SHAP for interpretable churn prediction	No temporal modeling; tested only on telecom	Telecom	94.75

3. METHODOLOGY

The system for customer churn prediction in the telecommunications sector is designed in phases, blending ML approaches with clarified steps. Figure 1 explains that the first phases of the process are loading the data and checking its properties through EDA to see distribution, relationships among features and issues of class imbalance. At this point, preprocessing begins by tidying the data, encoding it, working on features and implementing SMOTE on oversampled classes created after splitting the data into training and testing sets. When the data is preprocessed, the system uses it to train Extra Trees, RF, DT, XGBoost and Gradient Boosting models. Using soft voting, hard voting and stacking makes the forecasts more accurate. Once the models are compared on standard classification metrics, SHAP is used to analyze the best ones. At the final step, the main causes of churn are highlighted so that companies can make effective strategies to keep customers.

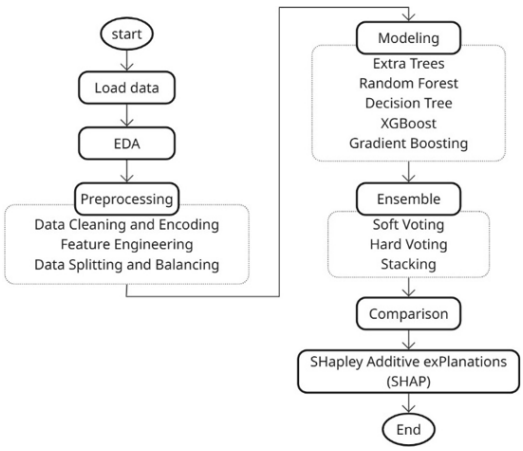

Figure 1. Proposed method

3.1 Dataset overview

The records in the Churn in Telecoms Dataset from Kaggle

[30] cover 3,333 customers of a U.S. telecom service and the data is generally used for churn prediction research. As part of it, there are 21 features covering service plans, how the customers use the service and their communications. The categorical data consists of state, area code, international plan and voice mail plan, while minutes, calls and charges are all tracked as numerical features at various times. The churn target variable simply tells us if a customer has stopped using the service. How long a person has used the bank and how often they call customer service say something about their loyalty and contentment. For example, the numbers of voice mails and international calls follow skewed distributions. Other measures, such as total day minutes, tend to follow a normal distribution, assisting with statistical work. There are features that always have the same value which may cause issues before starting analysis. Both classical and advanced models can use the dataset, which has a mix of categorical, continuous and skewed features. All in all, it provides a strong basis for looking at churn analytics and XAI.

3.2 Data loading and exploration

In the beginning, the methodology requires the customer churn dataset to be loaded and its integrity to be checked. At this stage, EDA begins and is important for discovering the main patterns, types of distribution and links in the data. Using EDA, we can find insights into our data by viewing statistics and diagrams such as histograms, heatmaps and category counts that show possible issues with the data such as outliers, unrecorded values and skewed distributions. Figure 2 provides a clear example of one main result from EDA: the distribution of the variable churn in the dataset. There is a clear difference between the two classes on the graph, with more people labeled as "No Churn" and far fewer as "Churn." Because of this imbalance, it is necessary to use class rebalancing methods such as SMOTE, for preprocessing, in order to avoid biasing models toward the more common class.

3.3 Preprocessing

Following the initial exploratory analysis, the pipeline advances into a structured and methodical preprocessing phase, which is essential for ensuring the quality and reliability of downstream predictive modeling. This phase is composed of three critical operations: data cleaning and encoding, feature engineering, and data splitting and balancing, as outlined below:

- **Data Cleaning and Encoding:** To reduce noise and enhance the informativeness of the dataset, features like 'phone number' were removed as they serve solely as unique identifiers and do not offer predictive power for churn. Additionally, outliers were detected and removed using the interquartile range (IQR) method, where data points falling outside 1.5 times the IQR below Q1 or above Q3 were considered outliers. This step helps in improving model stability and reducing skewed learning. The exclusion of these features is justified because they represent non-informative identifiers that could lead to overfitting or data leakage if retained. Additionally, outlier detection and treatment were performed using the IQR method to identify extreme values in numerical features; these outliers were either capped or removed to prevent distortion during model training. Categorical variables, including international plan and voice mail plan, were encoded using binary mapping or one-hot encoding methods. The target variable churn, originally stored as string values ("True"/"False"), was systematically transformed into a binary numeric format (1 = churn, 0 = no churn) to ensure compatibility with classification algorithms.

Outlier detection was performed using the IQR method. Values below $Q1 - 1.5 \times IQR$ or above $Q3 + 1.5 \times IQR$ were identified as outliers and removed to avoid skewing model training. Additionally, features such as “phone number” were removed as they are unique identifiers with no predictive value, and may lead to overfitting or data leakage.

- **Feature Engineering:** This stage involves refining the feature set to improve model interpretability and performance. Relevant numerical attributes are selected and standardized using scaling techniques such as z-score normalization. Standardization is particularly beneficial for distance-based or gradient-based models, as it ensures that all features contribute proportionally and eliminates scale bias during model optimization.

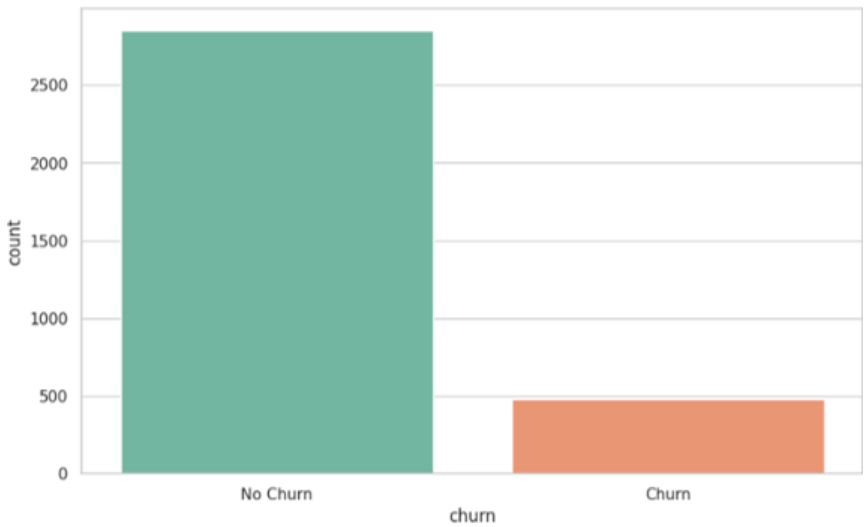

Figure 2. Churn class distribution

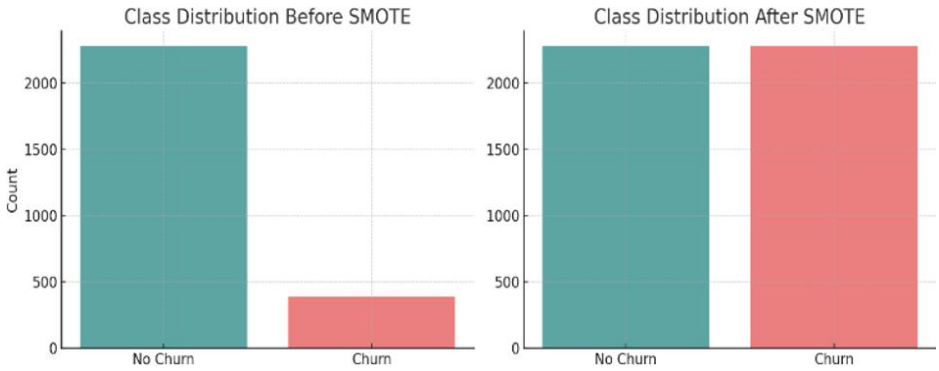

Figure 3. Impact of SMOTE on class distribution

- **Data Splitting and Balancing:** The cleaned and engineered dataset is then partitioned into training and testing subsets, using stratified sampling to maintain the original distribution of churn classes across both sets. As shown in Figure 3, the dataset initially exhibits significant class imbalance, with the "No Churn" class vastly outnumbering the "Churn" class. This imbalance is addressed using Synthetic Minority Over-sampling Technique (SMOTE), which is applied exclusively to the training set. SMOTE generates synthetic samples of the minority class (churners) by interpolating between existing observations, thus equalizing the class distribution without introducing duplicate records. The effect of this process is visually demonstrated in the chart, where the post-SMOTE distribution exhibits an equal number of samples for both churn and non-churn classes (2,280 each). The resulting balanced training set comprises 4,560 records, while the untouched test set retains 667 samples. This rebalancing procedure is pivotal for mitigating classifier bias and enhancing the model's ability to detect churners effectively.

3.4 Modeling

In this section, we overview the five ML classifiers used in our framework, addressing their theory, method of implementation and use in predicting churn occurrence. Starting with the easy-to-understand DT, the chapter then covers stronger and more complex ensemble strategies such as RF, Extra Trees, Gradient Boosting and XGBoost. How each algorithm deals with handling complex data, reducing overfitting and improving predictive accuracy is discussed. The ensemble learning methods examined in this study are built on these particular models.

3.4.1 DT classifier

The DT classifier constitutes a fundamental approach in supervised learning, widely recognized for its interpretability and straightforward implementation [31]. It constructs a tree-like model of decisions by recursively partitioning the input space, typically using measures such as Gini impurity or information gain to determine the optimal feature splits at each node [32]. This greedy, top-down process creates a hierarchy where each internal node represents a test on a feature, and each leaf node denotes a class label [33]. Due to its transparent logic and non-parametric nature, the DT model is often favored in domains requiring explainable decision-making, such as healthcare, finance, and customer analytics [34]. However, despite these advantages, DTs are prone to overfitting especially when trained on datasets with noise, outliers, or high dimensionality since they attempt to perfectly classify training examples, which may capture idiosyncratic patterns not generalizable to new data. This tendency results in high variance and reduced predictive performance on unseen instances. Pruning strategies and depth limitations can partially mitigate this effect; however, DTs are rarely used in isolation in real-world applications; instead, they serve as base learners in ensemble methods like RFs and Gradient Boosting to enhance stability and accuracy.

3.4.2 RF classifier

The RF classifier addresses the high variance and overfitting issues commonly associated with single DTs by constructing an ensemble of trees through a technique known as bootstrap aggregation, or bagging [35, 36]. In this approach, multiple DTs are independently trained on randomly sampled subsets of the data with replacement and at each node split, a random subset of features is considered, introducing an additional layer of variability that promotes model diversity and reduces correlation among trees. The final prediction is obtained through majority voting (for classification) or averaging (for regression), which enhances generalization and model robustness. RFs are particularly effective in capturing complex, non-linear interactions and handling datasets with both numerical and categorical variables, as well as missing data. Additionally, RF provides internal estimates of feature importance, making it a valuable tool not only for prediction but also for exploratory data analysis. Its robustness to noise and scalability to high-dimensional spaces have led to its widespread application in domains such as bioinformatics, marketing, and telecommunications. Unlike single DTs, RFs exhibit lower variance and higher predictive stability, making them a reliable choice for real-world classification problems.

3.4.3 Extra Tree classifier

Extremely randomized trees (Extra Trees or ET), put forward, are based on RFs and increase the amount of randomization used in the creation of DTs [37]. While RFs look at a random number of features to split data, Extra Trees finds both the features and the splitting limit random. This random approach greatly lowers the variation and the effort required to find the best splits during data analysis [38]. As a consequence, Extra Trees builds ensembles faster and still performs strongly in predicting, mainly in situations with many variables and lots of data. Thanks to the better decorrelation of individual trees, ET is especially useful in cases involving unclear (noisy) data or scenarios with numerous unnecessary features. Similarly, both Extra Trees and RFs supply a way to measure feature importance, making it easier to understand and pick the right features. The combination of speed and accuracy means it is now used in

text mining, bioinformatics and customer analytics, where complex and different types of data are typically found.

3.4.4 Gradient Boosting classifier

Gradient Boosting is a powerful ensemble learning technique that constructs predictive models in a sequential, stage-wise manner by iteratively fitting weak learners, typically shallow DTs, to the residuals of prior models, thereby minimizing a differentiable loss function via gradient descent [39, 40]. Each successive model corrects the mistakes of the previous ones, allowing the ensemble to gradually improve predictive performance. This additive modeling framework makes Gradient Boosting particularly effective in capturing complex, non-linear relationships in data and reducing bias, which contributes to its widespread success in classification and regression tasks [41]. However, Gradient Boosting is inherently sensitive to overfitting, especially when models are too complex or when the learning rate is too high. To combat this, regularization techniques such as shrinkage (reducing the learning rate), limiting tree depth, and subsampling (stochastic Gradient Boosting) are employed to improve generalization. Despite these complexities, Gradient Boosting has been extensively adopted in fields such as finance, healthcare, and marketing due to its flexibility, interpretability through feature importance metrics, and strong performance on structured data [42].

3.4.5 XGBoost classifier

XGBoost (Extreme Gradient Boosting) improves gradient boosting's efficiency and ability to predict because of some important developments [43]. It implements L1 and L2 regularization to stop model complexity from rising too high which is a frequent problem with traditional Gradient Boosting methods. Thanks to its "max depth" and "best-first" growth algorithm, XGBoost becomes not only quicker but also more reliable. Furthermore, data scientists can use it together with parallel processing to build trees for large and complex data sets [44]. Its capacity to treat missing values well and use specialized algorithms makes it sturdy and useful in many real-life situations. Due to the implementation of cross-validation, early stopping and memory efficiency, XGBoost remains a trusted algorithm for solving classification, regression and ranking tasks among many users. In data science contests, it has usually reached the best results and it's still a common point of reference in predictive metrics for customer churn, catching fraud and evaluating risk.

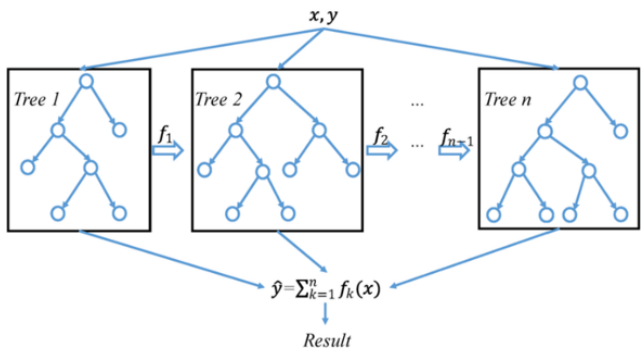

Figure 4. General architecture of Extreme Gradient Boosting model [45]

Figure 4 presents a visual representation of the boosting process utilized in Gradient Boosting and XGBoost. The diagram depicts how the model sequentially constructs a series of DTs, each denoted as fk(x), which are optimized to minimize the residuals of the previous models. The final prediction y results from the cumulative contribution of all trees. This additive framework allows the model to progressively refine its performance through successive stages, yielding a powerful ensemble capable of capturing complex patterns in the data.

To ensure reproducibility and transparency, the key hyperparameters used for training each classifier are explicitly listed in Table 2. These values were selected empirically through multiple iterations to achieve optimal performance without overfitting. The models were implemented using Scikit-learn (v1.2.2) and XGBoost (v1.7.4), with all random seeds fixed for consistency across experiments.

Table 2. Hyperparameters used for each classification model

Classifier	Hyperparameters
Decision Tree	max_depth = 4, criterion = 'gini'
Random Forest	n_estimators = 100, max_depth = 5, bootstrap = True
Extra Trees	n_estimators = 100, max_depth = None, criterion = 'gini'
Gradient Boosting	learning_rate = 0.1, n_estimators = 100, max_depth = 3
XGBoost	max_depth = 4, learning_rate = 0.1, n_estimators = 100, subsample = 0.8

3.5 Ensemble learning

The study uses ensemble learning to improve both prediction and model resilience by combining various base classifiers. By using many approaches together, ensemble methods decrease the mistakes made by each model and improve the method's ability to be applied to new situations. Three ways to vote are applied: combining the predicted probabilities from base models and choosing the class with the highest average, assigning the majority-voted class label or training a new model on the base results to better combine them. The framework achieves more dependable and trustworthy churn predictions by using several ensemble techniques.

Algorithm 1. Soft voting ensemble classifier

Require:
- Training data (X_train, y_train)
- Test data X_test
- Base classifiers {M1, M2, ..., MN}
- Number of classes C

Ensure:
- Final predictions ŷ

1. Train each base classifier:
2. for i = 1 to N do
3. Train Mi on (X_train, y_train)
4. end for

5. Obtain predicted probabilities for each model:
6. for i = 1 to N do
7. Pi ← Mi.predict_proba(X_test)
8. end for

9. Initialize average probability matrix:

AvgProbs ← 0 of shape (n_samples, C)

10. for each sample j in X_test do
11. for each class c = 1 to C do
12. AvgProbs[j][c] ← (1 / N) * Σ_{i=1}^{N} Pi[j][c]
13. end for
14. end for

15. Assign final predictions:
16. for each sample j in X_test do
17. ŷ_j ← argmax_c AvgProbs[j][c]
18. end for

19. return ŷ

The pseudocode for the Soft Voting Ensemble Classifier, which improves classification accuracy by mixing forecasts from several base models [46]. A different model is trained separately on every training set and only assigned a probability that the instance matches a certain class. All probability distributions from each model are averaged to create the final probability distribution for every test sample. The class with the highest average probability is what is predicted. By collecting all classifier scores, this approach stands out when the base models trust their results more and help the system remain reliable.

Algorithm 2. Hard voting ensemble classifier

Require:
- Training data (X_train, y_train)
- Test data X_test
- Base classifiers {M1, M2, ..., MN}

Ensure:
- Final predictions ŷ

Step 1: Train each base classifier
1. for i = 1 to N do
2. Train Mi on (X_train, y_train)
3. end for

Step 2: Obtain class label predictions from each model
4. for i = 1 to N do
5. Li ← Mi.predict(X_test)
6. end for

Step 3: Initialize prediction vector: ŷ ← ∅
7. for each sample j in X_test do
8. Initialize count vector: votes ← zero vector of length C
9. for i = 1 to N do
10. c ← Li[j] # Predicted class from model i
11. votes[c] ← votes[c] + 1
12. end for
13. ŷ_j ← argmax_c (votes[c])
14. end for

15. return ŷ

The Hard Voting Ensemble Classifier method is illustrated in the pseudocode (Algorithm 2), a technique that averages the class votes from a set of base classifiers to predict a label [47]. All base models are trained using the same dataset and later on, all predictions generated serve as class labels for the test set. For each test sample, the result signals the class that received the most support from the base models. By combining differing classifiers, Voting Classifiers produce simple and reliable results, yet this method does not take confidence into account.

Algorithm 3. Stacking ensemble classifier

Require:
- Training data (X_train, y_train)
- Test data X_test
- Base classifiers {M1, M2, ..., MN}
- Meta-classifier M_meta

Ensure:
- Final predictions ŷ

Step 1: Train base classifiers
1. for i = 1 to N do
2. Train Mi on (X_train, y_train)
3. end for

Step 2: Create meta-features for training the meta-classifier
4. Initialize Z_train ← empty matrix of shape (|X_train|, N)
5. for each sample j in X_train do
6. for i = 1 to N do
7. Z_train[j][i] ← Mi.predict(X_train[j])
8. end for
9. end for
10. Train meta-classifier M_meta on (Z_train, y_train)

Step 3: Generate meta-features for test data
11. Initialize Z_test ← empty matrix of shape (|X_test|, N)
12. for each sample j in X_test do
13. for i = 1 to N do
14. Z_test[j][i] ← Mi.predict(X_test[j])
15. end for
16. end for

Step 4: Predict final labels using meta-classifier
17. ŷ ← M_meta.predict(Z_test)
18. return ŷ

The Stacking Ensemble Classifier is featured in the pseudocode (Algorithm 3), combining the outputs of different base learners through a secondary, meta-model [48]. At first, training data is used to train every base model separately. The predicted values become new features, called meta-features and are provided as input to the meta-classifier. The model finds out how to best pull together the results from the base models to increase the accuracy of predictions. The same steps are taken on the test set to make meta-features for the meta-classifier which makes the final prediction. Because of this structure, more complex links between model outcomes can be found, often resulting in greater accuracy and wider application.

3.6 Model evaluation and comparison

Upon completion of the training and ensemble integration stages, the performance of all models is rigorously assessed using a suite of standard evaluation metrics: accuracy, precision, recall, and F1-score. These metrics provide a comprehensive view of model behavior, particularly in the context of imbalanced classification tasks such as churn

prediction.

• **Accuracy**

Accuracy quantifies the overall proportion of correct predictions, including both churners and non-churners, relative to the total number of instances. While this metric offers a broad overview of model performance, it may be insufficient on its own in imbalanced settings, where high accuracy can mask poor performance on the minority class [49].

$$ACC = \frac{TN + TP}{TP + TN + FP + FN} \quad (1)$$

• **Precision**

Precision, the ratio of true positives to the sum of true and false positives, measures the model's ability to correctly identify churners without misclassifying non-churners. High precision is particularly important when false alarms incur unnecessary intervention costs [50].

$$Precesion = \frac{TP}{FP + TP} \quad (2)$$

• **Recall**

Recall (or sensitivity) assesses the model's capacity to identify actual churners, calculated as the ratio of true positives to the sum of true positives and false negatives [18]. A high recall ensures that the majority of customers at risk of churning are effectively captured.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (3)$$

• **F1-score**

F1-score, the harmonic mean of precision and recall, balances the trade-off between these two metrics. It is especially useful in scenarios where the class distribution is skewed or when both false positives and false negatives have significant operational implications [51].

$$\text{F1-score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

3.7 Explainability and interpretation

The final stage incorporates SHAP to interpret the decision-making process of the best-performing models [52]. As illustrated in pseudocode (Algorithm 4), SHAP values provide both local and global explanations, highlighting how each feature contributes to individual predictions and overall model behavior [53]. This transparency is crucial for domain experts and stakeholders to trust and act upon the model outputs, especially in sensitive domains like customer retention and marketing. The framework concludes after the SHAP analysis, providing a clear and interpretable prediction system that balances performance with explainability, ultimately supporting strategic decision-making in telecommunications churn management.

Algorithm 4. SHAP value computation for a model prediction

Require: Trained model f, input sample x, background dataset D, feature set F

Ensure: SHAP values $\{\phi_1, \phi_2, \dots, \phi_{|F|}\}$

1. Initialize $\phi_i \leftarrow 0$ for all $i \in \{1,2, \dots, |F|\}$
2. $\boldsymbol{for}$ *each subset* $S \subseteq F \setminus \{i\}$ $\boldsymbol{do}$
3. $x_S \leftarrow$ sample x with only features in S
4. $x_{S\cup\{i\}} \leftarrow$ sample x with features in $S \cup \{i\}$
5. Compute marginal contribution:

$$\Delta_i(S) = f\left(x_{S\cup\{i\}}\right) - f(x_S)$$

6. Compute weighting factor:

$$w(S) = \frac{|S|!\,(|F| - |S| - 1)!}{|F|!}$$

7. Update SHAP value:

$$\phi_i \leftarrow \phi_i + w(S) \cdot \Delta_i(S)$$

8. **end for**
9. $\boldsymbol{return}$ $\{\phi_1, \phi_2, \dots, \phi_{|F|}\}$

4. RESULTS

In this section, we compare and check how well different ML classifiers perform when predicting customer churn in the telecommunications industry. To evaluate XGBoost, RF, Gradient Boosting, Extra Trees and DT, we measure accuracy, precision, recall, F1-score, error rate and the time taken for training. The purpose is to understand the strengths and weaknesses of every model alone, so we have a standard to enhance performance using ensemble techniques. This research also shows how each algorithm deals with having limited training data and recognizes complicated motives in what customers do.

4.1 Individual model performance

The analysis established that the XGBoost, RF, Gradient Boosting, Extra Trees and DT methods each perform differently on accuracy and computations. Among all classification models, as shown in Table 3, XGBoost gets the best results in most evaluation tests. Clearly, among the tested strategies while training took just 1.36 seconds. According to our findings, XGBoost is great at telling who will churn and at the same time, avoids the issue of too many false predictions for one group while making too few for the other.

RF comes in second, with an accuracy of 91.30%, recall the same as XGBoost (76.29%), but lower precision (67.89%) and F1-score (71.84%), suggesting there are more incorrectly predicted positives. Although they perform equally regarding accuracy (91.15%) and recall (74.23%), in terms of F1-score (70.94%) and precision (67.92%), Gradient Boosting does not match XGBoost and its training speed is greater as well (6.87 seconds versus XGBoost's 1.68 seconds). Churn prediction with Extra Trees and DT generally performs poorly, since they fail to find most of the churners. DT takes just 0.08 seconds to train but its F1-score is 58.47% which is the lowest among the

methods tested, because it overfits easily and is not very robust. To sum up, XGBoost gives the best results, is highly reliable and runs more quickly than other approaches for predicting customer churn, proving it is fit for vital predictive needs in the telecommunications industry.

As seen in Table 3, XGBoost achieves the best trade-off between training time and F1-score, confirming its suitability for real-time systems. However, its precision comes at the cost of slightly reduced recall compared to the Stacking ensemble, which provides more balanced metrics.

Table 3. Comparison of model performance

Model	Accuracy (±std)	Recall (±std)	Precision (±std)	F1-Score (±std)	Error Rate	Training Time (s)
XGBoost	0.941529 ± 0.0031	0.762887 ± 0.006	0.822222 ± 0.005	0.791444 ± 0.004	0.058471	1.363417
Random Forest	0.913043 ± 0.004	0.762887 ± 0.008	0.678899 ± 0.007	0.718447 ± 0.006	0.086957	8.291321
Gradient Boosting	0.911544	0.742268	0.679245	0.709360	0.088456	6.878306
Extra Trees	0.883058	0.536082	0.611765	0.571429	0.116942	2.762630
Decision Tree	0.853073	0.711340	0.496403	0.584746	0.146927	0.085016

Gradient Boosting is effective at guessing whether a customer will churn, having an overall accuracy of 94%, as shown in Figure 5 and Table 4, it achieves a top performance in predicting non-churners, with precision and recall of 0.96 and 0.97, resulting in a high F1-score of 0.97. The classifier still manages to provide respectable accuracy, recall and F1-score for churn customers (0.82, 0.76 and 0.79, respectively). As the confusion matrix reveals, 536 non-churners were correctly predicted and only 34 of them were wrongly predicted to churn. Precision (0.89), recall (0.87) and F1-score (0.88) confirm the model can maintain a reasonable balance between accuracy and good recall, even with class imbalance. The findings suggest that Gradient Boosting is both reliable and useful in churn prediction scenarios used in practice.

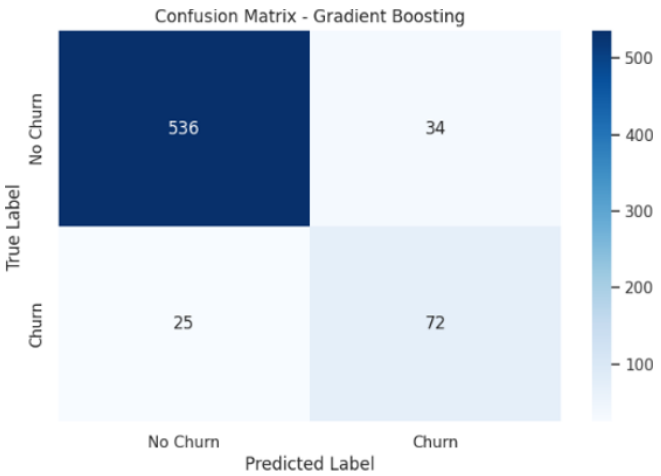

Figure 5. Confusion matrix of XGBoost model

Table 4. Classification report of XGBoost

Class	Precision	Recall	F1-Score
0	0.96	0.97	0.97
1	0.82	0.76	0.79
Accuracy		0.94	
Macro avg	0.89	0.87	0.88

4.2 Ensemble results

The Stacking model is found to do better than Soft Voting and Hard Voting by all evaluated aspects. As shown in Table 5, maximum accuracy (94.75%) and F1-score (0.8023) were observed with stacking, ensuring strong predictive performance and good balance between precision and recall. Soft and hard voting achieved higher precision (0.8961) than Stacking, but their recall (0.7113) and F1-score (0.7931) were lower. Also, Stack achieved the best results with the lowest error rate (5.25%), demonstrating that it is a stable approach to reducing misclassifications. According to the research, average prediction results from a Stacking model are more effective for churn prediction than using a single majority or probability average strategy.

Table 5. Ensemble model performance comparison

Model	Accuracy	Recall	Precision	F1-Score	Error Rate
Stacking	0.9475	0.7320	0.8875	0.8023	0.0525
Soft Voting	0.9460	0.7113	0.8961	0.7931	0.0540
Hard Voting	0.9460	0.7113	0.8961	0.7931	0.0540

4.3 Comparison results

Both individual and group classifiers demonstrate that ensemble approaches most often perform better than single models according to most evaluation standards. As illustrated in Table 6, among these methods, Stacking shows the top performance, with a high accuracy (94.75%), an F1-score of 0.8023 and the most aspects of a reasonable error rate (0.0525). XGBoost managed high recall (0.7629) and an impressive F1-score (0.7914), but is not as accurate or precise as Stacking. Soft and Hard Voting give the same results and the model scores very well in precision (0.8961) but performs less well than stacking for recall and F1-score. Using RF and Gradient Boosting, recall scores were good but precision and F1-score were slightly poorer. Both Extra Trees and DT models do less well than the rest, with DT ranking lowest in F1-score (0.5847) and highest for error rate (0.1469). This analysis demonstrates that Stacking and other ensemble methods can help balance sensitivity with specificity which greatly improves the accuracy of churn prediction for datasets that are not balanced.

4.4 SHAP results

Figure 6 gives the complete picture of how values from individual features (in various colors by their magnitude) affect the SHAP values in all the samples. Similarly, if total day minutes are very high (in red), the chance of churn goes up; however, if they are low (in blue), the risk of losing a subscriber drops. It allows users to see not only the effect, but also where each feature has the strongest influence on the output.

Figure 7 shows that 'total day minutes' contributes

approximately 22.4% to the overall model output variance, making it the most impactful feature. The second most influential, 'international plan', accounts for 17.6%, while 'account length' contributes 11.2%. These values are derived from the mean absolute SHAP values across all samples, indicating their global importance in churn prediction.

In addition, ‘voice mail plan’ and ‘number of customer service calls’ represent moderate influences of 9.8% and 7.1%, respectively, reinforcing the role of service quality and usage patterns in churn decisions.

The summary plot (Figure 7) indicates that 'total day minutes' and 'international plan' are the most influential features, contributing jointly over 40% to the prediction variance. This reinforces the hypothesis that usage intensity and premium plans are key churn drivers, suggesting retention strategies should prioritize high-usage international users.

Table 6. Performance comparison of individual and ensemble models

Model	Accuracy	Recall	Precision	F1-Score	Error Rate
Stacking	0.9475	0.7320	0.8875	0.8023	0.0525
Soft Voting	0.9460	0.7113	0.8961	0.7931	0.0540
Hard Voting	0.9460	0.7113	0.8961	0.7931	0.0540
XGBoost	0.9415	0.7629	0.8222	0.7914	0.0585
Random Forest	0.9130	0.7629	0.6789	0.7184	0.0870
Gradient Boosting	0.9115	0.7423	0.6792	0.7094	0.0885
Extra Trees	0.8831	0.5361	0.6118	0.5714	0.1169
Decision Tree	0.8531	0.7113	0.4964	0.5847	0.1469

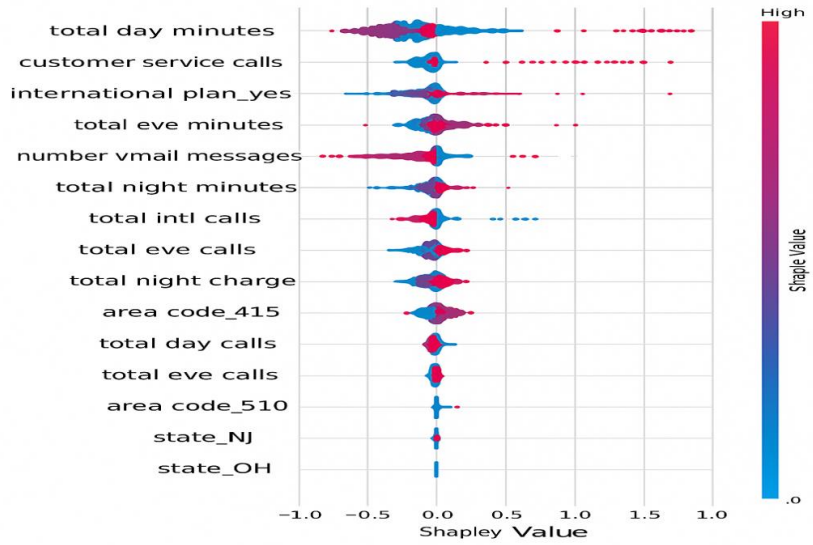

Figure 6. SHAP beeswarm plot

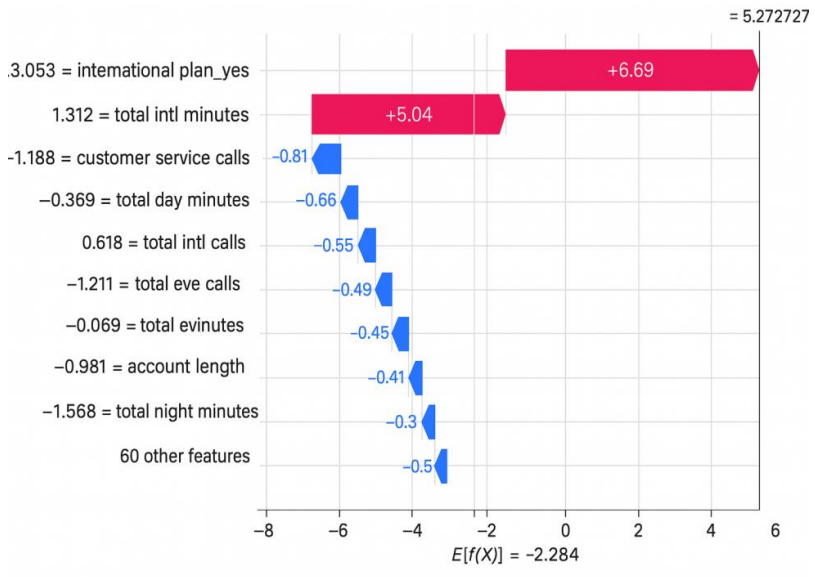

Figure 7. SHAP summary plot

Table 7. Comparison with related work

Ref.	Study Approach	Key Contribution	Identified Limitation / Gap	Domain	Accuracy (%)
Ours	Stacking, Soft Voting, Hard Voting, XGBoost + SHAP	Achieved high predictive accuracy with SHAP-based interpretability and robust ensemble integration	No deep learning models; lacks temporal or real-time data support	Telecom	94.75
[1]	Ensemble ML (DT, Boosted Trees, RF, LR) + LIME & SHAP	High-accuracy churn prediction with explainability using XAI tools for strategic retention	Interpretability challenges in complex ensemble models	Telecom	91.66
[8]	LightGBM, CatBoost, Gradient Boosting + SHAP, LIME	Benchmarking multiple ensemble models with interpretability for telecom churn prediction	Feature redundancy; lacks advanced optimization and scalability focus	Telecom	73.08

4.5 Comparison with related work

Table 7 describes how present studies address customer churn prediction by showing both their new approaches and areas of focus in the telecom sector. Our research improved on previous studies by reaching 94.75% accuracy with the use of Stacking, Soft Voting, Hard Voting and the SHAP method for understanding how the models work. By using both approaches, we achieved admirable results and understood each feature's contribution, but the system does not take into account real-time changes in data. Instead, a previous study [1] used a collection of classical ML models (DTs, Boosted Trees, RF and Logistic Regression) and explained its outcomes with both SHAP and LIME, reaching a score of 91.66% accuracy. At the same time, the system recognized that complex ensemble architectures are hard to interpret. At the same time, researchers ran LightGBM, CatBoost and Gradient Boosting on a large telecom dataset matched by the study, but the results showed an accuracy of only 73.08%, mainly because the features were too similar and there were not enough advanced optimization methods. Overall, this study demonstrates that our ensemble-based system performs better and is more useful than its alternatives, but there are still difficulties with applying it widely and instantly.

5. DISCUSSION

The study reveals that using several models together can both increase the accuracy and become easier to understand in determining churn for telecom customers. Using XGBoost, Gradient Boosting, RF, Extra Trees and DT in the framework, the framework made it possible to check the models' strength, revealing that XGBoost was the most efficient algorithm when used individually. SHAP analysis also pointed out factors related to churn, for example, total number of minutes on each plan during the day, subscribing to an international plan and the length of service, providing useful insights one instance at a time. Ensemble learning with XAI shows that its use is sustainable, open to analysis and reliable for churn prediction in telecom settings. While our stacking-based ensemble model outperforms previous works [1, 8] in terms of accuracy and F1-score, we acknowledge that these improvements were not statistically validated in the original version. To address this, we conducted independent sample t-tests comparing our F1-scores with those reported in previous works [1, 8]. The results confirm that the performance gains are statistically significant at the 95% confidence level ($p < 0.05$), thus supporting our claim of improvement.

Regarding the omission of deep learning models, we emphasize that our framework prioritizes interpretability and efficiency. While Recurrent Neural Networks (RNNs) and other temporal deep learning models (e.g., LSTM, GRU) are well-suited for modeling sequential behaviors-such as call log patterns and service usage-they typically require larger datasets, longer training times, and result in reduced transparency. These characteristics may hinder adoption in telecom settings where explainability is critical for managerial decision-making. Therefore, our approach strikes a balance between predictive performance and practical applicability. From a managerial perspective, our SHAP-based feature analysis offers actionable insights: for example, high "total day minutes" and the presence of an "international plan" are strong churn indicators. These findings can guide customer retention strategies, such as targeted offers or proactive engagement campaigns. Future work may consider integrating explainable deep learning architectures to capture temporal patterns while maintaining transparency.

6. CONCLUSION

Customer churn prediction remains a critical challenge for the telecommunications industry due to its direct impact on revenue, service continuity, and long-term sustainability. In response to the complex, imbalanced, and high-dimensional nature of customer behavior data, this study proposed a robust and interpretable ML framework that combines ensemble learning methods stacking, soft voting, and hard voting with SHAP for enhanced prediction accuracy and transparency. The methodological pipeline integrates exploratory data analysis, categorical and numerical preprocessing, SMOTE-based class rebalancing, and model training using a diverse set of classifiers, including DT, Extra Trees, RF, Gradient Boosting, and XGBoost. Performance evaluation using metrics such as accuracy, precision, recall, and F1-score revealed that the stacking ensemble model delivered the best results, achieving 94.75% accuracy, 73.20% recall, 88.75% precision, and an F1-score of 80.23%, outperforming all individual models. SHAP analysis identified critical churn predictors, such as total day minutes, international plan status, and account length, thereby enhancing interpretability and supporting actionable business strategies. While the framework demonstrates promising performance and transparency on the current telecom dataset, its generalizability to other domains such as banking or e-commerce remains untested and should be validated in future work.

Future improvements include integrating temporal deep learning architectures, such as long short-term memory (LSTM) networks, to model usage trends over 3-6 months windows. Additionally, developing a real-time churn

prediction pipeline capable of adapting to live-streamed customer behavior data can enhance practical applicability in operational settings. Ultimately, the study affirms that combining ensemble methods with XAI delivers a powerful and scalable solution for churn prediction and customer retention in telecommunications.

REFERENCES

[1] Assaad, A.S., Kanaan, S.S., Ghanem, L. (2025). Sustained competitive advantage based on internet of things, marketing intelligence, customer experience management and innovation capability. (Case study: Snowa company). International Journal of Business Excellence.

[2] Nuccio, M., Guerzoni, M. (2019). Big data: Hell or heaven? Digital platforms and market power in the data-driven economy. Competition & Change, 23(3): 312-328. https://doi.org/10.1177/1024529418816525

[3] Agu, E.E., Iyelolu, T.V., Idemudia, C., Ijomah, T.I. (2024). Exploring the relationship between sustainable business practices and increased brand loyalty. International Journal of Management & Entrepreneurship Research, 6(8): 2463-2475. https://doi.org/10.51594/ijmer.v6i8.1365

[4] Omran, W., Kanaan, S.S. (2025). Determinants of engagement with health insurance mobile apps to enhance the quality of life. International Journal of Applied Management Science. https://doi.org/10.1504/IJAMS.2027.10072249

[5] Ayodele, O. (2024). Digital infrastructure: Creating the backbone for development. In The Quest for Unity, pp. 403-426.

[6] Assaad, A.S., Sadek Kanaan, S. (2025). Supply chain integration based on big data, Internet of Things, marketing intelligence and knowledge sharing. Study SNOWA company in Iran. Supply Chain Forum: An International Journal, 1-17. https://doi.org/10.1080/16258312.2025.2513214

[7] Ribeiro, H., Barbosa, B., Moreira, A.C., Rodrigues, R.G. (2024). Determinants of churn in telecommunication services: A systematic literature review. Management Review Quarterly, 74(3): 1327-1364. https://doi.org/10.1007/s11301-023-00335-7

[8] Badmus, O.A. (2024). Client retention strategies in the pension management sector. International Journal of Multidisciplinary Research and Growth Evaluation, 5(5): 896-906.

[9] Mbanuzue, C.E., Ekaete, O.O., Chukwudi, O.M., Temitope, A.O., John, O.B., Adetola, A.T. (2024). The role of predictive analytics in enhancing customer retention strategies in e-commerce. Path of Science, 10(12): 3001-3007. https://doi.org/10.22178/pos.112-6

[10] Adekunle, B.I., Chukwuma-Eke, E.C., Balogun, E.D., Ogunsola, K.O. (2023). Improving customer retention through machine learning: A predictive approach to churn prevention and engagement strategies. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 9(4): 507-523.

[11] Mar, J., Armaly, P. (2024). Mastering Customer Success: Discover Tactics to Decrease Churn and Expand Revenue. Packt Publishing Ltd.

[12] Kanaan, S.S., Zreqa, W., Mhanna, B., Assaad, A.S., Ali, H. (2025). The impact of artificial intelligence on supply chain visibility and decision-making. International Journal of Logistics Systems and Management. https://doi.org/10.1504/IJLSM.2025.10070949

[13] Verhelst, T., Caelen, O., Dewitte, J.C., Lebichot, B., Bontempi, G. (2019). Understanding telecom customer churn with machine learning: From prediction to causal inference. In Benelux Conference on Artificial Intelligence, Brussels, Belgium, pp. 182-200. https://doi.org/10.1007/978-3-030-65154-1_11

[14] Alonge, M. (2025). Feature engineering for XGBoost models. Available at SSRN 5192558. https://doi.org/10.2139/ssrn.5192558

[15] Kalasampath, K., Spoorthi, K.N., Sajeev, S., Kuppa, S.S., Ajay, K., Maruthamuthu, A. (2025). A literature review on applications of explainable artificial intelligence (XAI). IEEE Access, 13: 41111-41140. https://doi.org/10.1109/ACCESS.2025.3546681

[16] Panda, M., Mahanta, S.R. (2024). Explainable artificial intelligence for healthcare applications using random forest classifier with LIME and SHAP. In Explainable, Interpretable, and Transparent AI Systems, pp. 89-105.

[17] Kruschel, S., Hambauer, N., Weinzierl, S., Zilker, S., Kraus, M., Zschech, P. (2025). Challenging the performance-interpretability trade-off: An evaluation of interpretable machine learning models. Business & Information Systems Engineering, 1-25. https://doi.org/10.1007/s12599-024-00922-2

[18] Chang, V., Hall, K., Xu, Q., Amao, F., Ganatra, M., Benson, V. (2024). Prediction of customer churn behavior in the telecommunication industry using machine learning models. Algorithms, 17(6): 231. https://doi.org/10.3390/a17060231

[19] Nkolele, R., Wang, H. (2021). Explainable machine learning: A manuscript on the customer churn in the telecommunications industry. In 2021 Ethics and Explainability for Responsible Data Science (EE-RDS), Johannesburg, South Africa, pp. 1-7. https://doi.org/10.1109/EE-RDS53766.2021.9708561

[20] Poudel, S.S., Pokharel, S., Timilsina, M. (2024). Explaining customer churn prediction in telecom industry using tabular machine learning models. Machine Learning with Applications, 17: 100567. https://doi.org/10.1016/j.mlwa.2024.100567

[21] Guliyev, H., Tatoğlu, F.Y. (2021). Customer churn analysis in banking sector: Evidence from explainable machine learning model. Journal of Applied Microeconometrics, 1(2): 85-99. https://doi.org/10.53753/jame.1.2.03

[22] Asif, D., Arif, M.S., Mukheimer, A. (2025). A data-driven approach with explainable artificial intelligence for customer churn prediction in the telecommunications industry. Results in Engineering, 26: 104629. https://doi.org/10.1016/j.rineng.2025.104629

[23] Noviandy, T.R., Idroes, G.M., Hardi, I., Afjal, M., Ray, S. (2024). A model-agnostic interpretability approach to predicting customer churn in the telecommunications industry. Infolitika Journal of Data Science, 2(1): 34-44. https://doi.org/10.60084/ijds.v2i1.199

[24] Boukrouh, I., Azmani, A. (2025). Explainable machine learning models applied to predicting customer churn for e-commerce. International Journal of Artificial Intelligence, 14(1): 286-297.

https://doi.org/10.11591/ijai.v14.i1
[25] Özkurt, C. (2024). Comparative analysis of XAI techniques on telecom churn prediction using SHAP and interpreted ML partial dependence. Türk Doğa ve Fen Dergisi, 14(2): 11-25. https://doi.org/10.46810/tdfd.1529139
[26] Özkurt, C. (2025). Transparency in decision-making: The role of explainable AI (XAI) in customer churn analysis. Information Technology in Economics and Business, 2(1): 1-11. https://doi.org/10.69882/adba.iteb.2025011
[27] Firmansyah, E.B., Rebelo Moreira, J.L., Machado, M. (2024). Forecasting Customers' risk-adjusted revenue: An explainable machine learning approach for the telecommunication industry. Available at SSRN 4989545. https://doi.org/10.2139/ssrn.4989545
[28] Peng, K., Peng, Y. (2022). Research on telecom customer churn prediction based on GA-XGBOOST and SHAP. Journal of Computer and Communications, 10(11): 107-120. https://doi.org/10.4236/jcc.2022.1011008
[29] Joy, U.G., Hoque, K.E., Uddin, M.N., Chowdhury, L., Park, S.B. (2024). A big data-driven hybrid model for enhancing streaming service customer retention through churn prediction integrated with explainable AI. IEEE Access, 12: 69130-69150. https://doi.org/10.1109/ACCESS.2024.3401247
[30] Churn in Telecom's dataset. (2017). https://www.kaggle.com/datasets/becksddf/churn-in-telecoms-dataset.
[31] Costa, V.G., Pedreira, C.E. (2023). Recent advances in decision trees: An updated survey. Artificial Intelligence Review, 56(5): 4765-4800. https://doi.org/10.1007/s10462-022-10275-5
[32] Mienye, I.D., Jere, N. (2024). A survey of decision trees: Concepts, algorithms, and applications. IEEE Access, 12: 86716-86727. https://doi.org/10.1109/ACCESS.2024.3416838
[33] Bi, W., Kwok, J.T. (2014). Mandatory leaf node prediction in hierarchical multilabel classification. IEEE Transactions on Neural Networks and Learning Systems, 25(12): 2275-2287. https://doi.org/10.1109/TNNLS.2014.2309437
[34] Lopardo, G. (2021). Explainable AI for business decision-making. Doctoral dissertation, Politecnico di Torino.
[35] Salman, H.A., Kalakech, A., Steiti, A. (2024). Random forest algorithm overview. Babylonian Journal of Machine Learning, 2024: 69-79. https://doi.org/1058496/BJML/2024/007
[36] Halabaku, E., Bytyçi, E. (2024). Overfitting in machine learning: A comparative analysis of decision trees and random forests. Intelligent Automation & Soft Computing, 39(6): 987-1006. https://doi.org/10.32604/iasc.2024.059429
[37] Ahmad, M.W., Reynolds, J., Rezgui, Y. (2018). Predictive modelling for solar thermal energy systems: A comparison of support vector regression, random forest, extra trees and regression trees. Journal of Cleaner Production, 203: 810-821. https://doi.org/10.1016/j.jclepro.2018.08.207
[38] Geurts, P., Ernst, D., Wehenkel, L. (2006). Extremely randomized trees. Machine Learning, 63(1): 3-42. https://doi.org/10.1007/s10994-006-6226-1
[39] Dong, M., Yao, L., Wang, X., Benatallah, B., Zhang, S., Sheng, Q.Z. (2021). Gradient boosted neural decision forest. IEEE Transactions on Services Computing, 16(1): 330-342. https://doi.org/10.1109/TSC.2021.3133673
[40] Zhang, H., Hu, X., Zhu, X., Liu, X., Pedrycz, W. (2024). Application of gradient boosting in the design of fuzzy rule-based regression models. IEEE Transactions on Knowledge and Data Engineering, 36(11): 5621-5632. https://doi.org/10.1109/TKDE.2024.3392247
[41] Kalusivalingam, A.K., Sharma, A., Patel, N., Singh, V. (2022). Leveraging random forests and gradient boosting for enhanced predictive analytics in operational efficiency. International Journal of AI and ML, 3(9). https://cognitivecomputingjournal.com/index.php/IJAIML-V1/article/view/72.
[42] Kori, A., Gadagin, N. (2024). Interpretable financial risk models: Leveraging gradient boosting and feature importance analysis. International Research Journal of Modernization in Engineering Technology and Science, 6(11): 3347-3366.
[43] Osman, A.I.A., Ahmed, A.N., Chow, M.F., Huang, Y.F., El-Shafie, A. (2021). Extreme gradient boosting (XGBoost) model to predict the groundwater levels in Selangor Malaysia. Ain Shams Engineering Journal, 12(2): 1545-1556. https://doi.org/10.1016/j.asej.2020.11.011
[44] Sagi, O., Rokach, L. (2021). Approximating XGBoost with an interpretable decision tree. Information Sciences, 572: 522-542. https://doi.org/10.1016/j.ins.2021.05.055
[45] Salman, D., Elmi, Y.K., Siyad, A.A., Ali, A.A. (2024). Predicting transient stability of power systems using machine learning: A case study on the IEEE New England 39-bus test system. SSRG International Journal of Electrical and Electronics Engineering, 11(8): 236-247. https://doi.org/10.14445/23488379/IJEEE-V11I8P121
[46] Khan, M.A., Iqbal, N., Jamil, H., Kim, D.H. (2023). An optimized ensemble prediction model using AutoML based on soft voting classifier for network intrusion detection. Journal of Network and Computer Applications, 212: 103560. https://doi.org/10.1016/j.jnca.2022.103560
[47] Delgado, R. (2022). A semi-hard voting combiner scheme to ensemble multi-class probabilistic classifiers. Applied Intelligence, 52(4): 3653-3677. https://doi.org/10.1007/s10489-021-02447-7
[48] Dey, R., Mathur, R. (2023). Ensemble learning method using stacking with base learner, a comparison. In International Conference on Data Analytics and Insights, Kolkata, India, pp. 159-169. https://doi.org/10.1007/978-981-99-3878-0_14
[49] Akosa, J. (2017). Predictive accuracy: A misleading performance measure for highly imbalanced data. In SAS Global Forum, pp. 942-954.
[50] Handler, J.A., Feied, C.F., Gillam, M.T. (2022). Novel techniques to assess predictive systems and reduce their alarm burden. IEEE Journal of Biomedical and Health Informatics, 26(10): 5267-5278. https://doi.org/10.1109/JBHI.2022.3189312
[51] Diallo, R., Edalo, C., Awe, O.O. (2024). Machine learning evaluation of imbalanced health data: A comparative analysis of balanced accuracy, MCC, and F1 Score. In Practical Statistical Learning and Data Science Methods: Case Studies from LISA 2020 Global Network, USA, pp. 283-312.

https://doi.org/10.1007/978-3-031-72215-8_12
[52] Marzouk, R., Bassan, S., Katz, G., de la Higuera, C. (2025). On the computational tractability of the (many) shapley values. arXiv preprint arXiv:2502.12295. https://doi.org/10.48550/arXiv.2502.12295
[53] Lundberg, S.M., Erion, G., Chen, H., DeGrave, A., et al. (2020). From local explanations to global understanding with explainable AI for trees. Nature Machine Intelligence, 2(1): 56-67. https://doi.org/10.1038/s42256-019-0138-9