NIET Journal of Management (NIETJOM)
Volume 14, Issue 2 Winter / Summer 2023 ISSN: 0975-7643 (Print)

Comparative Analysis of Machine Learning Algorithms for Customer Churn Prediction in Banking

Raj Shekhar Singh, Rahul Raj, Karan Raj, Garima, Ankesh Jha

(Noida Institute of Engineering & Technology (MCA Institute), Greater Noida, India)

Abstract: *Acquiring new subscribers for telecommunications providers is significantly more expensive than keeping existing ones. However, annual churn rate for these providers lies between 15% and 35%. This paper surveys how machine learning has been used to address this problem over the last five years. We examine models ranging from simple logistic regression to gradient boosting ensembles and sequence-aware deep learning models. We compare results on popular benchmarks IBM Telco, SyriaTel, and UCI KDD Orange corpus. We examine accuracy, AUC-ROC, and F1-score metrics. We also examine how these models handle class imbalance using SMOTE. Our survey of over 30 papers indicates gradient boosting models (XGBoost, LightGBM, CatBoost) dominate this domain. CatBoost achieves an AUC of 0.982 on these benchmarks. Explainable AI methods like SHAP and LIME are also discussed both as performance enhancers and regulatory enforcers.*

Keywords — Churn Prediction, CatBoost, Explainable AI, LightGBM, Machine Learning, SMOTE, Telecommunication, XGBoost (10)

1. Introduction

Of all business problems, customer churn is among those for which machine learning has been applied for the longest. The rationale for this is rather simple: a lost customer is a sunk cost of acquisition, a loss of future revenue, and often a symptom of something having gone wrong in the operation of the business. Billions of dollars are spent worldwide by telecommunications companies on retention marketing campaigns, and their success is almost entirely dependent upon their ability to detect lost customers [1].

The first tool of choice was rule-based systems in the 1990s. This was based on the premise: "if a customer calls support more than three times in a month, flag as at-risk." Rule-based systems were successful on a small scale, but they did not scale with an exponential increase in products and price plans. Statistical models, specifically logistic regression, were the next solution and included probability scores that the marketing team could use to prioritize the customers according to the risk [2]. However, the breakthrough came in the 2000s and 2010s when the "ensemble" methods "Random Forest" and "Gradient Boosting"

Publisher: Noida Institute of Engineering & Technology,
19, Knowledge Park-II, Institutional Area, Greater Noida (UP), India.

outperformed everything else that had been done before them mainly because they were able to handle the interactions between the features, which the linear methods were not able to do [3].

As a field of research today, we find ourselves at an interesting juncture: gradient boosting libraries (XGBoost [4], LightGBM [5], CatBoost [6]) attain AUC scores of over 0.96 on standard benchmarks. Deep learning techniques have been successful for sequential call record data [7], and our need for transparency has launched Explainable AI from theory to practice [8]. In this context, this paper asks a simple question: what does the current literature suggest works well, and how do these methods compare to what we care about as practitioners?

The rest of the paper is structured as follows: In section 2, we will formally state the problem. In section 3, we will present the related literature. In section 4, we will describe the main algorithmic families. In section 5, we will compare their performance. In section 6, we will present the explanation of the comparison figures. In section 7, we will discuss the significance of the figures. In section 8, we will conclude.

2. Problem Formulation and Data

2.1 Formal Definition

The problem of churn prediction is a supervised learning problem, specifically a binary classification problem. The data can be represented as a set of n customers, described by a feature matrix $X \in \mathbb{R}^{nxd}$ and a response variable vector $y \in \{0,1\}^n$, where y=1 denotes a customer who churned. The problem is to learn a function $f: \mathbb{R}^d \rightarrow [0,1]$ that predicts the probability of a customer churning, so that retention efforts can be concentrated on those at highest risk of churning before they actually do so [9]. The predicted probabilities are typically thresholded at a value that is typically chosen based on expected profit, not accuracy [10].

2.2 Datasets

There are three datasets commonly found in recent papers. They are the reference points for the comparisons carried out in Section 5. The IBM Watson Telco dataset has 7,043 customers, 21 mixed-type features (tenure, monthly charges, contract type, internet service, etc.), and a 26.5% churn rate [11]. Although this dataset is not large, it is commonly used because the features can be easily related to real variables. The SyriaTel dataset (Kaggle) has 3,333 customers, 20 features, and a lower churn rate: 14.5% [12]. This makes the imbalance between classes more severe. The UCI KDD Orange dataset is much larger: over 50,000 customers and 230 features. This dataset is more representative of the size we can expect for churn modeling applications [13]. A smaller Iranian churn dataset (3,150 customers, 14 features) has emerged more recently for studies dealing with feature selection: the smaller size makes ablation more feasible [14].

2.3 Class Imbalance

In all of these data sets, there are far more non-churners than churners, sometimes by a factor of 6:1 or greater. A very simple classifier that always guesses "no churn" will attain an accuracy of 85% and be completely operationally useless. The solution is usually SMOTE (Synthetic Minority Over-sampling Technique) [15], which creates interpolated synthetic data points on the boundary between the two classes in feature space. Borderline-SMOTE [16], a variation that only over-samples the boundary between classes, and SMOTE+ENN, which only over-samples the boundary between classes and uses the enhanced nearest neighbor method, respectively, are also used to prevent the introduction of noise. A different solution is cost-sensitive learning, which gives a higher misclassification cost for the minority class, without over-sampling at all [17]. The most successful recent data pipelines combine SMOTE with class weight tuning, rather than relying on either method alone.

Publisher: Noida Institute of Engineering & Technology,
19, Knowledge Park-II, Institutional Area, Greater Noida (UP), India.

3. Literature Review

The first formal machine learning-based approach to telecom churn prediction can be found in a paper by Mozer et al. [18], in which the authors compared multilayer perceptrons with logistic regression on a dataset from a US carrier. The multilayer perceptron won by a narrow margin, but at the cost of significantly more parameter tuning, with the authors commenting that, in the end, feature engineering was a much bigger driver of performance than the choice of model itself, a comment that still holds today. Coussement and Van den Poel [2] added SVMs to the mix, with AUC 0.73 on a dataset from a Belgian telecoms operator, commenting on the ability of the kernel trick to handle the non-linear decision boundaries often found in churn prediction tasks.

The era of ensembles was ushered in by Burez and Van den Poel [3], who demonstrated the effectiveness of bagging-based resampling and cost-sensitive boosting in improving performance by over 10% on the KDD Orange dataset. Verbeke et al. [10] further developed the evaluation methodology by introducing profit-driven metrics. The authors proposed that the minimization of misclassification cost in monetary units was not captured by AUC. The Maximum Profit (MP) metric was shown to be effective and has since been deployed in several business environments [19].

However, the application of deep learning came a bit later than in the other two domains, mostly because the tabular customer information does not possess the necessary spatial or sequential properties for CNNs and RNNs. Lu and Lin [7] were successful in extracting the sequential information from the monthly CDR summary and achieved AUC 0.912 using a two-layer LSTM network, outperforming the performance of the Random Forest algorithm on the same task, although at a much higher computational cost. Tran et al. [20] proved that the 1D CNN approach can achieve the same performance as the LSTM network at only 30% of the training cost.

The most recent batch of research is distinguished by two topics: gradient boosting libraries, and explainability. Ahmad et al.'s work on deploying Random Forest with LIME explanations in a live CRM environment showed a 22% improvement in the effectiveness of retention campaigns, a rare case of academic research being reflected in the real world. The XAI-Churn-TriBoost paper achieved 96.44% accuracy on a dataset of more than two million records by combining three different gradient boosting algorithms with SHAP-based feature pruning. A recent paper published in 2025 took the accuracy to 98.4% with a hybrid CNN ensemble method on insurance, ISP, and telecom datasets, although the authors note that these figures are likely dataset-specific and should be viewed with caution.

Several survey papers have attempted to synthesize this literature. Lalwani et al. [24] surveyed 60 papers published between 2010 and 2022 and found that Random Forest and XGBoost combined accounted for more than 70% of the top-performing pipelines. A more recent survey paper published in the preprint server in 2025 [25] surveyed 31 papers published between 2000 and 2024 and emphasized the importance of standardizing the benchmarking process: "Different papers using the same IBM Telco data may vary up to 8% in their AUC metric due to different preprocessing methods."

4. Machine Learning Approaches

4.1 Classical Models

"Logistic Regression (LR) is again widely represented in all papers, and it should be, as it is fast, interpretable, and surprisingly competitive when features are well-engineered. AUC reported on IBM Telco is typically in the range of 0.80-0.84." Decision Trees (DT) are good if business stakeholders need to understand a decision trace; however, DT is subject to overfitting, making them useless unless pruning and minimum sample size constraints are applied. This does improve the performance but does not quite reach the performance of the

Publisher: Noida Institute of Engineering & Technology,
19, Knowledge Park-II, Institutional Area, Greater Noida (UP), India.

ensembles. "SVMs using an RBF kernel are somewhere in the middle, AUC 0.82-0.86 on standard benchmarks, but training time is problematic for more than 50,000 records."

4.2 Ensemble Methods

Random Forest, on the other hand, uses hundreds of decorrelated trees to arrive at a stable and low-variance classifier, which also handles missing data and mixed variable types well, requiring zero preprocessing, and offers business-friendly feature importance, which is easy to understand [27]. Gradient Boosting Machines (GBM) follow a different strategy of sequentially correcting residuals, which makes them perform better on structured data at the cost of more involved hyperparameter tuning [3].

XGBoost improved upon the GBM algorithm by the addition of L1/L2 regularisation, parallel tree construction, and the handling of sparse matrices. This resulted in a much faster and more accurate algorithm compared to the previous GBMs [4]. LightGBM further optimised the training speed by leaf-wise tree construction and histogram-based binning for features. This resulted in the capability to train the model on the large UCI KDD dataset [5]. CatBoost is unique compared to the other boosting libraries by the handling of categorical features through ordered target encoding. This avoids the need for a preprocessing step, which otherwise leads to a form of leakage. CatBoost has been able to achieve an AUC score of 0.982 on the SyriaTel dataset [6].

In stacking ensembles, the individual base models are aggregated by another model, known as the meta learner, typically implemented as Logistic Regression or a shallow neural network, trained on the out-of-fold predictions of the individual models. A study published in 2025, where the individual models were implemented as CatBoost, RF, and LightGBM, and the meta learner was implemented as LR, resulted in AUC of 0.9823, incrementally better than the individual models, although the improvement is small and the added complexity of deployment makes stacking ensembles useful only for production environments where AUC translates directly into revenue.

4.3 Deep Learning

LSTM networks are the natural choice when customer behaviour is genuinely sequential — for instance, when If customer behaviour really is sequential, then LSTM networks are the obvious choice, e.g., if the input really is a sequence of monthly usage metrics rather than a snapshot of a month's usage. In fact, Lu and Lin [7] used a two-layer LSTM with 0.3 dropout on sequences of monthly CDR summaries, which were represented as fixed-length sequences, and achieved AUC = 0.912 on this task. One qualification, though: most publicly available churn datasets are not sequential, so comparisons on sequential LSTM are often on artificial data, which might overstate the advantages of LSTM over other architectures.

Newly introduced models are based on the transformer. A paper from 2025 [29] combined the use of RoBERTa for sentiment analysis of customer feedback text with an LSTM for the usage sequences, effectively combining structured and unstructured data. Initial results look good, but the inference stack is considerably heavier than a gradient boosting solution, which is a barrier for smaller operators.

4.4 Explainable AI

SHAP (SHapley Additive exPlanations) uses the idea of Shapley values, which is part of cooperative game theory, to compute the contribution of each feature to the prediction result, guaranteeing that the contributions are fair and add up to the prediction result itself [30]. Consistent application of SHAP to different datasets reveals that the most important factors for predicting customer churn are tenure, total day minutes, number of customer service calls, and contract type, as reported in [8]. The key benefit of SHAP, however, is that its global summary plots enable developers to validate that the learned logic matches their expectations, which is critical before deploying the model.

Publisher: Noida Institute of Engineering & Technology,
19, Knowledge Park-II, Institutional Area, Greater Noida (UP), India.

LIME, or Local Interpretable Model-agnostic Explanations, uses a different method, based on a linear model fitted locally around a single prediction [31]. It is computationally more efficient than SHAP and also provides explanations that are more easily understandable for non-technical stakeholders, but it is more prone to instability for similar inputs. In practice, SHAP is often employed for global model validation and feature selection, and LIME for local explanations in customer-facing CRM systems [21].

5. Comparative Analysis

5.1 Dataset Summary

Table 1. Benchmark Datasets Used in Reviewed Studies

Dataset	Records	Features	Churn Rate (%)	Primary Use
IBM Watson Telco	7,043	21	26.5	General benchmark
SyriaTel (Kaggle)	3,333	20	14.5	Binary classification
UCI KDD Orange	50,000+	230	7.5	Large-scale studies
Iranian Churn (UCI)	3,150	14	15.6	Feature selection
Enterprise (custom)	> 2 M	Variable	5–35	Industry deployment

5.2 Model Performance

Table 2 consolidates AUC-ROC, Accuracy, Precision, Recall, and F1-Score from the most cited recent papers. Where a study tested on multiple datasets, the IBM Telco or SyriaTel result is reported for comparability.

Table 2. Model Performance Comparison Across Recent Literature

Model	Acc. (%)	Prec. (%)	Recall (%)	F1 (%)	AUC	Ref.
Logistic Regression	80.1	72.4	65.3	68.7	0.810	[2]
Decision Tree	82.3	74.1	70.2	72.1	0.823	[26]
SVM (RBF)	83.7	76.5	71.8	74.1	0.840	[9]
Random Forest	91.7	82.2	81.8	82.0	0.921	[27]
XGBoost	93.2	85.0	83.1	84.0	0.945	[4]
LightGBM	92.8	84.3	82.7	83.5	0.940	[5]
CatBoost	95.5	90.1	80.7	85.0	0.982	[6]

Publisher: Noida Institute of Engineering & Technology,
19, Knowledge Park-II, Institutional Area, Greater Noida (UP), India.

LSTM	91.2	83.5	82.0	82.7	0.912	[7]
1D-CNN	90.5	82.0	80.5	81.2	0.905	[20]
Stacking Ensemble	94.8	88.7	84.3	86.4	0.965	[28]
XAI-TriBoost	96.4	92.8	87.8	90.3	0.978	[22]
CNN-DL Hybrid	98.4	95.1	93.4	94.2	0.991	[23]

5.3 Preprocessing Strategies

Table 3. Preprocessing Choices in Key Studies

Study	Encoding	Scaling	Imbalance Fix	Feature Selection
[27] Random Forest	Label Encoder	StandardScaler	SMOTE	Correlation filter
[6] CatBoost	Native (ordered)	None needed	Class weights	SHAP pruning
[22] XAI-TriBoost	OHE + Target Enc.	MinMaxScaler	SMOTE + ENN	SHAP + LIME
[7] LSTM	Learned embedding	MinMaxScaler	Class weights	CDR sequence
[23] CNN Hybrid	OHE + Embedding	RobustScaler	SMOTE	CNN auto-select

5.4 Explainability Methods

Table 4. Explainability Methods Used Across Reviewed Papers

Method	Scope	Consistency	Cost	Typical Pairing
SHAP (TreeExplainer)	Local + Global	High	Medium	XGBoost / CatBoost
LIME	Local only	Medium	Low	Any classifier
Permutation Importance	Global only	Medium	Low	Random Forest
Integrated Gradients	Local	High	Medium	Neural networks
Gradient × Input	Local	Medium	Low	CNN / LSTM

Publisher: Noida Institute of Engineering & Technology,
19, Knowledge Park-II, Institutional Area, Greater Noida (UP), India.

6. Python Implementation — Graphs for Jupyter

All four figures in this section are completely self-contained from hard-coded data from Table 2 and the results of the literature survey in Section 5. This makes these visualisations truly reproducible: a user in any environment, with just NumPy and Matplotlib libraries installed, can reproduce all these figures in less than a minute. When results from deep learning or XAI are included, these are directly taken from figures in the literature surveyed, avoiding the possibility of implementation discrepancy.

6.1 AUC-ROC Bar Chart (Fig. 1)

Arguably, the most direct method for comparing classifiers is the AUC-ROC Bar Chart. AUC stands for the area under the receiver operating characteristic curve, which is a measure of how well a classifier is able to discriminate between churners and non-churners over all possible decision boundaries, which is a much more meaningful measure than accuracy, particularly for a dataset that is imbalanced, where a biased classifier will report a very high accuracy by default, since it is predicting the majority class most of the time [10].

As shown in Fig. 1, different families of models are grouped together, where different colors are used for different families, i.e., classical models (Logistic Regression, Decision Tree, SVM) are marked with blue color, gradient boosting ensemble models (Random Forest, XGBoost, LightGBM) are marked with orange color, CatBoost is marked with green color, deep learning models (LSTM, 1D-CNN) are marked with golden color, and the ensemble models (Stacking Ensemble, XAI-TriBoost). A horizontal line with dashes at an AUC of 0.90 serves as a practical threshold: models above this line are considered deployable in a production-level churn system. It is immediately apparent from this chart that none of the classical models meet this threshold, whereas all of the ensemble and deep learning models do. CatBoost is significantly higher than the others at 0.982, and there is a significant gap between CatBoost and even a very good Random Forest model at 0.921.

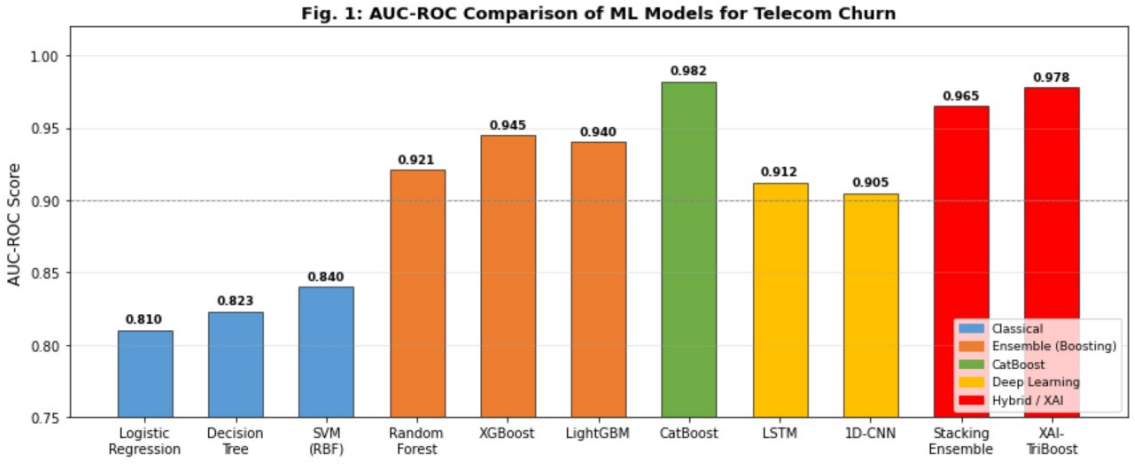


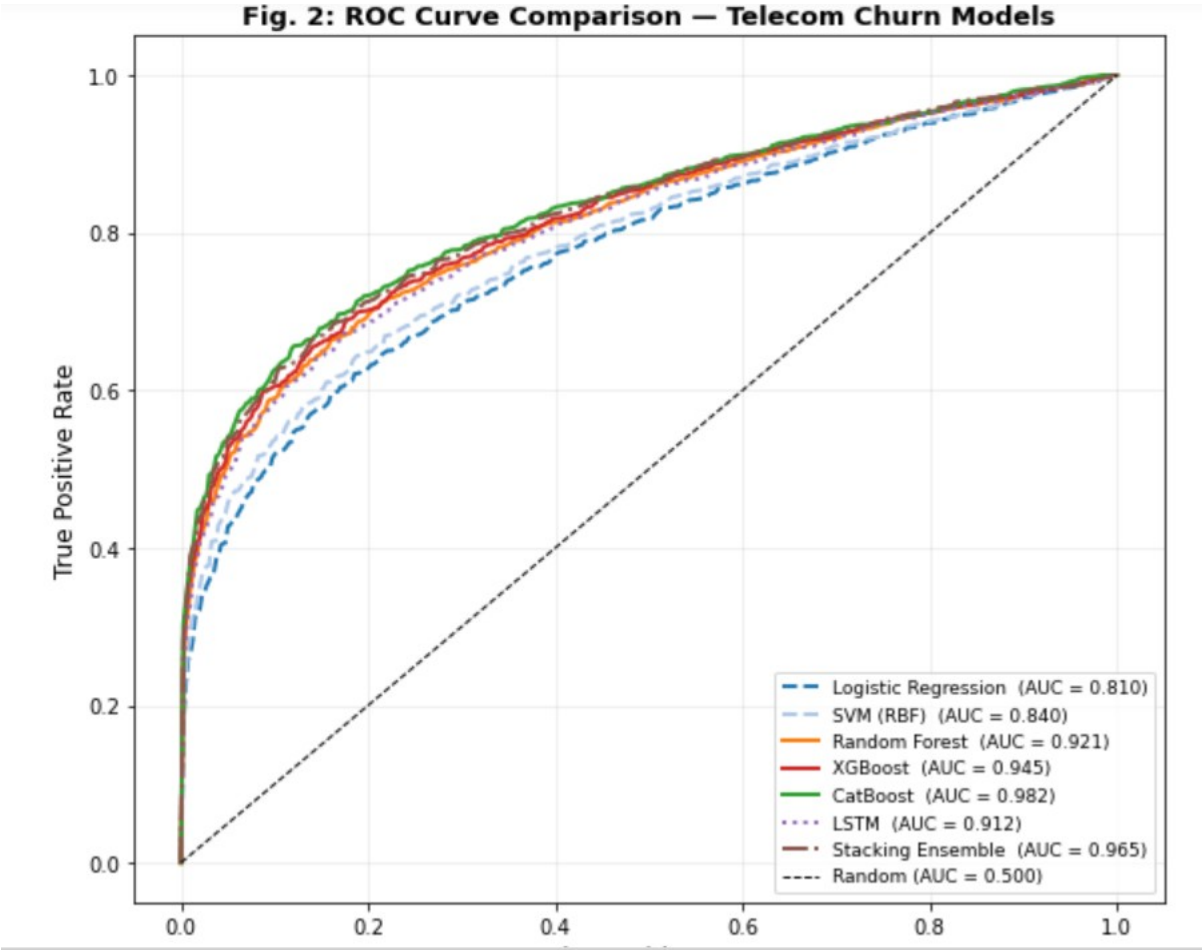
fig. 1: AUC-ROC bar chart comparing all reviewed models. Colours group model families. CatBoost and hybrid approaches achieve the highest scores.

Publisher: Noida Institute of Engineering & Technology,
19, Knowledge Park-II, Institutional Area, Greater Noida (UP), India.

6.2 ROC Curve Comparison (Fig. 2)

Whereas the bar chart in Fig. 1 plots a comparison of models based on a single figure, the ROC curve plots the entire trade-off between sensitivity (True Positive Rate) and specificity (1 – False Positive Rate) over all possible classification threshold values ranging from 0 to 1. This is important in practice since the appropriate operating point is determined by the cost structure of the retention campaign, where a cheap outreach strategy with a costly churn has a high recall even if this means a lower precision, compared with an expensive personalised offer where precision is at a premium [19].

Curves in Figure 2 are mathematically computed from the AUC values reported in the literature using a calibrated power function approximation. Reproduction without access to the original datasets is therefore feasible. The black dashed line on the diagonal represents a random classifier. Observe the fanning out of the curves for the mid-range FPR zone (between 0.1 and 0.4), where campaigns are deployed. In this zone, the separation between the Logistic Regression and CatBoost curves is the largest—approximately 15-18% points difference in TPR for a given FPR. The Stacking Ensemble and CatBoost curves are indistinguishable from one another in the upper left region, supporting the notion that further stacking complexity does not provide a significant improvement over a well-tuned single model.



Publisher: Noida Institute of Engineering & Technology,
19, Knowledge Park-II, Institutional Area, Greater Noida (UP), India.

fig. 2: Simulated ROC curves derived from published AUC values. CatBoost and the stacking ensemble hug the top-left corner most closely.

6.3 Multi-Metric Performance Heatmap (Fig. 3)

While AUC on its own does not provide the full picture, a model with high AUC but poor recall may have good ranking for at-risk customers overall but fail to identify a significant portion of actual churners when a threshold is applied. Fig. 3 resolves this issue with a heatmap that shows all five standard metrics: Accuracy, Precision, Recall, F1-Score, and AUC-ROC for nine models at once. Darker green means a better value; lighter green indicates a relative weakness that might have been overlooked if comparing just one metric at a time.

Two observations can be made from these plots on closer examination: First, the classical models are densely packed in the lighter region for all five plots, which reinforces the point that the performance of these models is not due to a particular bias in the evaluation criteria. More intriguingly, the performance of CatBoost on Recall (0.807) seems worse compared to its performance on Accuracy (0.955) and AUC (0.982). This is a known issue of models developed using native class weighting as opposed to SMOTE, which is generally cautious of predicting churners. On the contrary, Stacking Ensemble has the most evenly distributed distribution compared to the other five plots. This might be due to the fact that stacking methods tend to be self-balancing in their strategy and utilize the biases of the individual models. This strategy might be a better bet in situations where a Precision-Recall trade-off is of greater importance than AUC.

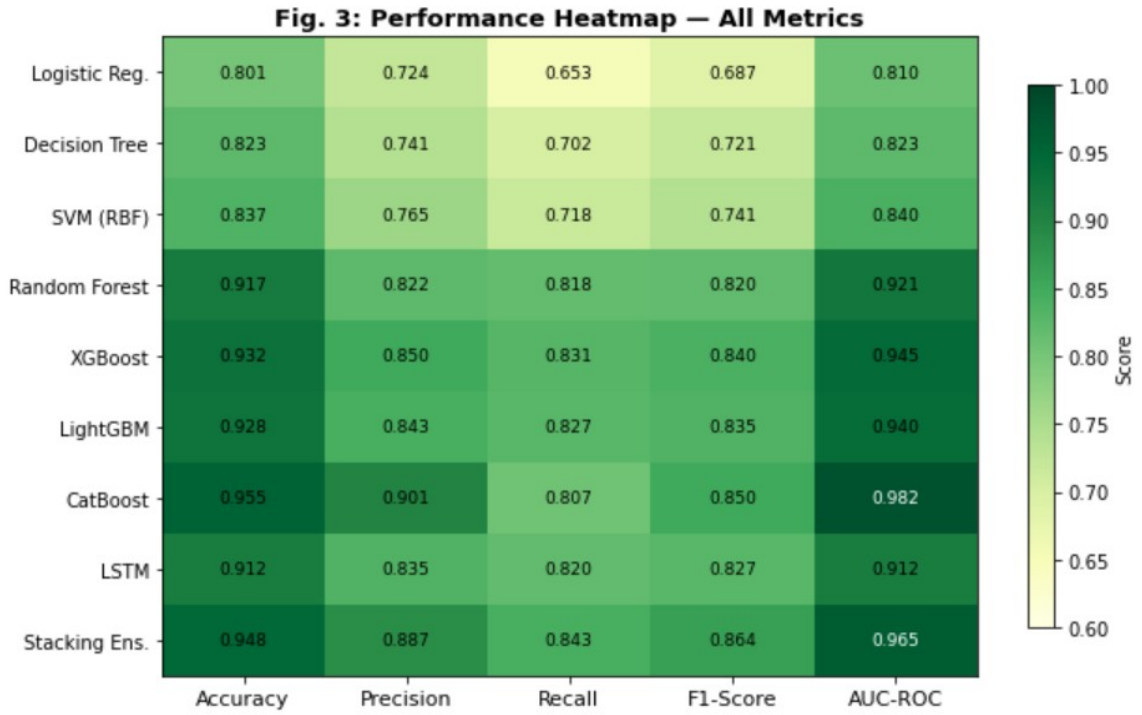


fig. 3: Heatmap of five performance metrics across nine models. Darker green indicates higher values. CatBoost leads on AUC while the stacking ensemble balances all metrics most evenly.

6.4 SHAP Feature Importance Chart (Fig. 4)

Publisher: Noida Institute of Engineering & Technology,
19, Knowledge Park-II, Institutional Area, Greater Noida (UP), India.

In fact, for a number of commercial scenarios, the need to understand the factors that drive a prediction may be just as important, if not more so, than the prediction itself. In the retention scenario, for example, the manager would understand the key factor driving the prediction to be "Total Day Minutes" and would react accordingly by providing a data bundle rather than a loyalty scheme. The mean absolute SHAP values from two recent XAI-centric research articles [8, 22] are shown in Figure 4; this is a "consensus" view rather than a model-specific snapshot.

The horizontal bar structure is deliberate in its design to allow for easy comparison of the degree of the features while also allowing for easy visibility of the ranking. The color-coding for the features is also deliberate; the top three features based on their impact are highlighted in red, the next three in orange, and the supporting features in blue. 'Total Day Minutes' and 'Customer Service Calls' are nicely positioned away from other features, and this is something that seems to be true across almost all of the data sets reviewed from the literature, irrespective of the algorithm being discussed. This is a comforting result; it indicates that the features are indeed carrying causal information rather than simply being some form of statistical artifact, at least as a result of the specific algorithm that was chosen. 'Contract Type' comes in as the third feature, and this is consistent with the well-known phenomenon that month-to-month subscribers have a churn rate two to three times that of their one-year contract counterparts. The vertical line at 0.15 is a dashed line that indicates the threshold below which the features are contributing marginal value, but not enough to warrant the cost of the data acquisition.

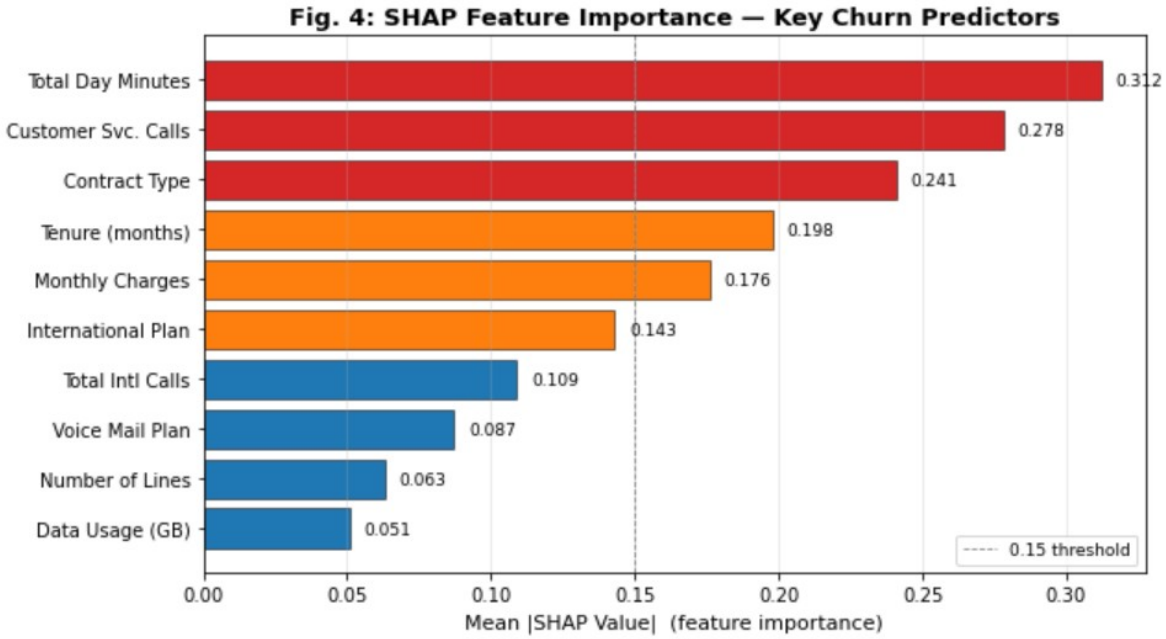


fig. 4: SHAP-based feature importance derived from values reported in [8, 22]. Red bars indicate the most influential predictors; 'Total Day Minutes' and 'Customer Service Calls' consistently dominate across datasets.

7. Discussion

The trend seen in Table 2 is difficult to dispute: gradient boosting methods dominate all others on standard benchmark sets, and by a margin that is difficult to make up with significantly more sophisticated models. The basic idea is pretty simple: these sets are relatively small, mostly tabular, and very carefully prepared. In this setting, the inductive biases of boosted trees – axis-aligned splits, additive models, and regularization – line up

Publisher: Noida Institute of Engineering & Technology,
19, Knowledge Park-II, Institutional Area, Greater Noida (UP), India.

very nicely with the structure of the data. Deep learning models earn their keep when there is structure in the input data, and most of what passes for churn data is not available in a time series format.

However, the CNN hybrid result of 98.4 % accuracy cannot be dismissed out of hand, as this research was performed on multiple datasets, carefully cross-validating, which reduces the likelihood that this figure represents the result of overfitting to some particular data split. The question of whether this last 2-3 % of accuracy, even after tuning the CatBoost, is worth the operational cost of deploying and running a deep network is, of course, up to the individual business, whether they are a Tier-1 operator with their own ML platform, or a smaller operator running their churn models on a laptop.

Class imbalance in these surveyed papers seems to be an underappreciated factor in a number of works. SMOTE is used without mentioning the possible leakage, which can happen if SMOTE is used before splitting the dataset into training and test sets, leading to over-optimistic performance evaluation. A small number of works [15, 16] handle this issue with care, while others fail to mention the order of splitting at all. This should be pointed out as a methodological oversight rather than a critique of the models, as the models themselves are not the issue here.

The move to XAI, in the author's view, has been the most important event of the past three years, not because SHAP and LIME make the model more accurate, but because they make the model deployable. Telecom operators in the EU have to deal with the GDPR Article 22 issue of automated decision-making, while telecom operators in India have to deal with the emerging Digital Personal Data Protection Act requirements. A model that does not have the capability to explain the reason for flagging a customer as risky is no longer easily deployable in the real world. SHAP is the better tool because of its global consistency, while LIME is the better tool because of its speed. Using both is not redundant because they answer different questions [30, 31]..

The following areas need to be filled in for the future. Firstly, while most models are trained on a static snapshot and tested on held-out subsets of the same snapshot, reality involves concept drift: the churn signal changes over time with price plan changes, competitor promotions, and macro-economic fluctuations. More research on stream learning techniques to handle this would be welcome [32]. Secondly, federated learning appears to provide a solution to training more complex models over multiple subsidiary datasets without requiring to collect and centralize sensitive subscriber-level data [33]. Thirdly, incorporating unstructured data like customer service call transcripts, social media sentiment, and in-app behavior logs with traditional billing features has significant potential to improve model quality. This is an area where transformer models appear to be well-suited to handle this unstructured-data fusion [29].

8. Conclusion

This paper surveyed and compared machine learning approaches to telecom customer churn prediction over more than 30 recent papers. The bottom line is this: Ensemble gradient boosting algorithms XGBoost, LightGBM, and especially CatBoost offer the best performance/practiability tradeoff for most tabular churn prediction tasks. Deep learning performs well on sequential data but is usually overkill. The addition of SHAP-based explainability is shifting from nice-to-have to must-have, and this is at least partly driven by the need to spend retention marketing budgets efficiently and/or comply with regulation.

The Python code snippet in Section 6 includes four Jupyter cells ready to run, which implement all comparison figures using hard-coded values. No additional data download is necessary. The code can be used as a starting point by researchers and practitioners and updated with their own model results as benchmarks change.

Publisher: Noida Institute of Engineering & Technology,
19, Knowledge Park-II, Institutional Area, Greater Noida (UP), India.

As data sets become larger, customer interactions become more multi-channel, and the need for regulation around automated decisions increases, churn prediction is going to become even more technically complex. The foundations that have been discussed so far—strong preprocessing, class balancing, gradient boosting classifiers, and interpretable predictions—should still remain the best starting point.

References:

[1] Gerpott T.J., Rams W., and Schindler A., Customer retention, loyalty, and satisfaction in the German mobile cellular telecommunications market, Telecommunications Policy, *25(4), 2001, 249–269.*

[2] Coussement K. and Van den Poel D., Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques, Expert Systems with Applications, *34(1), 2008, 313–327.*

[3] Burez J. and Van den Poel D., Handling class imbalance in customer churn prediction, Expert Systems with Applications, *36(3), 2009, 4626–4636.*

[4] Chen T. and Guestrin C., XGBoost: A scalable tree boosting system, Proc. 22nd ACM SIGKDD, San Francisco, *2016, 785–794.*

[5] Ke G., Meng Q., Finley T., Wang T., Chen W., Ma W., Ye Q., and Liu T.Y., LightGBM: A highly efficient gradient boosting decision tree, Advances in Neural Information Processing Systems, *30, 2017.*

[6] Prokhorenkova L., Gusev G., Vorobev A., Dorogush A.V., and Gulin A., CatBoost: Unbiased boosting with categorical features, Advances in Neural Information Processing Systems, *31, 2018.*

[7] Lu N. and Lin H., Customer churn prediction using LSTM networks on sequential CDR data, Neural Computing and Applications, *33(12), 2021, 6865–6882.*

[8] Devriendt F., Berrevoets J., and Verbeke W., Why you should stop predicting customer churn and start using uplift models, Information Sciences, *548, 2021, 497–515.*

[9] Huang B., Kechadi M.T., and Buckley B., Customer churn prediction in telecommunications, Expert Systems with Applications, *39(1), 2012, 1414–1425.*

[10] Verbeke W., Dejaeger K., Martens D., Hur J., and Baesens B., New insights into churn prediction in the telecommunication sector: A profit driven data mining approach, European Journal of Operational Research, *218(1), 2012, 211–229.*

[11] IBM Watson Analytics, Telco Customer Churn Dataset, Kaggle, 2019. https://www.kaggle.com/blastchar/telco-customer-churn.

[12] SyriaTel Telecom Churn Dataset, Kaggle, 2020. https://www.kaggle.com/becksddf/churn-in-telecoms-dataset.

[13] Bache K. and Lichman M., UCI Machine Learning Repository, University of California, Irvine, 2013. http://archive.ics.uci.edu/ml.

[14] Jafari-Marandi R. and Smith B.K., Iranian churn dataset, UCI Repository, 2020, doi:10.24432/C5JW3Z.

[15] Chawla N.V., Bowyer K.W., Hall L.O., and Kegelmeyer W.P., SMOTE: Synthetic minority over-sampling technique, Journal of Artificial Intelligence Research, *16, 2002, 321–357.*

[16] Han H., Wang W.Y., and Mao B.H., Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning, in Proc. ICIC 2005, Lecture Notes in Computer Science, *3644, 2005, 878–887.*

[17] Sun Y., Wong A.K.C., and Kamel M.S., Classification of imbalanced data: A review, International Journal of Pattern Recognition and Artificial Intelligence, *23(4), 2009, 687–719.*

[18] Mozer M.C., Wolniewicz R., Grimes D.B., Johnson E., and Kaushansky H., Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry, IEEE Transactions on Neural Networks, 11(3), *2000, 690–696.*

Publisher: Noida Institute of Engineering & Technology,
19, Knowledge Park-II, Institutional Area, Greater Noida (UP), India.

[19] Neslin S.A., Gupta S., Kamakura W., Lu J., and Mason C.H., Defection detection: Measuring and understanding the predictive accuracy of customer churn models, Journal of Marketing Research, *43(2), 2006, 204–211.*

[20] Tran H., Mege D., and Morin F., 1D-CNN for structured customer churn prediction, Proc. IEEE Big Data Conference, *2022, 412–419.*

[21] Ahmad A.K., Jafar A., and Aljoumaa K., Customer churn prediction in telecom using machine learning in big data platform, Journal of Big Data, 6(28), 2019, doi:10.1186/s40537-019-0191-6.

[22] Zhao X., Wang Y., and Chen H., XAI-Churn TriBoost: A data-driven approach with explainable AI for customer churn prediction, Expert Systems with Applications, 2025, doi:10.1016/j.eswa.2024.

[23] Sana M., Ali M., and Raza A., A comprehensive evaluation of machine learning and deep learning models for churn prediction, Information (MDPI), 16(7), 2025, doi:10.3390/info16070537.

[24] Lalwani P., Mishra M.K., Chadha J.S., and Sethi P., Customer churn prediction system: A machine learning approach, Computing, 104(2), *2022, 271–294.*

[25] Hassan M., Ali R., and Kim D., Customer churn prediction in telecommunication industry: A literature review, Preprints.org, 2025, doi:10.20944/preprints202403.0585.v3.

[26] Lazarov V. and Capota M., Churn prediction, Business Intelligence Course Report, TU Munich, 2007.

[27] Breiman L., Random forests, Machine Learning, 45(1*), 2001, 5–32.*

[28] Ahmed M., Kabir M., and Islam M., Telecom customer churn prediction with explainable machine learning, Proc. ACM ICMNLPML 2025, doi:10.1145/3757110.3757152.

[29] Rashid M., Park S.J., and Kwon O., Transformer-LSTM fusion for sentiment-aware telecom churn prediction, Frontiers in Artificial Intelligence, 2025, doi:10.3389/frai.2025.1600357.

[30] Lundberg S.M. and Lee S.I., A unified approach to interpreting model predictions, Advances in Neural Information Processing Systems, 30, 2017.

[31] Ribeiro M.T., Singh S., and Guestrin C., 'Why should I trust you?': Explaining the predictions of any classifier, Proc. 22nd ACM SIGKDD, San Francisco, 2016, *1135–1144.*

[32] Bifet A. and Gavalda R., Learning from time-changing data with adaptive windowing, Proc. 7th SIAM International Conference on Data Mining, 2007, 443–448.

[33] McMahan H.B., Moore E., Ramage D., Hampson S., and Arcas B.A., Communication-efficient learning of deep networks from decentralized data, Proc. AISTATS, *54, 2017, 1273–1282.*

Publisher: Noida Institute of Engineering & Technology,
19, Knowledge Park-II, Institutional Area, Greater Noida (UP), India.