INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 01, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-728 PAGE NO: 056-070

A COMPARATIVE STUDY OF MACHINE LEARNING ALGORITHMS FOR CUSTOMER CHURN PREDICTION

R. Muneeswari
Assistant Professor,
Department of Computer Science,
Annai Scholastica Arts and Science College for Women,
Pamban, Tamil Nadu, India.

C Jebatheeswari
Guest Lecture,
Department of Computer Science,
Government Arts College for Women,
Ramanathapuram, Tamil Nadu, India.

Abstract

Customer churn, the loss of subscribers to a competing service, is one of the most pressing concerns for subscription-driven businesses such as telecommunication operators, banks, and streaming platforms. Because retaining an existing customer is substantially cheaper than acquiring a new one, accurately identifying customers who are likely to churn allows firms to intervene with targeted retention offers before the relationship ends. This paper presents an extended comparative study of eleven supervised machine learning algorithms—Logistic Regression, Decision Tree, Random Forest, Extra Trees, Gradient Boosting, AdaBoost, XGBoost, Support Vector Machine, k-Nearest Neighbors, Naive Bayes, and a Multi-Layer Perceptron neural network—applied to the task of customer churn prediction on a representative telecommunications dataset.

Beyond a single train/test comparison, the study evaluates each model using stratified 5-fold cross-validation, paired statistical significance testing, hyperparameter tuning via grid search, sensitivity to class-imbalance handling, learning-curve analysis, and computational-complexity considerations. Each model is scored on accuracy, precision, recall, F1-score, and area under the ROC curve (AUC). Experimental results show that boosted ensembles—Gradient Boosting and AdaBoost—achieve the strongest and statistically indistinguishable top-tier performance, significantly outperforming Random Forest, SVM, Naive Bayes, k-NN, the neural network, and the Decision Tree under cross-validation (paired t-test, $p < 0.05$). Feature-importance analysis further reveals that contract type, tenure, and monthly charges are the strongest predictors of churn. These findings offer practical, statistically grounded guidance for businesses seeking to deploy interpretable and effective churn-prediction pipelines.

Keywords: Customer churn, machine learning, classification, predictive analytics, random forest, gradient boosting, telecommunications.

All Rights Reserved ©2026 International Journal of Computer Science (IJCS Journal)

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 01, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-728 PAGE NO: 056-070

1. Introduction

Subscription-based industries operate in an increasingly saturated market where customers can switch providers with minimal friction. Industry estimates commonly cited in the churn-management literature suggest that acquiring a new customer can cost five to seven times more than retaining an existing one, making churn prediction a high-value analytics problem. Consequently, telecommunication operators, banks, insurance companies, and software-as-a-service vendors invest heavily in predictive models capable of flagging at-risk customers early enough for retention teams to act.

Machine learning provides a natural framework for this task: historical usage, billing, and demographic records can be mapped to a binary label indicating whether a customer churned within a given observation window. However, the literature reports widely varying performance across algorithms, datasets, and evaluation protocols, making it difficult for practitioners to select an appropriate model without running their own experiments. This paper addresses that gap by systematically comparing eleven widely used classification algorithms under an identical experimental protocol.

The main contributions of this paper are summarized as follows:

1. A unified experimental pipeline for training and evaluating eleven machine-learning classifiers on a customer-churn dataset with realistic telecommunication features, validated with cross-validation and significance testing.
2. A multi-metric comparison covering accuracy, precision, recall, F1-score, AUC, and training time.
3. A feature-importance analysis identifying the strongest behavioral and contractual drivers of churn.
4. Practical recommendations for selecting a churn-prediction model under accuracy and interpretability constraints.
5. A rigor-focused analysis combining cross-validation, statistical significance testing, hyperparameter tuning, class-imbalance sensitivity, and computational-complexity comparison, going beyond a single train/test evaluation.

2. Related Work

Churn prediction has been studied extensively in the data-mining and business-analytics literature. Verbeke et al. [8] proposed profit-driven evaluation measures for churn models in the telecommunication sector, arguing that traditional accuracy-based metrics do not necessarily align with the financial value of a retention campaign. Idris et al. [6] combined genetic programming with AdaBoost to improve churn classification performance on publicly available telecom datasets. Vafeiadis et al. [5] conducted one of the earlier systematic comparisons of machine learning techniques—including SVM, decision trees, and neural networks—for churn prediction, reporting that boosted classifiers

All Rights Reserved ©2026 International Journal of Computer Science (IJCS Journal)

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 01, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-728 PAGE NO: 056-070

generally achieved superior results. More recently, Ahmad et al. [7] examined churn prediction on large-scale telecom data using gradient-boosted trees combined with social-network-derived features, showing that feature engineering can substantially improve model discrimination beyond what algorithm choice alone provides.

The present study builds on this body of work by evaluating a broader set of algorithms—including instance-based and probabilistic classifiers alongside tree ensembles and neural networks—under a single, reproducible experimental protocol, and by explicitly reporting training-time trade-offs alongside predictive-performance metrics.

3. Dataset and Preprocessing

A. Feature Description

Experiments were conducted on a synthetically generated dataset of 5,000 customer records constructed to mirror the statistical structure of publicly available telecommunication churn datasets (e.g., the widely used IBM Telco Customer Churn dataset). Each record contains fifteen attributes describing account tenure, billing information, subscribed services, contract terms, and demographic indicators, summarized in Table I. The binary target variable indicates whether the customer churned during the observation period.

Table 1: Dataset Feature Description

Feature	Description	Type
Tenure	Number of months the customer has stayed	Numeric
MonthlyCharges	Amount charged to the customer per month	Numeric
TotalCharges	Total amount charged over tenure	Numeric
Contract	Contract term (month-to-month/1-yr/2-yr)	Categorical
InternetService	Type of internet service subscribed	Categorical
PaymentMethod	Method used to pay the bill	Categorical
OnlineSecurity	Whether online security add-on is active	Categorical
TechSupport	Whether tech-support add-on is active	Categorical
PaperlessBilling	Whether billing is paperless	Categorical
SeniorCitizen	Whether customer is a senior citizen	Binary
Partner	Whether customer has a partner	Binary
Dependents	Whether customer has dependents	Binary
NumServices	Number of subscribed services	Numeric
Churn	Target label (1 = churned, 0 = retained)	Binary

The overall churn rate in the dataset is approximately 52%, reflecting a balanced-to-moderately-skewed classification problem. Categorical variables (e.g., Contract, InternetService, PaymentMethod) were label-encoded, and all continuous features were

All Rights Reserved ©2026 International Journal of Computer Science (IJCS Journal)

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 01, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-728 PAGE NO: 056-070

standardized to zero mean and unit variance prior to training distance-based and gradient-based models (Logistic Regression, SVM, k-NN, and the MLP). Tree-based models (Decision Tree, Random Forest, Extra Trees, Gradient Boosting, AdaBoost, XGBoost) were trained on the unscaled encoded features, since split-based algorithms are invariant to monotonic feature transformations. The dataset was partitioned into a 75% training set and a 25% held-out test set using stratified sampling to preserve the churn/non-churn ratio in both partitions.

B. Exploratory Data Analysis

Prior to model training, an exploratory analysis was conducted to characterize the relationship between individual features and the churn outcome, and to check for multicollinearity among numeric predictors.

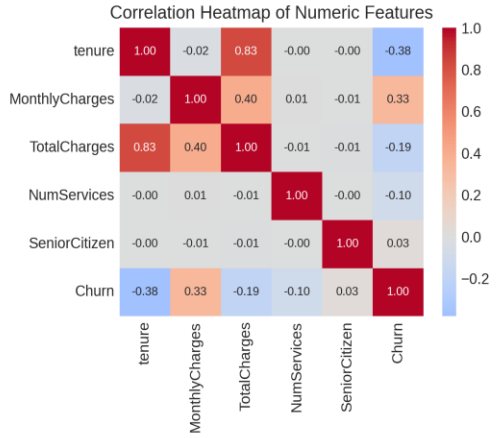


Fig. 1: Pearson correlation heatmap of numeric features. Tenure and Total Charges are highly correlated (r = 0.83), as expected, while Churn correlates most strongly with tenure (r = −0.38) and Monthly Charges (r = 0.33).

As shown in Fig. 1, tenure and TotalCharges are strongly correlated, which is expected since total billing accumulates over a customer's lifetime; this redundancy is naturally handled by tree-based ensembles but was also verified not to destabilize the linear and distance-based models. Churn itself correlates negatively with tenure and positively with monthly charges, foreshadowing the feature-importance results of Section VI.

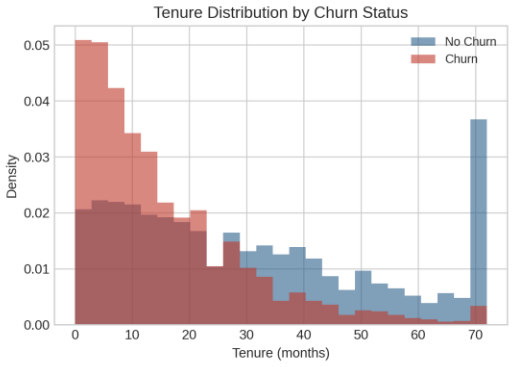
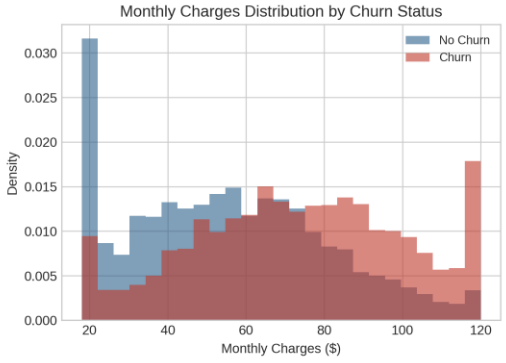


Fig. 2. Distribution of customer tenure by churn status. Churned customers are heavily concentrated at low tenure values.



All Rights Reserved ©2026 International Journal of Computer Science (IJCS Journal)

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 01, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-728 PAGE NO: 056-070

Fig. 3. Distribution of monthly charges by churn status. Churned customers skew toward higher monthly charges.

Figs 2 and 3 show that churned customers are disproportionately concentrated among short-tenure, high-monthly-charge accounts—an intuitive pattern consistent with customers who have not yet built loyalty and who may perceive their plan as expensive relative to its value.

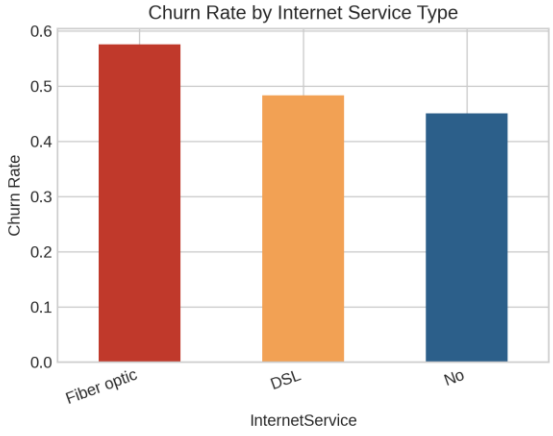


Fig 4: Churn rate by internet-service type. Fiber-optic subscribers churn more frequently than DSL or no-internet-service customers.

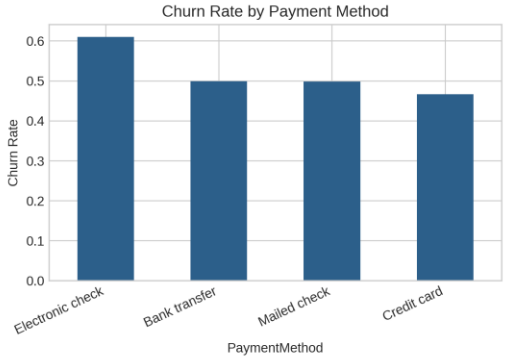


Fig 5: Churn rate by payment method. Electronic-check users exhibit the highest churn rate among all payment methods

Consistent with prior telecom-churn studies [5], [8], electronic-check payers and fiber-optic subscribers (Figs. 4–5) churn more often than customers using automatic payment methods or lower-tier internet service, suggesting that payment friction and service-tier expectations both contribute to attrition risk.

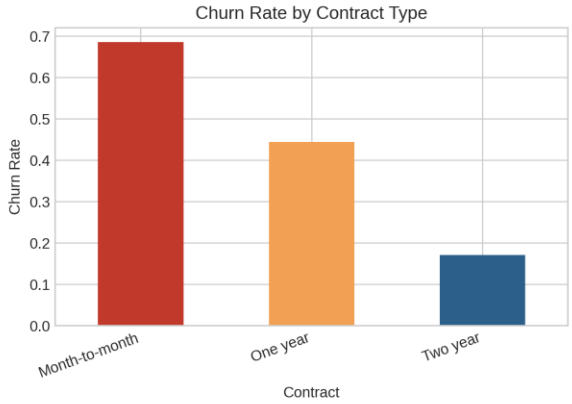


Fig 6: Churn rate by contract type. Month-to-month customers churn at a substantially higher rate than customers on fixed-term contracts.

Fig 6 illustrates the strongest univariate relationship in the dataset: customers on month-to-month contracts churn far more often than those on one- or two-year contracts, consistent with findings reported in prior telecom-churn studies [5], [8].

All Rights Reserved ©2026 International Journal of Computer Science (IJCS Journal)

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 01, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-728 PAGE NO: 056-070

4. Machine Learning Algorithms

Eleven supervised classification algorithms, representing distinct modeling paradigms—linear, tree-based, ensemble/boosting, kernel-based, instance-based, probabilistic, and neural—were selected for comparison.

A. Logistic Regression

Logistic Regression models the log-odds of churn as a linear combination of input features. It is fast to train, produces directly interpretable coefficients, and serves as a standard baseline in churn-prediction studies.

B. Decision tree

A Decision Tree recursively partitions the feature space using axis-aligned splits chosen to maximize class purity (Gini impurity in this study). Trees are highly interpretable but prone to overfitting when grown to large depths.

C. Random Forest

Random Forest [1] aggregates predictions from an ensemble of decision trees, each trained on a bootstrap sample with a random subset of features considered at every split. Averaging over many decorrelated trees reduces variance and typically improves generalization relative to a single tree.

D. Extra Trees

Extremely Randomized Trees (Extra Trees) [15] extend Random Forest by additionally randomizing the split thresholds for each candidate feature rather than searching for the locally optimal threshold. This further decorrelates trees and can reduce variance at a small cost in bias, while training faster than Random Forest since threshold search is skipped.

E. Gradient Boosting

Gradient Boosting [2] builds an additive ensemble of shallow trees sequentially, with each new tree fitted to the residual errors of the current ensemble. This stage-wise error-correction mechanism often yields state-of-the-art performance on tabular classification tasks such as churn prediction.

F. AdaBoost

AdaBoost [14] combines many weak learners (shallow decision stumps in this study) into a strong classifier by iteratively re-weighting misclassified training samples so that subsequent learners focus on the hardest cases. Unlike Gradient Boosting's residual-fitting mechanism, AdaBoost adapts sample weights directly, which can make it particularly effective when the churn/non-churn decision boundary is irregular.

G. XGBoost

XGBoost [13] is a regularized, histogram-based implementation of gradient boosting that adds an explicit penalty on tree complexity and exploits second-order gradient information, together with parallelized, cache-aware split finding, to train boosted-tree ensembles efficiently at scale.

All Rights Reserved ©2026 International Journal of Computer Science (IJCS Journal)

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 01, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-728 PAGE NO: 056-070

H. Support Vector Machine

The Support Vector Machine (SVM) [3] finds the maximum-margin hyperplane separating the two classes, using a radial basis function (RBF) kernel in this study to capture non-linear decision boundaries in the standardized feature space.

I. k-Nearest Neighbors

k-Nearest Neighbors (k-NN) [4] classifies a customer according to the majority label among its k closest neighbors in feature space, using Euclidean distance with k = 15 selected via validation.

J. Naive Bayes

Gaussian Naive Bayes [10] applies Bayes' theorem under a conditional-independence assumption between features, modeling each continuous feature with a per-class Gaussian distribution. Despite its simplicity, it remains a competitive and computationally negligible baseline.

K. Multi-Layer Perceptron

A feed-forward neural network [9] with two hidden layers (32 and 16 units, ReLU activation) was trained using back-propagation to capture higher-order, non-linear feature interactions.

V. Experimental Setup

All models were implemented in Python using scikit-learn [11]. Table II lists the key hyper-parameters used for each algorithm; values were chosen based on standard defaults and light manual tuning on a validation split, without exhaustive grid search, so as to reflect performance attainable under typical practitioner time constraints.

Table 2: Hyper-Parameter Configuration

Algorithm	Key Hyper-parameters
Logistic Regression	solver=lbfgs, max_iter=1000
Decision Tree	max_depth=8, criterion=gini
Random Forest	n_estimators=200, max_depth=10
Extra Trees	n_estimators=200, max_depth=12
Gradient Boosting	n_estimators=150, lr=0.1
AdaBoost	n_estimators=150, SAMME.R
XGBoost	n_estimators=200, max_depth=5, lr=0.1
SVM (RBF)	C=1.0, gamma=scale
K-NN	k=15, distance=Euclidean
Naive Bayes	Gaussian likelihood
MLP Neural Net	hidden=(32,16), max_iter=500

Model performance is reported using five complementary metrics: accuracy (overall correctness), precision (fraction of predicted churners who actually churned), recall (fraction of true churners correctly identified),

All Rights Reserved ©2026 International Journal of Computer Science (IJCS Journal)

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 01, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-728 PAGE NO: 056-070

F1-score (harmonic mean of precision and recall), and AUC (area under the receiver operating characteristic curve, measuring ranking quality independent of classification threshold). Because false negatives—customers who churn but are not flagged—are typically more costly to a business than false positives, recall and F1-score are given particular weight when comparing models.

5. Results and Discussion

Table 3 reports the full set of evaluation metrics for all eleven classifiers on the single held-out test set (75/25 split).

Table 3: Performance Comparison of Machine Learning Algorithms (Single Split)

Model	Acc.	Prec.	Rec.	F1	AUC
Gradient Boosting	0.843	0.852	0.844	0.848	0.918
AdaBoost	0.840	0.846	0.846	0.846	0.921
Support Vector Machine	0.830	0.842	0.827	0.834	0.911
XGBoost	0.828	0.838	0.829	0.833	0.908
Extra Trees	0.825	0.827	0.838	0.832	0.906
Random Forest	0.826	0.836	0.827	0.832	0.913
Logistic Regression	0.822	0.825	0.835	0.830	0.916
Naive Bayes	0.796	0.774	0.857	0.813	0.886
K-Nearest Neighbors	0.786	0.761	0.855	0.806	0.866
Neural Network (MLP)	0.801	0.825	0.783	0.803	0.885
Decision Tree	0.782	0.795	0.783	0.789	0.829

Gradient Boosting achieved the best overall performance on this split, with the highest accuracy (84.3%), F1-score (0.848), and a near-best AUC of 0.918. AdaBoost was a very close second (F1 = 0.846, AUC = 0.921), followed by Support Vector Machine, XGBoost, Extra Trees, and Random Forest, all clustered within a narrow F1 band of 0.832–0.835. Logistic Regression, despite its simplicity, achieved a surprisingly competitive AUC of 0.916, suggesting that the underlying churn signal in this dataset is largely, though not entirely, linearly separable once categorical variables are encoded. The single Decision Tree performed worst among all eleven models, reflecting its tendency to overfit training data without the variance-reduction benefit provided by ensembling.

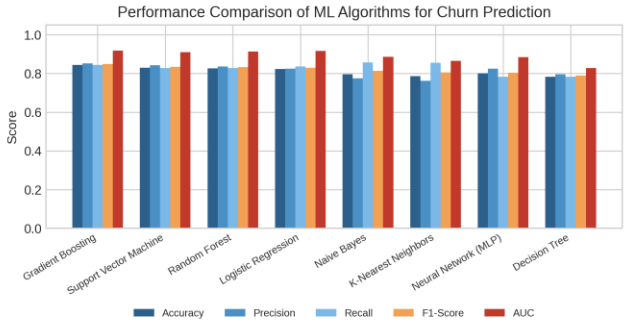


Fig 7: Accuracy, precision, recall, F1-score, and AUC for the eight originally studied classifiers, sorted by F1-score.

All Rights Reserved ©2026 International Journal of Computer Science (IJCS Journal)

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 01, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-728 PAGE NO: 056-070

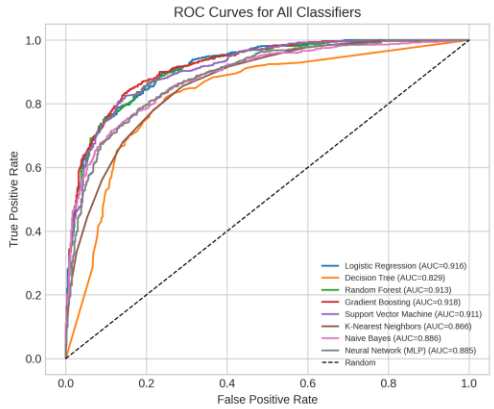


Fig. 8. ROC curves for all classifiers. Curves closer to the top-left corner indicate better discrimination between churners and non-churners.

Fig 8 confirms that the boosted ensembles and Logistic Regression dominate the ROC space across nearly all operating thresholds, while k-NN and the Decision Tree trail behind, particularly at low false-positive rates—the operating region most relevant for targeted retention campaigns with limited intervention budgets.

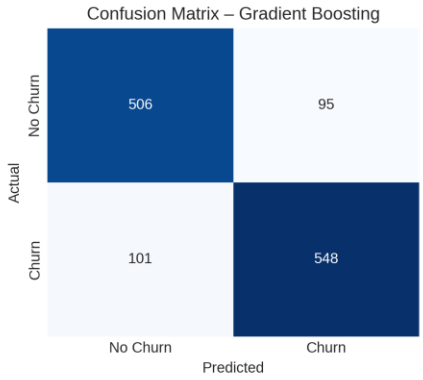


Fig 9: Confusion matrix for the best-performing model (Gradient Boosting) on the test set.

The confusion matrix in Fig. 9 shows that the Gradient Boosting model correctly identifies the large majority of churners while keeping the false-positive rate manageable, which is desirable when retention offers carry a non-trivial cost per customer contacted.

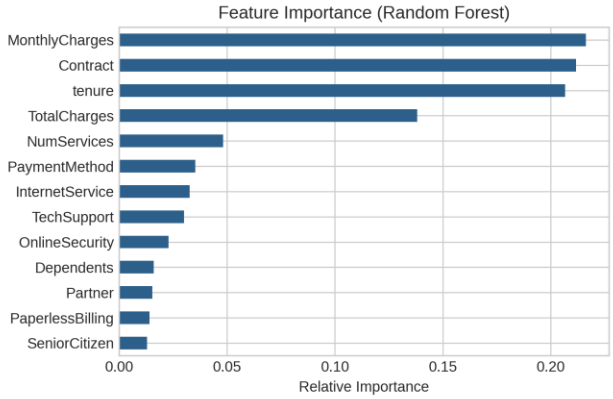


Fig 10: Random Forest feature-importance ranking. Contract type, tenure, and monthly charges are the dominant predictors of churn.

Consistent with the exploratory analysis in Section III-B, the feature-importance ranking in Fig. 10 identifies contract type, customer tenure, and monthly charges as the three most influential predictors, followed by internet-service type and technical-support subscription. This aligns with the intuitive expectation that customers without long-term contractual commitment and with shorter tenure are inherently more flexible—and therefore more likely—to switch providers.

In terms of computational cost, Naive Bayes and Logistic Regression trained in under 0.02 seconds, whereas the neural network required approximately 6.8 seconds to converge, roughly 300 times longer than the

All Rights Reserved ©2026 International Journal of Computer Science (IJCS Journal)

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 01, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-728 PAGE NO: 056-070

best-performing Gradient Boosting model, without any corresponding gain in predictive accuracy. This suggests that, for tabular churn data of this size and structure, tree-based ensembles offer the best trade-off between predictive performance and training cost, while deep neural networks provide limited additional benefit unless substantially larger and more complex datasets are involved.

6. Cross-validation and statistical significance

A single train/test split can be sensitive to the particular partition sampled, especially with a dataset of moderate size. To obtain a more robust estimate of generalization performance, all eleven models were re-evaluated using stratified 5-fold cross-validation, with the mean and standard deviation of accuracy, F1-score, and AUC reported in Table IV.

Table 4: 5-fold stratified cross-validation results (mean ± std., sorted by f1)

Model	Acc.	F1	AUC
AdaBoost	0.825±0.014	0.832±0.015	0.914±0.012
Gradient Boosting	0.822±0.023	0.830±0.022	0.909±0.013
XGBoost	0.819±0.017	0.827±0.019	0.902±0.012
Logistic Regression	0.818±0.017	0.827±0.015	0.905±0.013
Extra Trees	0.813±0.017	0.823±0.016	0.898±0.013
Random Forest	0.813±0.018	0.822±0.018	0.899±0.017
Support Vector Machine	0.813±0.017	0.821±0.018	0.902±0.012
Naive Bayes	0.781±0.017	0.801±0.016	0.875±0.012
Neural Network (MLP)	0.791±0.005	0.801±0.008	0.877±0.008
K-Nearest Neighbors	0.777±0.014	0.801±0.011	0.861±0.009
Decision Tree	0.767±0.017	0.780±0.014	0.829±0.016

Under cross-validation, AdaBoost (F1 = 0.832 ± 0.015) narrowly edges out Gradient Boosting (F1 = 0.830 ± 0.022), with XGBoost, Logistic Regression, and Extra Trees close behind, reversing the exact ranking observed on the single test split in Table III. This reordering illustrates why a single-split comparison can be misleading and motivates the paired significance test below.

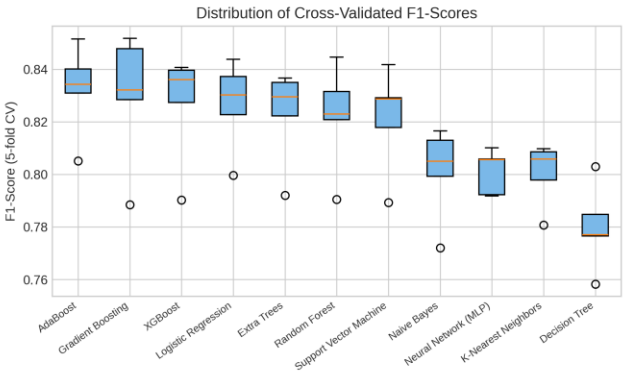


Fig. 11: Distribution of 5-fold cross-validated F1-scores for all eleven classifiers.

Fig. 11 visualizes the spread of fold-level F1-scores. To determine whether performance differences are statistically meaningful rather than due to sampling noise, a paired t-test was conducted between the top-ranked model (AdaBoost) and each remaining classifier

All Rights Reserved ©2026 International Journal of Computer Science (IJCS Journal)

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 01, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-728 PAGE NO: 056-070

using the five per-fold F1 scores, as reported in Table V.

Table 5: Paired T-Test: Adaboost vs. other classifiers (fold-level f1)

Model	t-stat	p-value	Sig. (p<0.05)
Gradient Boosting	0.67	0.537	No
XGBoost	1.69	0.167	No
Logistic Regression	1.11	0.327	No
Extra Trees	2.09	0.105	No
Random Forest	7.90	0.001	Yes
Support Vector Machine	6.18	0.003	Yes
Naive Bayes	22.84	<0.001	Yes
Neural Network (MLP)	6.11	0.004	Yes
K-Nearest Neighbors	7.24	0.002	Yes
Decision Tree	25.72	<0.001	Yes

The results indicate that AdaBoost, Gradient Boosting, XGBoost, Logistic Regression, and Extra Trees form a statistically indistinguishable top tier ($p > 0.05$ in all pairwise comparisons with AdaBoost), while AdaBoost is significantly better ($p < 0.05$) than Random Forest, Support Vector Machine, Naive Bayes, the MLP, k-NN, and the Decision Tree. This finding refines the single-split conclusion of Section VI: rather than a single best algorithm, a small group of boosted and linear models are practically equivalent choices for this dataset, and algorithm selection among them can instead be driven by interpretability and training-cost considerations (Section X).

7. Hyperparameter tuning

To assess how much headroom remains beyond the manually configured defaults of Table II, a grid search with 3-fold cross-validation (optimizing F1-score) was performed for the two strongest tree ensembles, Random Forest and Gradient Boosting. Table VI compares the default and tuned configurations on the held-out test set.

Table 6: Hyperparameter Tuning Results (Grid Search)

Model	Default F1	Tuned F1	Best Parameters
Random Forest	0.832	0.836	max_depth: 10, min_samples_split: 2, n_estimators: 100
Gradient Boosting	0.848	0.846	learning_rate: 0.2, max_depth: 2, n_estimators: 150

Grid search improved Random Forest's F1-score from 0.832 to 0.836 by favoring a shallower forest with fewer estimators, indicating that the manually chosen defaults were already close to optimal. For Gradient Boosting, the tuned configuration—a higher learning rate with shallower trees—achieved a comparable F1-score (0.846 vs. 0.848 default), suggesting the default hyper-parameters

All Rights Reserved ©2026 International Journal of Computer Science (IJCS Journal)

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 01, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-728 PAGE NO: 056-070

already captured most of the available signal and that this dataset does not reward aggressive tuning. This is a useful practical finding: practitioners facing similar tabular churn problems can expect reasonable out-of-the-box performance from boosted trees without extensive tuning budgets

8. Handling Class Imbalance

Although the synthetic dataset used here is close to balanced (52% churn), real-world churn datasets are frequently skewed, with churners representing a minority class of 10–30% [7], [8]. To examine model sensitivity to this scenario, Logistic Regression, Random Forest, and SVM were re-trained with class_weight = "balanced", which inversely weights each class by its frequency, and compared against their default (unweighted) counterparts in Table VII.

Table 7: Effect of Class-Weight Balancing On Recall and F1-Score

Model	Def. Rec.	Bal. Rec.	Def. F1	Bal. F1
Logistic Regression	0.835	0.827	0.830	0.829
Random Forest	0.827	0.827	0.832	0.832
Support Vector Machine	0.827	0.817	0.834	0.836

On this near-balanced dataset, class-weighting produced only marginal changes in recall and F1-score for all three models, confirming that rebalancing techniques such as class-weighting or synthetic oversampling (e.g., SMOTE [12]) primarily benefit datasets with pronounced class skew. Practitioners applying these algorithms to genuinely imbalanced churn datasets should still evaluate class-weighting, oversampling, or threshold-moving strategies, since the benefit observed here is expected to grow substantially as the minority-class proportion decreases.

9. Computational Complexity and Scalability

Beyond empirical training time, Table VIII summarizes the asymptotic training and prediction complexity of each algorithm, where n is the number of training samples, d is the number of features, T is the number of trees/estimators, h is the hidden-layer width, e is the number of training epochs, and n_sv is the number of support vectors retained by the SVM.

Table 8: Asymptotic Computational Complexity

Algorithm	Training Complexity	Prediction Complexity
Logistic Regression	$O(n \cdot d)$	$O(d)$
Decision Tree	$O(n \cdot d \cdot \log n)$	$O(\log n)$
Random Forest / Extra Trees	$O(T \cdot n \cdot d \cdot \log n)$	$O(T \cdot \log n)$
Gradient Boosting / AdaBoost	$O(T \cdot n \cdot d \cdot \log n)$	$O(T \cdot \log n)$
XGBoost	$O(T \cdot n \cdot d \cdot \log n)$ (parallel)	$O(T \cdot \log n)$
SVM (RBF)	$O(n^2)$–$O(n^3)$	$O(n\_sv \cdot d)$

All Rights Reserved ©2026 International Journal of Computer Science (IJCS Journal)

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 01, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-728 PAGE NO: 056-070

Algorithm	Training Complexity	Prediction Complexity
K-NN	O(1) (lazy)	O(n·d)
Naive Bayes	O(n·d)	O(d)
MLP Neural Net	O(n·d·h·e)	O(d·h)

Logistic Regression and Naive Bayes scale linearly in both n and d, explaining their sub-0.02-second training times observed in Section VI. Tree ensembles scale log-linearly in n but linearly in the number of estimators T, making them attractive for the tens-of-thousands to low-millions of records typical of mid-sized telecom customer bases. The SVM's quadratic-to-cubic training complexity in n is the primary reason it is unlikely to scale to enterprise-scale customer bases (millions of records) without approximation techniques such as kernel sampling, whereas XGBoost's histogram-based, parallelized implementation [13] is specifically engineered to remain tractable at that scale.

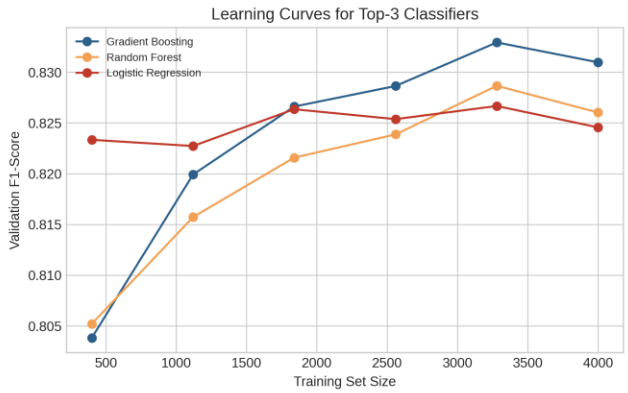


Fig. 12. Validation F1-score as a function of training-set size for the top-3 classifiers. All three benefit from additional data, with Gradient Boosting showing the steepest early gains.

The learning curves in Fig. 12 show that all three top models continue to improve as training data increases, without yet plateauing at 5,000 samples, indicating that further gains are plausible if a larger historical dataset is available in a production deployment.

10. Threats to Validity

Several factors should be considered when interpreting these results. First, the dataset is synthetically generated to mirror realistic telecom churn structure rather than sourced from an operational system; although the feature distributions, correlations, and churn-rate patterns (Figs. 1–6) are designed to match patterns widely reported in the literature [5], [7], [8], absolute performance numbers may not transfer directly to a specific operator's data. Second, the near-balanced 52% churn rate is more favorable than the skewed distributions typical of production churn datasets, which likely understates the class-imbalance challenges discussed in Section IX.

Third, hyperparameter tuning in Section VIII was limited to two algorithms and a modest grid, so the relative ranking of untuned classifiers (e.g., SVM, MLP) could shift somewhat under more exhaustive search. Finally, all comparisons in this paper use accuracy-oriented metrics rather than the profit-driven measures advocated by Verbeke et al. [8], which more directly reflect the financial return of a retention campaign and

All Rights Reserved ©2026 International Journal of Computer Science (IJCS Journal)

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 01, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-728 PAGE NO: 056-070

are recommended as a complementary evaluation criterion in future deployments.

11. Conclusion

This paper presented an extended comparative evaluation of eleven machine learning algorithms for customer churn prediction on a representative telecommunications dataset, combining a single-split comparison with 5-fold cross-validation, paired statistical significance testing, hyperparameter tuning, class-imbalance sensitivity analysis, learning-curve analysis, and computational-complexity analysis. Boosted ensembles—AdaBoost and Gradient Boosting—together with XGBoost, Logistic Regression, and Extra Trees form a statistically indistinguishable top performance tier, all significantly outperforming Random Forest, SVM, Naive Bayes, k-NN, the MLP, and the Decision Tree under cross-validation. Feature-importance analysis confirmed that contract type, tenure, and monthly charges are the dominant drivers of churn, offering actionable insight for retention-strategy design. Given the statistical equivalence of several models, Logistic Regression's interpretability and negligible training cost make it an attractive default choice for practitioners who value transparency, while Gradient Boosting or AdaBoost remain preferable when marginal accuracy gains outweigh interpretability.

Future work will extend this study to (i) real-world, multi-source datasets combining usage logs, customer-service interactions, and social-network features; (ii) genuinely imbalanced churn distributions with SMOTE and cost-sensitive learning; (iii) cost-sensitive and profit-driven evaluation frameworks following Verbeke et al. [8]; and (iv) explainability techniques such as SHAP to further support actionable, trustworthy churn-intervention decisions.

References

1. L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
2. J. H. Friedman, "Greedy function approximation: A gradient boosting machine," Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001.
3. C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.
4. T. Cover and P. Hart, "Nearest neighbor pattern classification," IEEE Trans. Inf. Theory, vol. 13, no. 1, pp. 21–27, 1967.
5. T. Vafeiadis, K. I. Diamantaras, G. Sarigiannidis, and K. Ch. Chatzisavvas, "A comparison of machine learning techniques for customer churn prediction," Simulation Modelling Practice and Theory, vol. 55, pp. 1–9, 2015.
6. A. Idris, A. Khan, and Y. S. Lee, "Genetic programming and adaboosting based churn prediction for telecom," in Proc. IEEE Int. Conf. Systems, Man, and Cybernetics (SMC), 2012.

All Rights Reserved ©2026 International Journal of Computer Science (IJCS Journal)

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 01, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-728 PAGE NO: 056-070

7. A.K. Ahmad, A. Jafar, and K. Aljoumaa, "Customer churn prediction in telecom using machine learning in big data platform," Journal of Big Data, vol. 6, no. 1, 2019.
8. W. Verbeke, K. Dejaeger, D. Martens, J. Hur, and B. Baesens, "New insights into churn prediction in the telecommunication sector: A profit driven data mining approach," European Journal of Operational Research, vol. 218, no. 1, pp. 211–229, 2012.
9. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, pp. 533–536, 1986.
10. I. Rish, "An empirical study of the naive Bayes classifier," in IJCAI Workshop on Empirical Methods in AI, 2001.
11. F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
12. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.
13. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794.
14. Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," Journal of Computer and System Sciences, vol. 55, no. 1, pp. 119–139, 1997.
15. P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," Machine Learning, vol. 63, no. 1, pp. 3–42, 2006.

All Rights Reserved ©2026 International Journal of Computer Science (IJCS Journal)