JOIN (Jurnal Online Informatika)
p-ISSN: 2528-1682, e-ISSN: 2527-9165
Volume 11 Number 1 | June 2026: 38-47
DOI: 10.15575/join.v11i1.1783

Optimizing Stacking Ensemble Models for Customer Churn Prediction in the Telecommunications Industry

Rofik[1], Jumanto Unjung[2], Dwika Ananda Agustina Pertiwi[3], Much Aziz Muslim[4]
[1,2,4]Department of Computer Science, Universitas Negeri Semarang, Indonesia
[3]Faculty Technology Management and Business, Universiti Tun Hussein Onn Malaysia, Malaysia

Article Info

Article history:

Received July 29, 2025
Revised October 26, 2025
Accepted November 01, 2025
Published April 25, 2026

Keywords:

Churn prediction
Logistic regression
Machine learning
Stacking ensemble
Telecommunications

ABSTRACT

One of the biggest challenges in the telecommunications industry is predicting churn, which is the condition when a customer unsubscribes and switches to another service provider. In an era of competitive market conditions, retaining customers is much more efficient than acquiring new customers. Conventional prediction models are often unable to capture the complexity of customer behavior patterns, resulting in a lower accuracy than optimal. This study aims to optimize customer churn prediction performance by developing a stacking ensemble model that combines several classification algorithms to improve model performance. Fourteen algorithms were tested, and the six algorithms with the best accuracy were selected as base learners, while Logistic Regression was selected as the meta-learner. The stacking model testing was carried out sequentially through a combination of 6 algorithms with the same meta-learner algorithm. Testing was also carried out with and without using the SMOTE data balancing method to evaluate the effect of data balancing on the prediction results. The results of this study show that the combination of the Adaboost, Ridge Classifier, and Logistic Regression algorithms can produce the highest accuracy of 82.97%, which exceeds the prediction performance of a single algorithm. This research contributes to demonstrating an effective stacking ensemble configuration for predicting customer churn in the telecommunications industry and emphasizes that the selection of the right algorithm combination has a greater impact on model performance than the number of algorithms used.

Corresponding Author:

Jumanto Unjung,
Department of Computer Science, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang, Kampus Sekaran, Semarang, Indonesia. 50229
Email: jumanto@mail.unnes.ac.id

1. INTRODUCTION

Customer churn is a condition where customers decide to unsubscribe or stop using the services of a company and switch to use and subscribe to another company [1], [2]. This is one of the impacts of increased competition in an industry. So, to maintain their business, companies must adapt and find ways to minimize customer churn, especially for companies engaged in the telecommunications industry [3], [4]. Customers are the main asset of a business. The existence of customers is one of the main factors in the sustainability of a company [5].

Unfortunately, the annual churn phenomenon is recorded to reach 20-40%, and the cost of acquiring a new customer can require 5-10 times more effort than maintaining an existing customer [6], [7]. These figures show that churn can cause huge losses for companies that have lost customers. Customer churn burdens companies because it forces them to spend more effort to attract new customers [8]. Especially if the efforts and costs incurred to target new customers are not able to attract customers to subscribe. Previous research suggests that customer churn can be understood through customer satisfaction theory and customer behavior itself [9], indicating that customers are more likely to churn if they perceive a decrease in the value they receive and a decline in their overall satisfaction. In the telecommunications industry, factors such as service quality, price, and customer experience are the main determinants of whether customers decide to churn or not [10], [11]. These behavioral factors indicate that customer churn is not random but is influenced by patterns that can be measured and identified through predictive analysis. Unfortunately, conventional analysis methods and limited resources often cause companies to miss opportunities for early intervention. To overcome these limitations, data science has emerged as a field of study that combines statistical analysis, computer science, and domain knowledge to extract meaningful insights from data. This approach enables organizations to understand complex patterns and make smarter decisions.

Machine learning has been developed and implemented in this era. Machine learning is a branch of artificial intelligence (AI) that allows systems to learn from data without the need to be explicitly programmed [12]. Machine learning applications can be seen in various industries [13]–[17]. The application of machine learning for customer churn prediction has been done by several researchers before. Wu et al. [18] compared 6 algorithms, data balancing methods, and 10-fold cross-validation, and found that Logistic regression and SMOTE were able to achieve the highest accuracy of 74.82%. Amin et al. [19] used a distance-based classification approach and obtained an accuracy of 78%. Pamina et al. applied XGBoost and achieved an accuracy of 79.8%. Meanwhile, Wang et al. [20] developed a model with FCLCNN with Maj Lasso feature selection, resulting in an accuracy of 81.21%. Haddadi et al. [21] used Random Forest with two stages of resampling and achieved the highest accuracy of 82%.

Unfortunately, individual models such as Decision Tree, KNN, Naïve Bayes, and others have limitations in capturing high data complexity. Individual models tend to have biases depending on the characteristics of the algorithm and are also prone to overfitting or underfitting [22], especially when used on data that has noise or inconsistent patterns. One of the ensemble methods, stacking ensemble learning, which works by combining several models as base learners and unifying their predictions through a meta-learner model [20], is considered promising because it works by utilizing the advantages of each algorithm used to produce stronger predictions. This method is considered effective and has proven to produce good performance in several previous cases [23]–[25].

In addition, a challenge that is often encountered when building churn prediction models is data imbalance [26]. Unbalanced data between classes makes the model tend to be biased towards the majority class [27]. One widely used oversampling method is the Synthetic Minority Over-sampling Technique (SMOTE), which works by generating synthetic data based on interpolation between samples in the minority class [28]. This approach has the potential to help improve the sensitivity of the model to minority classes so that the performance of the prediction model can be optimized.

This research aims to optimize the customer churn prediction model using a stacking ensemble approach. A total of 14 algorithms were tested, and then 6 with the highest accuracy were selected. The six algorithms were then tested again with and without SMOTE to determine the best data. Furthermore, 63 stacking models were built from a combination of six algorithms as base learners and Logistic Regression as a meta learner. Evaluation was done using a confusion matrix, and the model with the highest accuracy was selected as the best model. The methods used in this study demonstrate the authenticity of this research. This study demonstrates a systematic approach to improving the accuracy of churn prediction, which can serve as a reference for further research and application in customer analysis.

2. METHOD

The research stages were carried out sequentially, starting with the data collection stage, data pre-processing, modeling, optimization, and ending with evaluation. The detailed research flow can be seen in Figure 1.

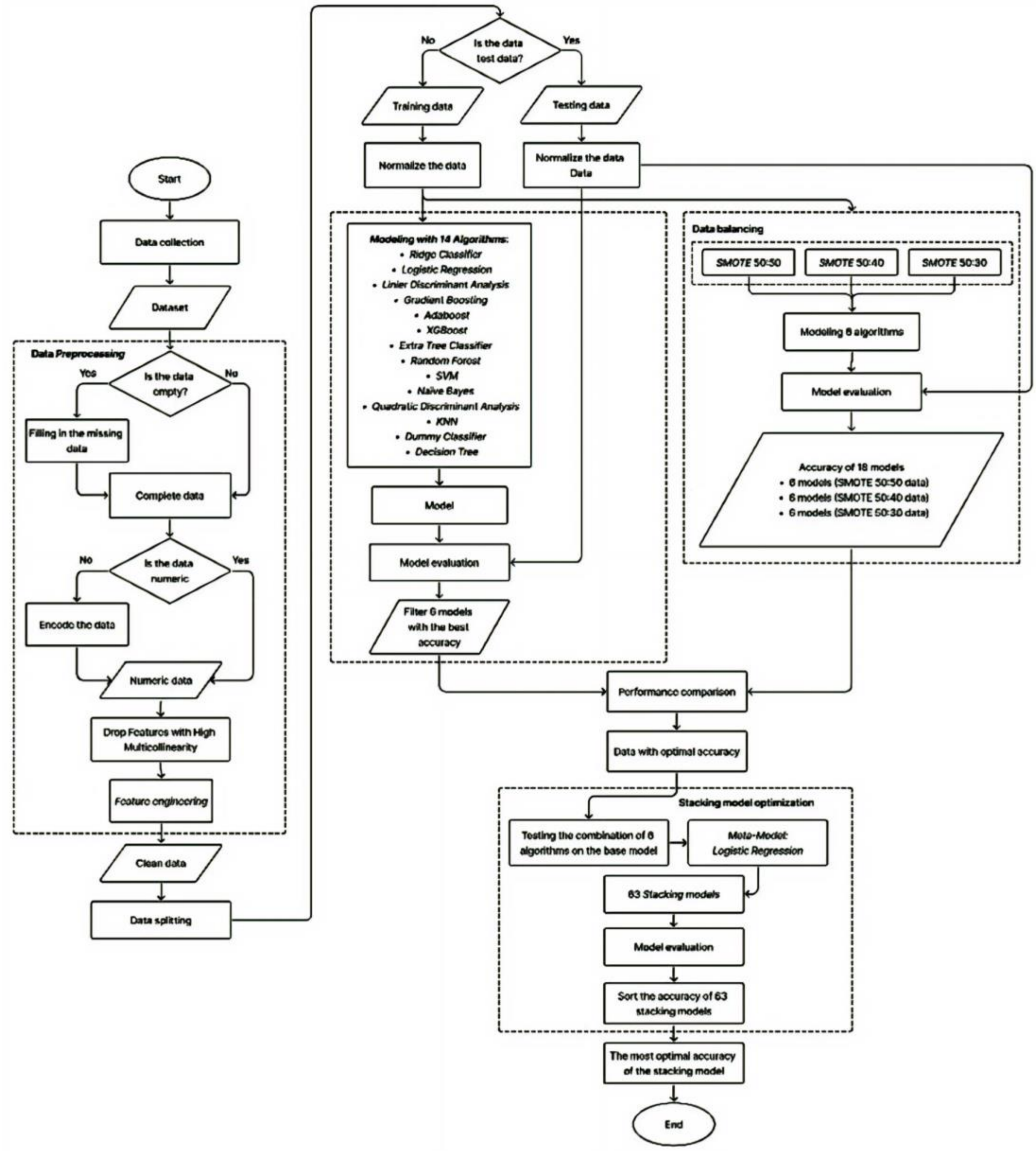

Figure 1. Research Flow of the Churn Prediction Model

2.1 Data Collection

Data is taken from the kaggle.com platform. This research uses the Telecom Customer Churn Prediction dataset, which can be accessed via the URL: https://www.kaggle.com/datasets/blastchar/telco-customer-*churn*. The dataset used in this research is a dataset that has been tested for validity, which has also been used by several previous studies, such as in research [18], [20], [21], [29]. The dataset consists of 7043 rows, with 21 features. A total of 5174 rows are non-churn customer data, and 1869 are churn customer data.

2.2 Data Preprocessing

Preprocessing is the process of preparing and cleaning data to ensure the data used is clean and ready to be trained in the modeling process. The preprocessing stage includes removing irrelevant features, handling empty data, encoding data, removing feature multicollinearity, and feature engineering. CustomerID is an irrelevant feature in customer churn prediction, because it only serves as an identifier or distinguish customer 1 from other customers, so the CustomerID feature is removed from the dataset. 11 blank data points are found in the dataset, where the blank data occurs in the total charges feature, whose tenure is 0. This indicates that the customer still does not have the total charges, because they have not paid and have not completed 1 month of subscription. Given that the data used is not too much, and missing values can reduce the quality and accuracy of the model [30], this study decided to input the same number in total charges as monthly charges. This research assumes that these customers will still pay their bills.

Machine learning models can only process numeric data [31], so it is necessary to convert data that is still of object data type into numeric data. Label encoding is applied to features that only have 2 unique values and features whose contents are based on hierarchy or order. Label encoding is applied to Phone Service, Paperless Billing, Churn, Gender, Partner, and Dependents features. While one-hot encoding is applied to features that have no order or hierarchy. One-Hot Encoding is applied to Multiple Lines, Internet Service, Online Security, Online Backup, Device Protection, Tech Support, Streaming TV, Streaming Movies, Payment Method, and Contract features.

The effect of encoding makes a feature have a very strong correlation with other features. This high correlation will only provide a computational burden without providing more information to the model, because it contains redundant information. In addition to increasing computational burden, high correlation between features or so-called feature multicollinearity will make the model less stable and reduce model interpretability [32]. Therefore, the feature multicollinearity shown in the correlation heatmap using Pearson's correlation is removed.

Feature engineering is done by creating new features from existing features to explore and enrich the information available in the dataset to help the model recognize more complex patterns [33]. This research creates new features based on insights gained from analyzing feature combinations. There is a tendency for churn to occur in conditions of tenure and monthly charges at certain values, so new features are created to help the model distinguish churn and non-churn customers.

2.3. Data Splitting

The data is divided into 2 main parts, namely training data (for training) and testing data (for testing the model). Data division is done using the train_test_split function. Where the percentage of division is 80% and 20%.

2.4. Modelling

After the data has gone through the preprocessing and division stages, the process continues with model training using the stacking ensemble method. Stacking ensemble is a concept built to minimize generalization errors in machine learning [34]. The concept works by training several basic models on the same dataset, and then the prediction results of each basic model are used as input to be re-modeled in the meta model [35]–[37]. This method effectively combines the advantages of each model and is proven to be better than a single model [38]–[40].

This research uses AdaBoost, ridge classifier, and Logistic Regression as the base learners, and Logistic Regression as the meta learner, as it is known to be simple yet powerful. This algorithm is also often chosen as a meta-learner model in previous implementations of stacking ensemble learning [41], [42]. The 3 algorithm models in the base learner are used together because, in addition to excelling in individual modeling, the algorithms are also considered to be complementary. Stacking model training is done by applying the cross-validation method, which is a method to divide the training data into several groups (folds) to ensure that the model does not only learn from a small portion of the data, but is also able to generalize well [43]. The workings of each algorithm in the base learner are as follows.

2.4.1 AdaBoost

AdaBoost is an ensemble algorithm that is known to perform well in some cases. It works by building a decision tree stump (simple decision tree) and correcting the errors of the previous model by giving more weight to the incorrectly predicted data. It starts by giving equal weight to all training data, then the weak learner (decision tree stump) is trained with these weights. The error rate is then calculated to determine how much the weak learner contributes to the final prediction. Models with low error rates will receive greater weight. Meanwhile, samples that are incorrectly predicted will have their weights increased. This is done so that the next weak learner will focus more on data that is difficult to predict [44]. The process is performed iteratively by continuously correcting the errors of the previous model. The final prediction is obtained by combining the output of all weak learners based on their respective contribution weights. In this way, the AdaBoost model can produce good prediction performance [45].

2.4.2 Logistic Regression

Logistic regression is a well-known robust algorithm [46] and has been widely used in various classification cases [47]. Logistic regression is a classification algorithm that works by calculating the probability value of the possibility of a class. Unlike linear regression, which predicts continuous values, logistic regression is a classification algorithm that can predict class categories in binary and multiclass classification problems. Logistic Regression works by calculating a linear combination of input features as in equation 1.

$$z = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n \tag{1}$$

The result of z or the calculation of the linear combination of features in the dataset is then entered into the sigmoid function. Where the sigmoid function can be seen in equation 2.

$$\sigma\,(z) = \frac{1}{1+\,e^{-z}} \tag{2}$$

This sigmoid function will convert the z value into a probability value from 0 to 1. The most common limitation used in logistic regression is, when the resulting probability is less than 0.5, then the sample will be classified as class 0, or in this case, non-churn. Conversely, when the resulting probability is greater than or equal to 0.5, then the sample is predicted as class 1 [48], or in this case, is a churn person.

2.4.3 Ridge Classifier

Ridge Classifier is a version of the logistic regression algorithm that uses a safety belt to prevent overfitting. This safety belt is in the form of L2 regularization, which is a technique that adds a penalty to the magnitude of the coefficients in the model [49]. The purpose of L2 regularization is to restrain the model from following the data to the extreme, especially if the data has many features and there is feature multicollinearity. L2 regularization in the Ridge Classifier algorithm can help the model to be resistant to noise and provide stable performance.

2.5 Evaluation Model

Model evaluation was conducted using the confusion matrix, which consists of four main components. True Positive (TP), which shows the number of correctly predicted data, where the model predicts as a positive class and in the actual data is also a positive class. True Negative (TN) also shows the number of correctly predicted data, which indicates the number of samples where the model predicts negative and the actual data is negative. False Positive (FP) shows the number of samples that the model predicts as positive, but the actual samples are negative. And False Negative (FN) indicates the number of samples when the model predicts negative, but they are positive. From these 4 components, evaluation metrics such as accuracy, precision, recall, and F1-score can be calculated.

1. Accuracy

Accuracy is a metric that shows a measure of how many model predictions are correct overall. This metric can be calculated as in equation 3.

$$Acuuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{3}$$

2. Precision

Precision measures how many positive predictions the model makes that are positive in the actual data are positive. This metric can be calculated as in equation 4.

$$Precision = \frac{TP}{TP+FP} \tag{4}$$

3. Recall

Recall is a metric that measures how many positive samples are successfully recognized by the model. This metric can be calculated as in equation 5.

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

4. F1-Score

F1-score is the harmonic mean of the precision and recall metrics, which measures the balance between the two. This metric can be calculated as in equation 6.

$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (6)$$

3. RESULT AND DISCUSSION

The preprocessing stages are carried out sequentially. The 11 missing values in the total charge feature are filled with the same value as the monthly charge, with the assumption that customers whose tenure is 0 (not yet subscribed for 1 month) will still pay their bills. The result of this step makes the data used complete.

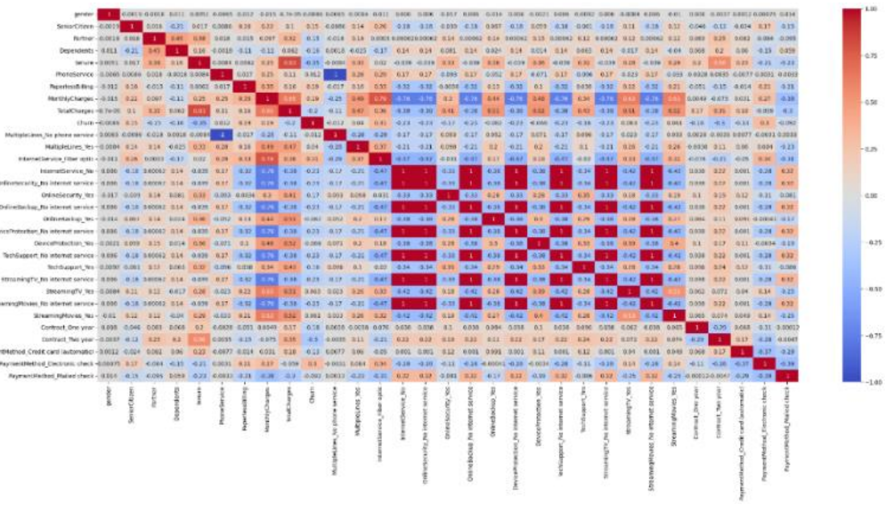

Figure 2. Dataset Heatmap Correlation

Label encoding and One-hot encoding are applied to convert categorical data into numerical data. Label encoding is used for features that have a hierarchy and only have 2 unique values. For example, the “Churn” feature, where ‘Yes’ will be converted to 1, while “No” will be converted to 0. One-hot encoding is applied to features that have no hierarchy and more than 2 unique values. The way it works is that each row of data will be assigned a value of 1 in the column that corresponds to its category, and 0 in the other columns. For example, the “Internet Service” feature has 3 categories, namely ‘DSL’, “Fiber Optic”, and “No”. 3 columns will be created, and if the original value in the row is “DSL”, then the encoding result is Internet Service_DSL = 1, Internet Service_Fiber optic = 0, and Internet Service_No = 0. Checking the multicollinearity of features (features that have a very high correlation) using the Pearson correlation method, which can then be seen in the correlation heatmap as shown in Figure 2.

The heatmap in Figure 2 shows the correlation between features through colors and numbers. Dark red indicates a strong positive correlation, while dark blue indicates a strong negative correlation. The correlation values range from -1 to 1, where values close to 1 indicate the features are very similar or even identical, which is called high multicollinearity. Such features are considered redundant, can increase computational load, and decrease model stability, so they are removed. After deletion, the number of features was reduced, and correlations were rechecked through Figure 3 to ensure that there were no longer excessively high correlations. Figure 3 shows that the dataset no longer contains features with very high correlation to other features. So the multicollinearity of features can be said to have been minimized. In the data analysis, this study found interesting insights from the scatterplot displayed from the combination of Tenure and monthly charge features. The scatterplot analysis can be seen in Figure 4.

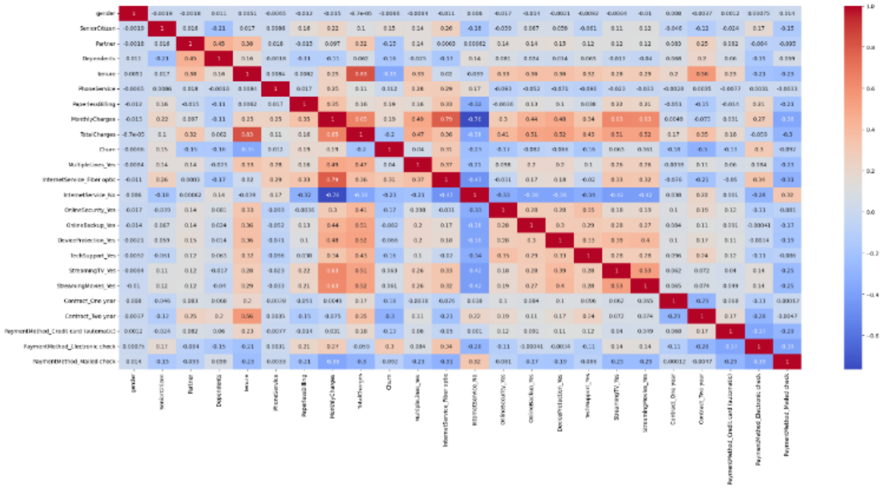

Figure 3: Correlation Heatmap After Feature Removal

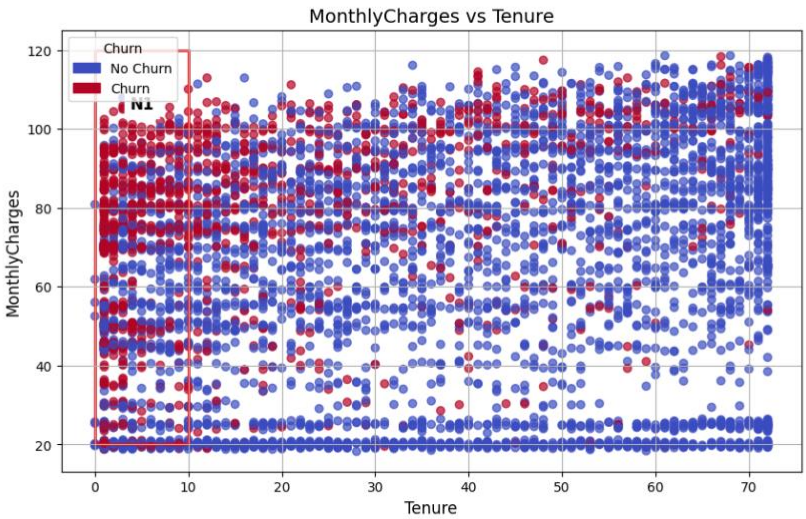

Figure 3. Scatter Plot of Tenure and Monthly Charges Features

Figure 4 shows the tendency of customers to churn at low tenure values and high monthly charges. To determine the appropriate threshold, three combinations of tenure and monthly charges conditions were tested on the X_train data. The first condition: tenure 0-10 and monthly charges 0-120; second: tenure 0-20 and monthly charges 65-120; third: tenure 0-40 and monthly charges 65-120. The results show that the percentage of churn is 48.98%, 60.48%, and 50.56% respectively, with the second condition having the highest churn rate. Based on these findings, a new feature was created that takes a value of 1 if the customer meets the second condition, and 0 otherwise. This feature was added to help the model recognize churn patterns, bringing the total features to 25 with 7043 complete data. After preprocessing is complete, the next step is modeling. To build a robust stacking model, 13 different algorithms were tested. The preliminary accuracy results are shown in Table 1

Tabel 1. Individual model results

Model	Train Accuracy	Test Accuracy
Ridge Classifier	79.62	82.19
Logistic Regression	80.30	82.11
Linier Discriminant Analysis	79.77	81.90
Gradient Boosting	85.37	81.41
AdaBoost	80.07	81.26
XGBoost	84.70	81.12

Model	Train Accuracy	Test Accuracy
Extra Tree Classifier	78.90	80.48
Random Forest	79.00	79.77
SVM	77.44	79.35
Naïve Bayes	75.47	77.08
Quadratic Discriminant Analysis	76.29	76.22
KNN	84.65	75.23
Dummy Classifier	73.45	73.53
Decision Tree	99.86	72.39

The 14 algorithms tested demonstrated varying performance, despite using the same data. Based on the accuracy in Table 1, some of the algorithms with the best results are Logistic Regression, Ridge Classifier, LDA, Gradient Boosting, and AdaBoost. To improve accuracy, oversampling is performed using the SMOTE method because the data is not balanced between churn and non-churn classes. SMOTE balances the data by creating new samples from the minority data using interpolation between nearest neighbors. Oversampling was only applied to the training data to prevent data leakage. Three class proportion scenarios were applied: 50:50, 50:40, and 50:30, to test the best balance between the amount of synthetic data and the quality of model learning. The accuracy results of modeling with oversampled data are shown in Table 2.

Tabel 2. Model performance with smote scenarios

Model	50:50		50:40		50:30	
	Train Acc	Test Acc	Train Acc	Test Acc	Train Acc	Test Acc
Gradient Boosting	87.06	78.85	78.05	80.27	78.28	81.69
XGBoost	86.66	78.64	78.05	79.49	76.92	81.55
QDA	82.46	78.42	86.20	79.42	76.25	81.05
SVM	81.14	77.93	86.65	78.85	80.11	80.06
LogReg	81.33	77.93	80.25	78.78	85.05	79.91
Naïve Bayes	79.11	77.79	80.25	78.64	80.17	79.91
Ridge	80.91	77.36	79.30	78.57	80.15	79.84
LDA	80.92	77.36	80.95	78.50	85.42	79.70
RF	80.41	77.29	80.49	78.50	79.17	79.63
ADA	80.64	77.29	80.42	78.21	79.52	78.57
ETC	80.11	77.22	77.47	77.50	76.78	76.79
Dummy Classifier	50.00	73.53	55.56	73.53	86.30	74.80
KNN	86.89	72.68	86.49	73.24	62.51	73.53
DT	99.90	71.54	99.89	72.60	99.88	73.31

Table 2 shows that the accuracy of the 14 algorithms changes significantly depending on the proportion of data balancing. The accuracy tends to be higher when the amount of oversampling data is less, indicating that overuse of SMOTE is not always effective. Some models, such as Random Forest and AdaBoost, achieved accuracies of around 81%, but still lower than the Ridge Classifier without SMOTE, which reached 82.18%. This means that some models perform better without oversampling.

Based on these results, the next experiment used the original data without SMOTE, applying the stacking ensemble method. The six best algorithms from the initial test-Logistic Regression, Ridge Classifier, Extra Tree, AdaBoost, Gradient Boosting, and LDA-were used as the base learner, with Logistic Regression as the meta learner. Combinations of various algorithms were performed using the combinations function of the Python itertools library, resulting in 63 stacking models with varying numbers and types of base learners. The performance of each stacking model is shown in Table 3.

Tabel 3. Top five best performing stacking ensemble models

Model	Acc	AUC	Rec	Precission	F1-Score
[LogReg, Ridge, AdaBoost]	82.97	86.31	59.79	71.25	65.01
[LogReg, Ridge, ETC, AdaBoost]	82.90	86.31	59.79	71.02	64.92
[LogReg, Ridge, AdaBoost, LDA]	82.90	86.34	59.79	71.02	64.92
[LogReg, ETC, AdaBoost]	82.82	86.31	59.25	71.06	64.62
[LogReg, Ridge, ETC, AdaBoost, LDA]	82.82	86.37	59.52	70.93	64.72

The five stacking models with the best accuracy are shown in Table 3. The results show that high accuracy does not always depend on the number of algorithms used. Simple combinations such as

AdaBoost, Ridge Classifier, and Logistic Regression actually gave the best results with 82.96% accuracy. The right combination of algorithms that do not interfere with each other can produce optimal performance. The use of Logistic Regression as a meta learner also plays an important role because it is stable and able to process the output of the base learner well.

The stacking model developed in this study can be used as a decision-making tool in customer retention efforts. More accurate and early identification of churn can enable companies to allocate marketing budgets and retention programs more efficiently, such as providing personalized offers, improving services, or offering loyalty programs for customers who are at higher risk of churn. The data-driven approach and ensemble-based prediction model optimization used in this study enable companies to shift from intuitive strategies to evidence-based (data-driven) strategies.

4. CONCLUSION

Customer churn presents a significant challenge in the telecommunications industry, directly affecting revenue streams. This study developed a churn prediction model using an ensemble stacking approach, leveraging the Telco Churn dataset. The methodology involved preprocessing, feature engineering, handling unbalanced data, testing 14 algorithms, and selecting the best six. The results indicated that the application of SMOTE did not enhance model performance, leading to the testing of 63 stacking models using the original data. The optimal combination was achieved with AdaBoost, Logistic Regression, and Ridge Classifier, resulting in an accuracy of 82.97%. This research demonstrates that selecting an appropriate combination of algorithms has a more significant impact on model performance than the number of models utilized. This study only tested one dataset, so generalization to other telecommunications datasets is still limited. Therefore, it is recommended that future studies test the stacking ensemble approach on larger and more diverse datasets to optimize model generalization.

ACKNOWLEDGEMENTS

We would like to thank the Artificial Intelligence and Data Mining Center (AIDMC) for providing guidance, direction, and support throughout the writing process as well as all the authors involved in ensuring the completion of this work through their diverse knowledge and expertise.

REFERENCES

[1] Y. Chen, R. Calabrese, and B. Martin-Barragan, “Interpretable machine learning for imbalanced credit scoring datasets,” *Eur. J. Oper. Res.*, vol. 312, no. 1, pp. 357–372, 2024.
[2] A. Amin, F. Al-Obeidat, B. Shah, A. Adnan, J. Loo, and S. Anwar, “Customer churn prediction in telecommunication industry using data certainty,” *J. Bus. Res.*, vol. 94, pp. 290–301, 2019.
[3] B. Prabadevi, R. Shalini, and B. R. Kavitha, “Customer churning analysis using machine learning algorithms,” *Int. J. Intell. Networks*, vol. 4, pp. 145–154, 2023.
[4] R. Rofik, J. Unjung, and B. Prasetiyo, “Enhancing Customer Churn Prediction with Stacking Ensemble and Stratified K-Fold,” *Bull. Electr. Eng. Informatics*, vol. 14, no. 1, pp. 398–408, Feb. 2025.
[5] M. Reppmann, S. Harms, L. M. Edinger-Schons, and J. N. Foege, “Activating the Sustainable Consumer: The Role of Customer Involvement in Corporate Sustainability,” *J. Acad. Mark. Sci.*, vol. 53, no. 2, pp. 310–340, Mar. 2025.
[6] A. Amin, F. Al-Obeidat, B. Shah, A. Adnan, J. Loo, and S. Anwar, “Cross-Company Customer Churn Prediction in Telecommunication: A Comparison of Data Transformation Methods,” *Int. J. Inf. Manage.*, vol. 46, pp. 304–319, Jun. 2019.
[7] Y. Li, B. Hou, Y. Wu, D. Zhao, A. Xie, and P. Zou, “Giant Fight: Customer Churn Prediction in Traditional Broadcast Industry,” *J. Bus. Res.*, vol. 131, pp. 630–639, Jul. 2021.
[8] E. Sivasankar and J. Vijaya, “Hybrid PPFCM-ANN Model: An Efficient System for Customer Churn Prediction through Probabilistic Possibilistic Fuzzy Clustering and Artificial Neural Network,” *Neural Comput. Appl.*, vol. 31, no. 11, pp. 7181–7200, Nov. 2019.
[9] S. M. Keaveney, “Customer Switching Behavior in Service Industries: An Exploratory Study,” *J. Mark.*, vol. 59, no. 2, 1995.
[10] J. H. Ahn, S. P. Han, and Y. S. Lee, “Customer churn analysis: Churn determinants and mediation effects of partial defection in the Korean mobile telecommunications service industry,” *Telecomm. Policy*, vol. 30, no. 10–11, pp. 552–568, 2006.
[11] A. Al-Refaie, M. Al-Tarawneh, N. Bata, and others, “Study Of Customer Churn In The Telecom Industry Using Structural Equation Modelling,” *Econ. Bus. J.*, vol. 12, no. 1, pp. 393–411, 2018.
[12] A. Alam, “Define Machine Learning and Describe the Main Types of Machine Learning.” Aug-2023.
[13] Y. Chen, R. Calabrese, and B. Martin-Barragan, “Interpretable machine learning for imbalanced credit scoring datasets,” *Eur. J. Oper. Res.*, no. xxxx, 2023.

[14] D. K. Sharma, S. Lohana, S. Arora, A. Dixit, M. Tiwari, and T. Tiwari, "E-Commerce product comparison portal for classification of customer data based on data mining," *Mater. Today Proc.*, vol. 51, pp. 166–171, 2021.

[15] A. Ishaq *et al.*, "Improving the Prediction of Heart Failure Patients' Survival Using SMOTE and Effective Data Mining Techniques," *IEEE Access*, vol. 9, pp. 39707–39716, 2021.

[16] B. Prasetiyo, Alamsyah, M. A. Muslim, Subhan, and N. Baroroh, "Artificial Neural Network Model for Bankruptcy Prediction," *J. Phys. Conf. Ser.*, vol. 1567, no. 3, p. 32022, Jun. 2020.

[17] Y. Dasril, M. A. Muslim, M. F. Al Hakim, Jumanto, and B. Prasetiyo, "Credit Risk Assessment in P2P Lending Using LightGBM and Particle Swarm Optimization," *Regist. J. Ilm. Teknol. Sist. Inf.*, vol. 9, no. 1, pp. 18–28, Feb. 2023.

[18] S. Wu, W.-C. Yau, T.-S. Ong, and S.-C. Chong, "Integrated Churn Prediction and Customer Segmentation Framework for Telco Business," *IEEE Access*, vol. 9, pp. 62118–62136, 2021.

[19] A. Amin, F. Al-Obeidat, B. Shah, A. Adnan, J. Loo, and S. Anwar, "Customer churn prediction in telecommunication industry under uncertain situation," *J. Bus. Res.*, 2018.

[20] C. Wang, C. Rao, F. Hu, X. Xiao, and M. Goh, "Risk Assessment of Customer Churn in Telco Using FCLCNN-LSTM Model," *Expert Syst. Appl.*, vol. 248, p. 123352, Jan. 2024.

[21] S. J. Haddadi, A. Farshidvard, F. dos S. Silva, J. C. dos Reis, and M. da Silva Reis, "Customer Churn Prediction in Imbalanced Datasets with Resampling Methods: A Comparative Study," *Expert Syst. Appl.*, vol. 246, p. 123086, Sep. 2024.

[22] S. D. Team, "6. Underfitting and Overfitting." .

[23] S. Guo, H. He, and X. Huang, "A Multi-Stage Self-Adaptive Classifier Ensemble Model With Application in Credit Scoring," *IEEE Access*, vol. 7, pp. 78549–78559, 2019.

[24] R. Jayapermana, A. Aradea, and N. I. Kurniati, "Implementation of Stacking Ensemble Classifier for Multi-class Classification of COVID-19 Vaccines Topics on Twitter," *Sci. J. Informatics*, vol. 9, no. 1, pp. 8–15, 2022.

[25] X. Yin, Q. Liu, Y. Pan, X. Huang, J. Wu, and X. Wang, "Strength of Stacking Technique of Ensemble Learning in Rockburst Prediction with Imbalanced Data: Comparison of Eight Single and Ensemble Models," *Nat. Resour. Res.*, vol. 30, no. 2, pp. 1795–1815, 2021.

[26] R. Rofik and J. Unjung, "Evaluation of Ridge Classifier and Logistic Regression for Customer Churn Prediction on Imbalanced Telecommunication Data," *Sci. J. Informatics*, vol. 12, no. 2, pp. 311–326, 2025.

[27] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A Survey on Bias and Fairness in Machine Learning," *ACM Comput. Surv.*, vol. 54, no. 6, 2021.

[28] H. Guan, L. Zhao, X. Dong, and C. Chen, "Extended natural neighborhood for SMOTE and its variants in imbalanced classification," *Eng. Appl. Artif. Intell.*, vol. 124, no. March, p. 106570, 2023.

[29] Y. Ortakci and H. Seker, "Optimising Customer Retention: An AI-driven Personalised Pricing Approach," *Comput. \& Ind. Eng.*, vol. 188, p. 109920, 2024.

[30] N. G. Ramadhan, "Comparative Analysis of ADASYN-SVM and SMOTE-SVM Methods on the Detection of Type 2 Diabetes Mellitus," *Sci. J. Informatics*, vol. 8, no. 2, pp. 276–282, 2021.

[31] Y. Yustikasari, H. Mubarok, and R. Rianto, "Comparative Analysis Performance of K-Nearest Neighbor Algorithm and Adaptive Boosting on the Prediction of Non-Cash Food Aid Recipients," *Sci. J. Informatics*, vol. 9, no. 2, pp. 205–217, 2022.

[32] K. I. Sundus, B. H. Hammo, M. B. Al-Zoubi, and A. Al-Omari, "Solving the Multicollinearity Problem to Improve the Stability of Machine Learning Algorithms Applied to a Fully Annotated Breast Cancer Dataset," *Informatics Med. Unlocked*, vol. 33, p. 101088, Sep. 2022.

[33] V. Lugat, "Pima Indians Diabetes - EDA & Prediction (0.906)," 2019. .

[34] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, p. 241259, 1992.

[35] J. Martinez-gil, "Machine Learning with Applications," *Mach. Learn. with Appl.*, vol. 10, no. October, p. 100423, 2022.

[36] E. K. Sahin and S. Demir, "Greedy-AutoML: A novel greedy-based stacking ensemble learning framework for assessing soil liquefaction potential," *Eng. Appl. Artif. Intell.*, vol. 119, no. December 2022, p. 105732, 2023.

[37] J. Unjung and others, "Soft Voting Ensemble Model to Improve Parkinson's Disease Prediction with SMOTE," *Int. J. Adv. Intell. Informatics*, vol. 11, no. 1, p. 120, Feb. 2025.

[38] W. Yin, B. Kirkulak-Uludag, D. Zhu, and Z. Zhou, "Stacking ensemble method for personal credit risk assessment in Peer-to-Peer lending," *Appl. Soft Comput.*, vol. 142, p. 110302, 2023.

[39] M. A. Muslim *et al.*, "New model combination meta-learner to improve accuracy prediction P2P lending with stacking ensemble learning," *Intell. Syst. with Appl.*, vol. 18, no. December 2022, p. 200204, 2023.

[40] Y. Yang, L. Wei, Y. Hu, Y. Wu, L. Hu, and S. Nie, "Classification of Parkinson's disease based on multi-modal features and stacking ensemble learning," *J. Neurosci. Methods*, vol. 350, no. May 2020, p. 109019, 2021.

[41] F. A. Rafrastara, C. Supriyanto, C. Paramita, and Y. P. Astuti, "Deteksi Malware Menggunakan Metode Stacking Berbasis Ensemble," *J. Inform. J. Pengemb. IT*, vol. 8, no. 1, pp. 11–16, 2023.

[42] C. A. Hafsath and A. S. Jereesh, "A Stacked Ensemble Approach for Enhancing Anti Cancer Drug Synergy Prediction," *Procedia Comput. Sci.*, vol. 235, pp. 2567–2576, 2024.

[43] T. Yan, S. L. Shen, A. Zhou, and X. Chen, "Prediction of geological characteristics from shield operational parameters by integrating grid search and K-fold cross validation into stacking classification algorithm," *J. Rock Mech. Geotech. Eng.*, vol. 14, no. 4, pp. 1292–1303, 2022.

[44] P. Lalwani, M. K. Mishra, J. S. Chadha, and P. Sethi, "Customer Churn Prediction System: A Machine Learning Approach," *Computing*, vol. 104, no. 2, pp. 271–294, 2022.

[45] W. Shan, D. Li, S. Liu, M. Song, S. Xiao, and H. Zhang, "A Random Feature Mapping Method Based on the AdaBoost Algorithm and Results Fusion for Enhancing Classification Performance," *Expert Syst. Appl.*, vol. 256, p. 124902, Jul. 2024.

[46] T. S. Lee, C. C. Chiu, Y. C. Chou, and C. J. Lu, "Mining the Customer Credit Using Classification and Regression Tree and Multivariate Adaptive Regression Splines," *Comput. Stat. \& Data Anal.*, vol. 50, no. 4, pp. 1113–1130, 2006.

[47] G. Nie, W. Rowe, L. Zhang, Y. Tian, and Y. Shi, "Credit Card Churn Forecasting by Logistic Regression and Decision Tree," *Expert Syst. Appl.*, vol. 38, no. 12, pp. 15273–15285, 2011.

[48] H. Jain, A. Khunteta, and S. Srivastava, "Churn Prediction in Telecommunication Using Logistic Regression and Logit Boost," in *Procedia Computer Science (Proceedings of ICCCA 2016?)*, 2020, vol. 167, pp. 101–112.

[49] A. Singh, B. S. Prakash, and K. Chandrasekaran, "A comparison of linear discriminant analysis and ridge classifier on Twitter data," *Proceeding - IEEE Int. Conf. Comput. Commun. Autom. ICCCA 2016*, pp. 133–138, 2017.