JSI : Jurnal Sistem Informasi *(E-Journal), VOL. 17, NO. 1, April 2025*
ISSN Print : 2085-1588
ISSN Online : 2355-4614
LINK: https://jsi.ejournal.unsri.ac.id/index.php/jsi/index

Implementation of Random Forest Method for Customer Churn Classification

Dian Kurniasari *[1], Lutfia Humairosi[2], Warsono[3], Notiragayu[4]
[1,2,3,4] Department of Mathematics, Faculty Mathematics and Natural Sciences, Universitas Lampung
e-mail: [1]dian.kurniasari@fmipa.unila.ac.id, [2]lutfia.humairosi.lh@gmail.com,
[3]warsono.1963@fmipa.unila.ac.id, [4]notiragayu@fmipa.unila.ac.id

Abstract

Annually, the banking sector consistently undergoes substantial expansion, as demonstrated by the escalating quantity of banks. Nevertheless, this expansion has led to escalating rivalry among banks as they strive to offer superior service to consumers, ultimately impacting customer migration across organizations. Customer churn, or attrition, substantially influences a company's financial performance. Hence, it is crucial to discern the conduct of clients who can discontinue their association with the organization. Precise identification is essential to gather the necessary information for the organization to retain clients and decrease churn rates. An effective strategy for addressing this issue is categorizing client behaviour using historical data. The study utilized the Random Forest approach, employing a 90% training data and 10% testing data ratio. The hyperparameter tuning findings indicate that the optimal parameter combination for constructing a Random Forest model is 400 n_estimators and 40 max_depth. The Synthetic Minority Over-Sampling Technique (SMOTE) mitigates data during categorization. The evaluation of the model demonstrates its exceptional performance in classifying imbalanced data, achieving an accuracy of 90.83%, precision of 89.29%, recall of 92.07%, and f1-score of 90.66%.

Keywords*: Churn Customer, Machine Learning, Random Forest, SMOTE*

1. INTRODUCTION

Annually, the banking sector consistently undergoes substantial expansion, as demonstrated by the escalating quantity of banks. Nevertheless, this expansion has led to escalating rivalry among banks as they strive to offer superior service to consumers, influencing customer migration between organizations. Pamina et al. [1] demonstrated that customer defection from one company to another might result in customer attrition over a specific timeframe. Customer churn, also referred to as customer attrition, substantially influences. Hence, it is crucial to ascertain the conduct of clients who possess the capacity to discontinue their association with the organization. Precise identification is essential to provide the organization with the information to retain consumers and minimize churn rates. An effective strategy for addressing this issue is categorizing or classifying client behaviour using historical data.

Classification is a crucial procedure for identifying models or functions that can effectively differentiate between various classes or concepts within a dataset [2]. However, a common issue during the classification process is imbalanced data, which refers to a situation where the amount of data for each class is unevenly distributed. This issue can lead to inaccuracies in the classification of minority classes and diminish the overall accuracy of the classification results. An excellent technique to address this issue is the Synthetic Minority Over-Sampling Technique (SMOTE). SMOTE tackles the issue of imbalanced data by creating artificial data points for the minority classes. The synthetic data were generated by sampling from the nearest neighbours of the minority classes [3].

SMOTE is crucial for enhancing data balance and boosting classification accuracy in minority classes.

The C4.5 algorithm, commonly employed for decision tree construction, is the favoured option in the classification process due to its capacity to produce pertinent rules and decision tree architectures. Nevertheless, in practical application, issues of overlap frequently arise within extensive datasets, resulting in a protracted decision-making process. Furthermore, the likelihood of overfitting the model increases as the decision tree's structure becomes more complex [4]. On the other hand, the Random Forest approach constructs a sequence of tiny trees using bootstrapping techniques. Every individual tree generates a decision, subsequently combined through majority voting to establish the ultimate conclusion. The Random Forest approach is highly effective for processing massive datasets and offers superior accuracy compared to the C4.5 algorithm.

The Random Forest algorithm has demonstrated efficacy in addressing diverse challenges within the banking sector. Rustam and Saragih [5] did a study to forecast the likelihood of the Bank experiencing financial bankruptcy due to the economic crisis that affected Turkey between 1994 and 2004. A series of tests determined that the Random Forest approach achieved an accuracy rate of up to 96%. In their study, Dewani et al. [6] compared Random Forest and K-Nearest Neighbors (KNN) algorithms for credit risk analysis. The researchers utilized datasets from the UCI Learning Dataset, a widely employed data repository in scientific research. According to the investigation, the Random Forest algorithm regularly outperformed KNN with an accuracy rate of 95.4%. This discovery demonstrates that Random Forest is more capable of identifying and categorizing credit risk than KNN.

Madaan et al. [7] compared Random Forest and Decision Tree algorithms to forecast possible loan defaults. Random Forest achieved an accuracy of 80% according to the findings of the conducted comparison. However, the Decision Tree model achieves an accuracy of approximately 73%. These findings suggest that the Random Forest outperforms the Decision Tree in terms of performance. Kuyoro et al. [8] demonstrated the efficacy of Random Forest in credit assessment by analyzing 32,581 observations. The results demonstrated that the Random Forest yielded a superior output accuracy of 91% based on the Gini Index for variable selection, compared to the Decision Tree, which achieved an output accuracy of 83%.

Prior studies have examined the application of Random Forest to an evenly distributed dataset, showcasing the model's exceptional ability to effectively handle balanced data within the banking sector. Nevertheless, the banking sector frequently encounters the challenge of data imbalance. When confronted with imbalanced data in classification tasks with Random Forest, the SMOTE method is frequently employed. Random Forest is a widely used method in banking, especially for classification tasks. However, its application is restricted to the credit scoring domain, as research conducted by [9–11] indicates. Hence, this work uses the Random Forest algorithm to address the issue of customer churn categorization on imbalanced data by employing the SMOTE technique.

JSI : Jurnal Sistem Informasi *(E-Journal), VOL. 17, NO. 1, April 2025*
ISSN Print : 2085-1588
ISSN Online : 2355-4614
LINK: https://jsi.ejournal.unsri.ac.id/index.php/jsi/index

2. RESEARCH METHOD

The research process is explained visually through the research flowchart in Figure 1. This flowchart is a graphical representation that illustrates the steps to be performed in the research, from dataset preparation to model evaluation.

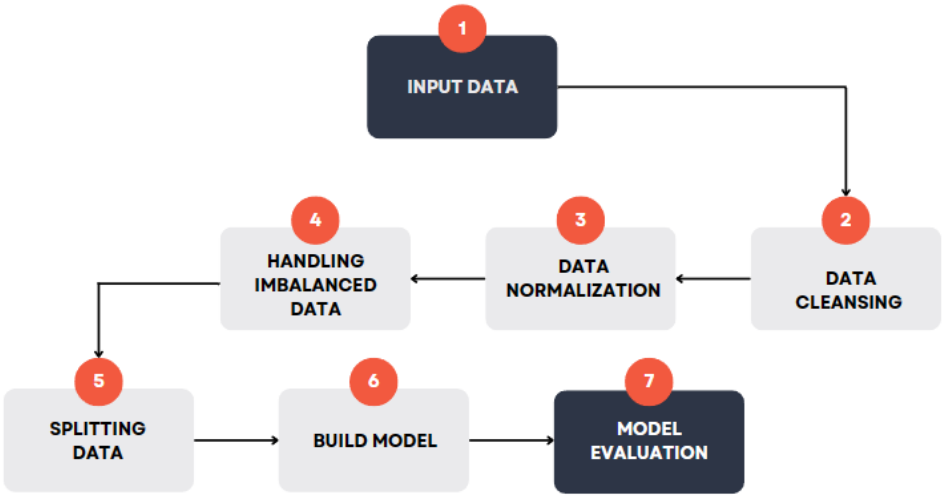

Figure 1. Research Methods

2.1. Input Data

The data utilized in this study is secondary data sourced from bank customers, acquired via https://www.kaggle.com/datasets/mathchi/churn-for-bank-customers, and downloaded in CSV format. The dataset contains a total of 10,000 bank customer records, each consisting of 14 characteristics. These variables include row number, customer ID, surname, credit score, geography, gender, age, tenure, balance, number of products, a credit card, an active number, estimated salary, and exited. An exited variable is a binary target variable with two classification classes. The first class represents customers who stop (churn) and is labelled "1". There are 2037 customers in this class. The second class represents customers who do not stop (not churn) and is labelled "0". There are 7963 customers in this class.

2.2. Data Cleansing

Data cleansing is a crucial process for ensuring data quality. It involves removing or rectifying incomplete data, resolving discrepancies, and standardizing data into a consistent format. The significance of this stage lies in the fact that any dataset, including incomplete, erroneous, inaccurate, or irrelevant data, might introduce ambiguity in the interpretation of analytic results. Data cleansing becomes more critical as the dataset size increases due to complex issues such as missing data, duplicate data, or incompatible data formats [12], [13].

2.3. Data Normalization

Normalization is a method of categorizing data properties in a model to enhance the flexibility of the data. However, normalization is crucial in minimizing ambiguity in

data. Normalization is performed to mitigate the impact of a wide range of feature values on a specific measure by employing feature scaling. Feature scaling, such as standardization or other scaling procedures, includes altering the values in data to facilitate the processing process. Applying feature scaling makes the data more manageable and accessible to analyze [14, 15].

The feature scaling technique used in this work is Z-score normalization, also known as Standard Scaler. A StandardScaler is a statistical tool that assumes the data follows a normal distribution and transforms it so that the range is centred around zero, with a standard deviation of one. This procedure entails computing each data element's average and standard deviation to avoid any data element having a disproportionately large value relative to the others. Subsequently, these elements are adjusted using the prescribed formula [16, 17]:

$$Z_{scaled} = \frac{(x - \mu)}{\sigma} \quad (1)$$

2.4. Handling Imbalanced Dataset

The solutions developed to tackle data imbalances in machine learning may be categorized into two primary approaches: algorithmic and data approaches. He et al. [18] have demonstrated the effectiveness of resampling approaches to address the issue of imbalanced data. Resampling is a data-level method. This technique offers the benefit of being independent of the classifier type, making it more adaptable [19].

In general, resampling techniques can be categorized into two primary groups: undersampling and oversampling. Research undertaken by Batista et al. [20] indicates that oversampling yields superior performance to undersampling. SMOTE is a widely recognized and commonly employed oversampling method for addressing data imbalance. SMOTE operates by augmenting minority classes using synthetic data. This synthetic data is newly generated and has been revived. The oversampling procedure involves extracting data from a minority class and introducing artificial samples along a line that connects one or more of the closest neighbours of that minority class's data. The selection of the number of nearest neighbours is made randomly. The formula for producing synthetic data using SMOTE is given by equation [21]:

$$X_{new} = X_i + \left(\widehat{X_k} - X_i\right) \times \delta \quad (2)$$

where X_{new} represent newly generated synthetic data, X_i represent data from minority classes, $\widehat{X_k}$ represent data from the k nearest neighbour with the closest distance to X_i, and δ represents a random number between 0 and 1. The Euclidean distance metric was used to measure the difference in the distance when identifying the nearest neighbour using numerical data.

2.5. Data Splitting

Data splitting refers to dividing data into two subsets with the intention of training and testing models independently. At this point, the dataset can be divided into two parts using different ratios, such as 60:40, 70:30, 80:20, and 90:10. The first portion is used to

train the model, while the second part is used to test the model's prediction abilities [22]. Additionally, it is advisable to have the training set size as large as possible to mitigate the risk of over-fitting.

Joseph and Vakayil [23] propose alternative approaches for dividing training sets into portions, including validation and standard partitioning into two data sets. Therefore, the data will be partitioned into training, validation, and test data. The primary rationale for partitioning the dataset into three distinct subsets is to mitigate potential bias from repeatedly utilizing the same validation or testing subset. However, researchers occasionally neglect the validation stage and split the data into two subsets: the training and test sets. Hence, the optimal approach to mitigate bias in a scenario where the data is partitioned into two subsets is to construct a model using training sets and fine-tune the hyper-parameters within the model [24].

2.6. Model Building

Building a proficient machine learning model is an intricate and time-consuming endeavour. This procedure entails carefully selecting an appropriate algorithm and creating an optimal model architecture by fine-tuning the hyper-parameters. Hyperparameters are variables used to customize machine learning models, such as the C parameter in Support Vector Machine (SVM) models or the learning rate in neural networks. Hyper-parameters can also specify techniques that effectively minimize loss functions, such as activation functions and optimization approaches in neural networks or kernel types in SVM [25], [26]. Conversely, the Random Forest model utilizes multiple parameters for hyperparameterization. These include n_estimators, which determines the number of trees in the model, and max_depth, which determines the maximum depth of each tree [27].

Hyper-parameter tuning refers to building an optimal model architecture with the most suitable hyper-parameter configuration [28]. Grid Search is a commonly employed method for hyper-parameter tuning. It aims to identify the optimal parameter combination from a model that can generate precise data predictions. Grid Search is a method that systematically tests different specified values for each parameter to identify the most practical combination of parameters.

2.7. Model Evaluation

Assessment metrics are crucial in evaluating the effectiveness of different machine-learning approaches, particularly in classification. A frequently employed evaluation measure is the confusion matrix. A confusion matrix is a graphical depiction of a model's performance using tables. Each row in this matrix reflects the actual categorization of the data, while each column represents the predicted categorization made by the model for that data or vice versa [29].

Table 1. Confusion matrix

		Prediction	
		Positive	**Negative**
Actual	**Positive**	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Sokolova and Lapalme [30] state that the data in Table 1 can compute the accuracy, precision, recall, and f1-score metrics. These computations are crucial for gaining a more profound comprehension of the effectiveness of the categorization model employed. Furthermore, employing these metrics may assess the model's proficiency in accurately categorizing data and gauging its ability to discover pertinent classes. The mathematical definition of the metrics accuracy, precision, recall, and f1-score is as follows:

1. Accuracy refers to the total efficiency of the classification outcomes.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (3)$$

2. Precision refers to the proportion of data labels correctly classified as positive out of all the labels assigned as positive by the classifier.

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

3. Recall refers to the classifier's ability to recognize positive labels accurately.

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

4. The F1-score is a metric that quantifies the balance between precision and recall in classification outcomes for positively labelled data.

$$F_1 = 2 \times \frac{precision\ .\ recall}{precision + recall} \quad (6)$$

3. RESULTS AND ANALYSIS

3.1. Data Visualization

Data visualization is crucial in condensing and presenting a comprehensive representation of the data under examination. The initial methodology employed in this investigation involves utilizing a Pie Chart to visually represent the proportion of client attrition, as depicted in Figure 2.

JSI : Jurnal Sistem Informasi *(E-Journal), VOL. 17, NO. 1, April 2025*
ISSN Print : 2085-1588
ISSN Online : 2355-4614
LINK: https://jsi.ejournal.unsri.ac.id/index.php/jsi/index

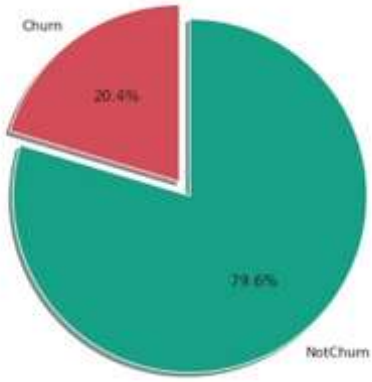

Figure 2. Visualize customer churn with a pie chart

According to Figure 2, 20.4% of 2037 clients chose to churn or go to another bank, while 79.6% or 7963 consumers chose to retain their usage of Bank services. This visualization provides insights into the percentage of customers who changed status and opted to stay loyal to the Bank. Nevertheless, additional details such as gender, geographic area, and age remain undisclosed. Hence, the subsequent action generates a Customer Churn Bar Chart, as depicted in Figure 3.

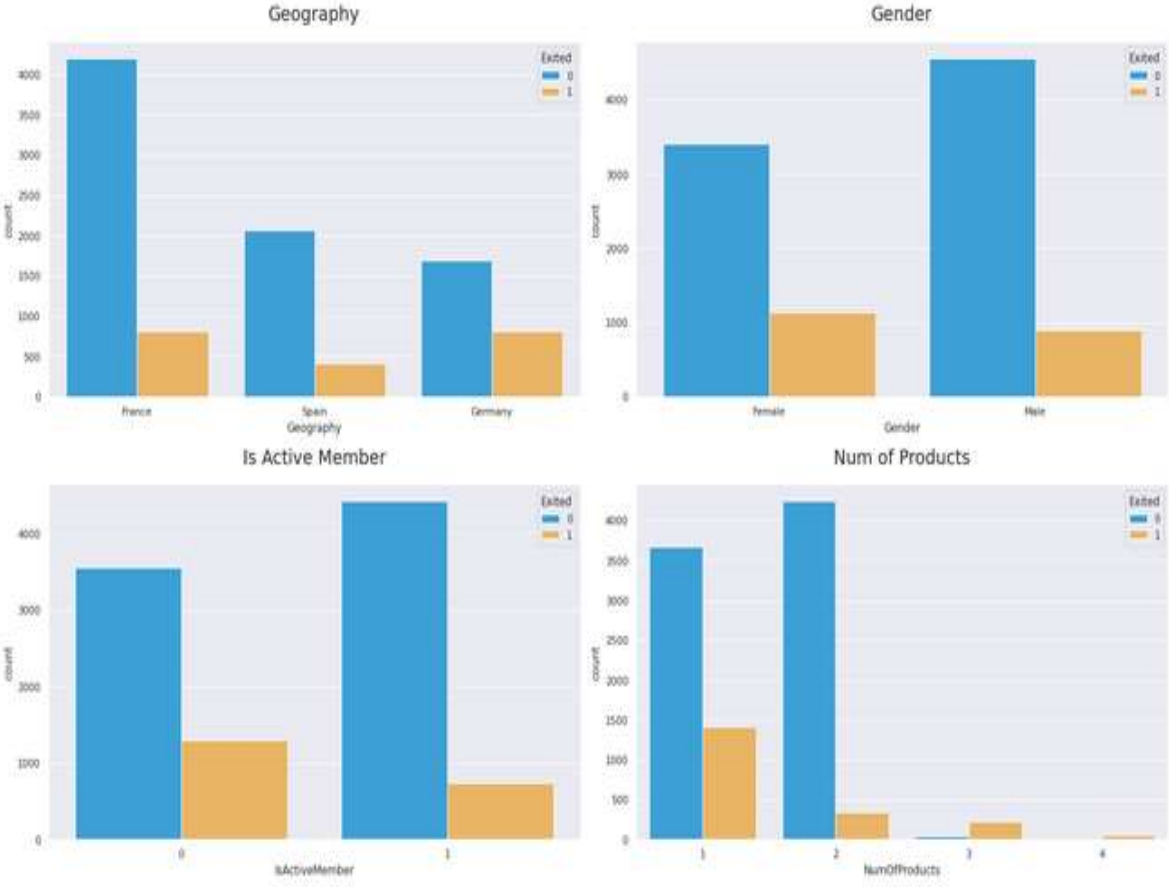

Figure 3. Customer churn bar chart

The analysis of the geography variables represented in Figure 3 reveals that customer turnover is more prevalent in France than in Germany and Spain. That indicates substantial variations in customer attrition patterns among the countries. Furthermore, the analysis of Figure 3 also demonstrated that gender significantly influences churn inclinations, with women exhibiting higher churn rates than men. In addition, consumers with limited purchases, especially those who only buy one or two products, are more likely to experience churn.

3.2 Data Preprocessing

Data preprocessing is essential in preparing raw data for machine learning algorithms. This step aims to enhance the efficiency and performance of the model that will be created. Several techniques can be used during preprocessing to ensure optimal data quality before the model's learning phase. These techniques include handling missing values, normalizing data, and coding variables.

3.2.1. Data Cleansing

Data cleansing is the initial step in the early phases of data preprocessing. During this phase, the process of verifying the presence of any missing or duplicated data is conducted. By employing syntax in Figure 4 and Figure 5, the process of identifying missing or duplicate data is conducted. The results of these checks indicate that the dataset does not contain any missing or duplicate data. Therefore, the dataset has met the necessary criteria to advance to the subsequent round of analysis or modelling.

```
#Menampilkan data hilang
data.isnull().sum()
```

Figure 4. Syntax for missing data

```
#Menampilkan data duplikat
data.duplicated().sum()
```

Figure 5. Syntax for duplicate data

3.2.2. Data Normalization

The data normalization process in this investigation involved using the StandardScaler approach. This approach was selected due to the lack of precise knowledge of the minimum and maximum values of the data. The data will be transformed using StandardScaler to achieve a mean of 0 and a standard deviation. That will aid in mitigating variations in scaling among attributes and provide uniformity in the data processing. The outcomes of data normalization are displayed in Table 2 as follows:

Table 2. Data normalization results

	CreditScore	Age	Tenure	Balance	NumOf Products	Estimated Salary
0	-0.326221	0.293517	-1.041760	-1.225848	-0.911583	0.02186
1	-0.440036	0.198164	-1.387538	0.117350	-0.911583	0.216534
2	-1.536794	0.293517	1.032908	1.333053	2.527057	0.240687
3	0.501521	0.007457	-1.387538	-1.225848	0.807737	-0.108918
....	....	....	....	....	....	....
9997	0.604988	-0.278604	0,687130	-1.2225848	-0.911583	-1.008643
9998	1.256835	0.293517	-0.695982	-0.022608	0.807737	-0.125231
9999	1.463771	-1.041433	-0.350204	0.859965	-0.911583	-1.076370

3.2.3. Handling Data Categorical

The subsequent task addresses categorical data after completing feature scaling or data normalization. Variables such as geography and gender belong to this category. Hence, a computer program must be transformed into a suitable format for comprehending and manipulating input. The one-hot encoding strategy was employed in this investigation due to the absence of any distinct hierarchy or scale in the categorical data. Subsequently, each category inside the categorical variable will be transformed into an individual binary variable, as depicted in Figure 6, in the following manner.

Geography_France	Geography_Germany	Geography_Spain	Gender_Female	Gender_Male
1	0	0	1	0
0	0	1	1	0
1	0	0	1	0
1	0	0	1	0
0	0	1	1	0

Figure 6. One hot encoding

This phase is crucial to ensure the optimal utilization of all features in the dataset throughout the subsequent analysis or modelling process.

3.3. Handling Imbalanced Data

The issue of data imbalance that develops in this study will be resolved by implementing the SMOTE approach. Before introducing SMOTE, there were 2037 data items about the customer churn category, while the remaining entries amounted to 7963, representing customers who did not churn. SMOTE is a technique that generates synthetic samples in the minority category (customer churn) to achieve a balanced number with the majority category (surviving customers). This approach is a crucial measure in mitigating the impact of data imbalance on the analysis process, which can detrimentally affect the performance of the produced model. In addition, Table 3 displays the outcomes of resampling using SMOTE.

Table 3. SMOTE

Class	Sum
1	7963
0	7963

Based on the information in Table 3, the quantity of data during the training phase has risen to 7963 for every category. The SMOTE approach has effectively achieved a balanced distribution of classes in the dataset.

3.4. Train Test Split

This study utilizes various division methods to determine the ratio of training data to test data. The data will be partitioned into a training set comprising 60% of the data and a testing set comprising 40% of the data. Secondly, the data is divided into 70% for training and 30% for testing. Third, 80% of the data should be allocated for training and 20% for testing. Ultimately, the data will be partitioned into a training set including 90% of the data and a testing set comprising the remaining 10%. The variation in applied ratios leads to varying training and test data proportions in each dataset, as outlined in Table 4.

Table 4. Train test split

Train Test Split	Data Training	Data Testing
60% and 40%	9555	6371
70% and 30%	11148	4778
80% and 20%	12740	3186
90% and 10%	14333	1593

3.5. Building a Model Random Forest

The Random Forest model has been constructed using a collection of hyperparameters that have been fine-tuned through a hyperparameter tuning process. The specific hyperparameter being referred to is stated in Table 5 as follows:

Table 5. Hyperparameter tuning

Hyperparameter	Definition	Value
n_estimators	Number of decision trees built	50, 100, 200, 400, 600
max_depth	Maximum depth of the decision tree	3, 15, 20, 40, 60

This study employed the Grid Search Cross-Validation (Grid Search CV) function from the sci-kit-learn library to fine-tune the hyperparameters. The hyperparameter tuning procedure involves selecting a range of potential values for each hyperparameter and systematically evaluating each combination using cross-validation techniques. This technique assesses the model's performance on the training data and guarantees that the final model exhibits strong generalization capabilities to previously unseen data. The following are the outcomes of the hyperparameter tweaking conducted.

Table 6. Best Parameters obtained from hyperparameter tuning

Ratio	Hyperparameter	Value
60 % and 40%	n_estimators	100
	max_depth	40
70 % and 30%	n_estimators	200
	max_depth	40
80 % and 20%	n_estimators	200
	max_depth	40
90 % and 10%	n_estimators	400
	max_depth	40

The outcome of the hyperparameter tuning process, as shown in Table 6, is a set of hyperparameters that yield the optimal performance according to preset assessment metrics, such as accuracy, precision, recall, or F1-score, for each data sharing scheme. The appropriate hyperparameters are utilized to train the final model, ensuring it achieves peak performance when applied to new data.

3.6. Model Evaluation

After the model is trained and produces a classification, these findings are subsequently evaluated with actual or testing data. The comparison is initially demonstrated using a confusion matrix, a visual aid that summarizes the accuracy of a classification model by presenting the number of correct and incorrect predictions for

each class. Figure 7 displays a graphical representation of the confusion matrix for a customer churn classification model. It illustrates how the predicted results are distributed compared to the actual data. Table 7 presents the computed evaluation metrics based on the generated confusion matrix. It offers a concise assessment of the model's performance using easily interpretable numerical values.

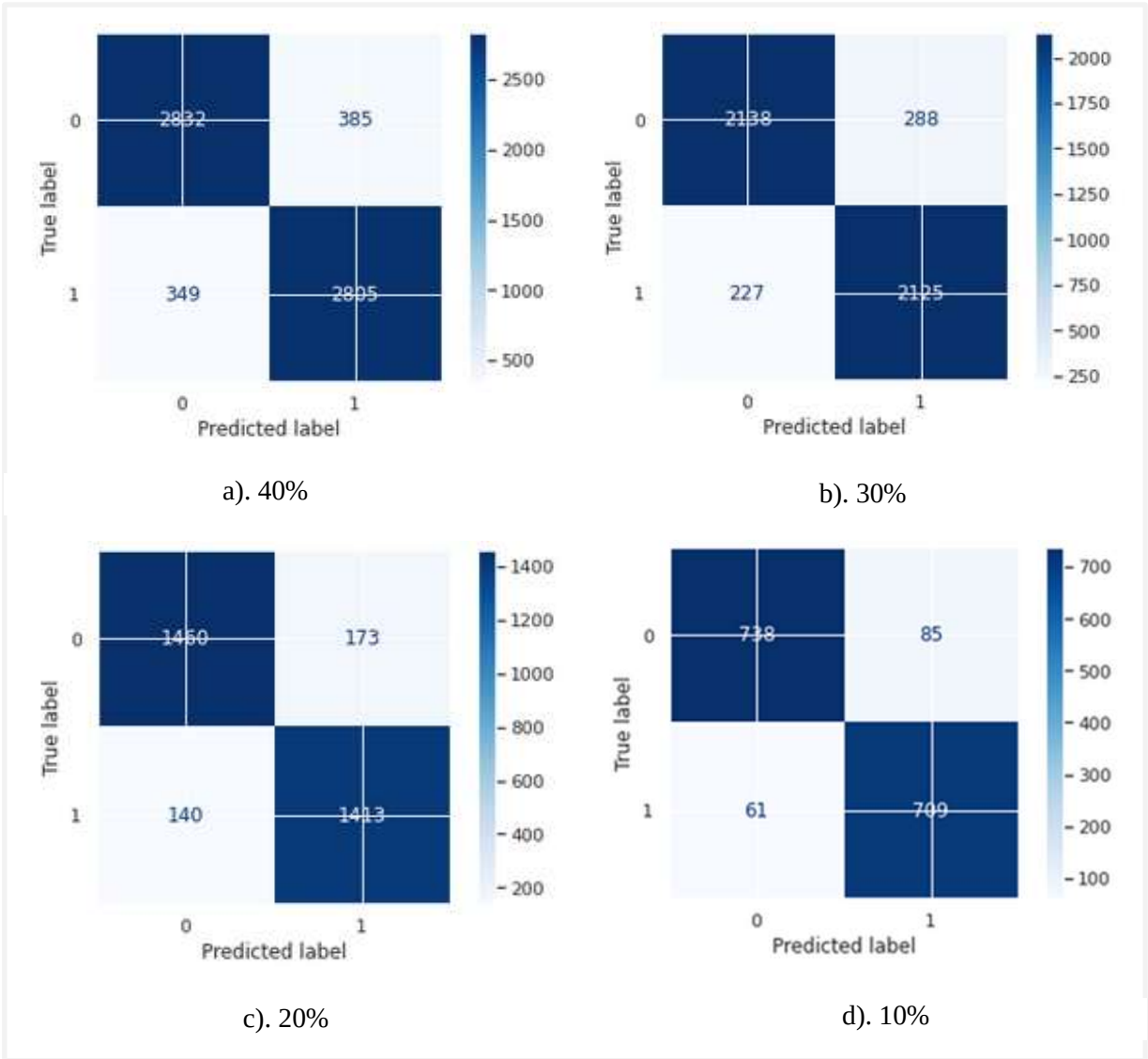

Figure 7. Confusion matrix with different data testing ratio

Table 7. Random forest model performance

Data Testing Ratio	Accuracy	Recall	Precision	F1-Score
40%	88,47%	88,93%	87,93%	90,39%
30%	89,22%	90,34%	88,06%	89,19%
20%	90,17%	90,98%	89,09%	90,02%
10%	90,83%	92,07%	89,29%	90,66%

Figure 7 and Table 7 provide an analysis that reveals changes in model performance when different data-splitting techniques are used for model evaluation. In general, it can be inferred that the performance of a model improves when more data is utilized for training, resulting in fewer data available for testing. The model achieved optimal performance on the 10% data testing data sharing scheme, exhibiting the most significant assessment metrics across all dimensions. That suggests the Random Forest model can accurately categorize churn instances when trained with a larger dataset.

Furthermore, the imbalance that occurred during the classification process was effectively resolved using the SMOTE technique. The high F1 score attained provides clear evidence of this success. The F1 score is the most optimal assessment statistic for analyzing the performance of a model on imbalanced data. The F1-score differs from accuracy because it incorporates data distribution by considering precision and recall. A higher F1 score indicates a superior model quality in categorizing imbalanced data.

4. CONCLUSION

Numerous significant conclusions can be inferred after performing a classification procedure using the Random Forest method to categorize customer attrition. Initially, we implemented the SMOTE technique to address the data imbalance issue. The outcomes demonstrated that partitioning the data into 90% for training and 10% for testing yielded the most favourable outcomes. Furthermore, by doing hyperparameter tuning, we successfully determined the ideal values for the model parameters, specifically 400 n-estimators and a maximum depth of 40. Furthermore, the utilization of SMOTE has demonstrated efficacy in enhancing the performance of the Random Forest model. The Random Forest approach in the data-splitting scheme achieved an accuracy rate of 90.83% and a f1-score of 90.66%, demonstrating the model's exceptional ability to classify imbalanced data.

REFERENCES

[1] J. Pamina *et al.*, "An effective classifier for predicting churn in telecommunication," *J. Adv. Res. Dyn. Control Syst.*, vol. 11, no. 1 Special Issue, pp. 221–229, 2019.

[2] Y.T. Utami, D. A. Shofiana, and Y. Heningtyas, “Penerapan Algoritma C4.5

Untuk Prediksi Churn Rate Pengguna Jasa Telekomunikasi," *J. Komputasi*, vol. 8, no. 2, 2020, doi: 10.23960/komputasi.v8i2.2647.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.

[4] N. Benediktus and R. S. Oetama, "Algoritma Klasifikasi Decision Tree C5.0 untuk Memprediksi Performa Akademik Siswa Natanael," *Ultim. J. Tek. Inform.*, vol. 12, no. 1, pp. 14–19, 2020, [Online]. Available: https://www.kaggle.com/aljarah/xAPI-Edu-Data

[5] Z. Rustam and G. S. Saragih, "Predicting Bank Financial Failures using Random Forest," in *2018 International Workshop on Big Data and Information Security, IWBIS 2018*, 2018, pp. 81–86. doi: 10.1109/IWBIS.2018.8471718.

[6] P. Dewani, M. Sippy, G. Punjabi, and A. Hatekar, "Credit Scoring : A Comparison between Random Forest Classifier and K-Nearest Neighbours for Credit Defaulters Prediction," *Int. Res. J. Eng. Technol.*, 2020, [Online]. Available: www.irjet.net

[7] M. Madaan, A. Kumar, C. Keshri, R. Jain, and P. Nagrath, "Loan default prediction using decision trees and random forest: A comparative study," in *IOP Conference Series: Materials Science and Engineering*, 2021. doi: 10.1088/1757-899X/1022/1/012042.

[8] A. O. Kuyoro, O. A. Ogunyolu, T. G. Ayanwola, and F. Y. Ayankoya, "Dynamic Effectiveness of Random Forest Algorithm in Financial Credit Risk Management for Improving Output Accuracy and Loan Classification Prediction," *Ing. des Syst. d'Information*, vol. 27, no. 5, pp. 815–821, 2022, doi: 10.18280/isi.270515.

[9] H. Aktar, M. A. Masud, N. J. Aunto, and S. N. Sakib, "Classification Using Random Forest on Imbalanced Credit Card Transaction Data," in *2021 3rd International Conference on Sustainable Technologies for Industry 4.0, STI 2021*, 2021. doi: 10.1109/STI53101.2021.9732553.

[10] A. H. M. Aburbeian and H. I. Ashqar, "Credit Card Fraud Detection Using Enhanced Random Forest Classifier for Imbalanced Data," in *Lecture Notes in Networks and Systems*, 2023, pp. 605–616. doi: 10.1007/978-3-031-33743-7_48.

[11] Y. Zhou, L. Shen, and L. Ballester, "A two-stage credit scoring model based on random forest: Evidence from Chinese small firms," *Int. Rev. Financ. Anal.*, vol. 89, 2023, doi: 10.1016/j.irfa.2023.102755.

[12] F. Ridzuan and W. M. N. Wan Zainon, "A review on data cleansing methods for big data," in *Procedia Computer Science*, 2019, pp. 731–738. doi: 10.1016/j.procs.2019.11.177.

[13] M. Hosseinzadeh *et al.*, "Data cleansing mechanisms and approaches for big data analytics: a systematic study," *J. Ambient Intell. Humaniz. Comput.*, vol. 14, no. 1, pp. 99–111, 2023, doi: 10.1007/s12652-021-03590-2.

[14] T. Jayalakshmi and A. Santhakumaran, "Statistical Normalization and Back Propagation for Classification," *Int. J. Comput. Theory Eng.*, pp. 89–93, 2011, doi: 10.7763/ijcte.2011.v3.288.

[15] S. Jain, S. Shukla, and R. Wadhvani, "Dynamic selection of normalization techniques using data complexity measures," *Expert Syst. Appl.*, vol. 106, pp. 252–262, 2018, doi: 10.1016/j.eswa.2018.04.008.

[16] V. N. G. Raju, K. P. Lakshmi, V. M. Jain, A. Kalidindi, and V. Padma, "Study the Influence of Normalization/Transformation process on the Accuracy of Supervised Classification," in *Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020*, 2020, pp. 729–735. doi: 10.1109/ICSSIT48917.2020.9214160.

[17] V. R. Prasetyo, M. Mercifia, A. Averina, L. Sunyoto, and B. Budiarjo, "Prediksi Rating Film Pada Website Imdb Menggunakan Metode Neural Network," *Netw. Eng. Res. Oper.*, vol. 7, no. 1, 2022, doi: 10.21107/nero.v7i1.268.

[18] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proceedings of the International Joint Conference on Neural Networks*, 2008, pp. 1322–1328. doi: 10.1109/IJCNN.2008.4633969.

[19] D. Ramyachitra and P. Manikandan, "Imbalanced Dataset Classification and Solutions: a Review," *Int. J. Comput. Bus. Res. ISSN (Online*, vol. 5, no. 4, pp. 2229–6166, 2014.

[20] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 20–29, 2004, doi: 10.1145/1007730.1007735.

[21] N. P. Y. T. Wijayanti, E. N. Kencana, and I. W. Sumarjaya, "SMOTE: Potensi dan Kekurangannya pada Survei," *E-Jurnal Mat.*, vol. 10, no. 4, 2021, doi: 10.24843/mtk.2021.v10.i04.p348.

[22] Q. H. Nguyen *et al.*, "Influence of data splitting on performance of machine learning models in prediction of shear strength of soil," *Math. Probl. Eng.*, vol. 2021, 2021, doi: 10.1155/2021/4832864.

[23] V. R. Joseph and A. Vakayil, "SPlit: An Optimal Method for Data Splitting," *Technometrics*, vol. 64, no. 2, pp. 166–176, 2022, doi: 10.1080/00401706.2021.1921037.

[24] I. O. Muraina, "Ideal Dataset Splitting Ratios in Machine Learning Algorithms: General Concerns for Data Scientists and Data Analysts," *7th Int. Mardin Artuklu Sci. Res. Conf.*, no. February, pp. 496–504, 2022.

[25] G. I. Diaz, A. Fokoue-Nkoutche, G. Nannicini, and H. Samulowitz, "An effective algorithm for hyperparameter optimization of neural networks," *IBM J. Res. Dev.*, vol. 61, no. 4, 2017, doi: 10.1147/JRD.2017.2709578.

[26] R. Elshawi, M. Maher, and S. Sakr, "Automated Machine Learning: State-of-The-

Art and Open Challenges," Jun. 2019, [Online]. Available: http://arxiv.org/abs/1906.02287

[27] M. Ataei and M. Osanloo, "Using a combination of genetic algorithm and the grid search method to determine optimum cutoff grades of multiple metal deposits," *Int. J. Surf. Mining, Reclam. Environ.*, vol. 18, no. 1, 2004, doi: 10.1076/ijsm.18.1.60.23543.

[28] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020, doi: 10.1016/j.neucom.2020.07.061.

[29] I. W. Saputro and B. W. Sari, "Uji Performa Algoritma Naïve Bayes untuk Prediksi Masa Studi Mahasiswa," *Creat. Inf. Technol. J.*, vol. 6, no. 1, 2020, doi: 10.24076/citec.2019v6i1.178.

[30] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, 2009, doi: 10.1016/j.ipm.2009.03.002.