JOIV : Int. J. Inform. Visualization, 9(3) - May 2025 1132-1141

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Optimizing Artificial Neural Network for Customer Churn: Advanced Data Balancing and Feature Selection

Aditiya Hermawan [a], Willy Wijaya [a], Benny Daniawan [b]

[a] *Department of Informatics Engineering, Faculty of Science and Technology, Buddhi Dharma University, Tangerang, Indonesia*
[b] *Department of Information Systems, Faculty of Science and Technology, Buddhi Dharma University, Tangerang, Indonesia*
Corresponding author: [*]aditiya.hermawan@ubd.ac.id

***Abstract*—Customers are valuable assets in the dynamic business world. However, service dissatisfaction often leads them to switch to competitors, a phenomenon known as customer churn. In the telecommunications industry, churn poses a significant challenge as it directly impacts revenue and influences other customers within their social networks to do the same. Consequently, predicting churn has become essential, with numerous researchers employing various methods to classify potential churners. This study builds upon prior research that utilized Artificial Neural Networks (ANN) or Deep Learning to predict churn, achieving an accuracy of 88.12%. To improve model performance, this research implements an Artificial Neural Network (ANN) as the primary algorithm, along with Random Over-Sampling (ROS) and Synthetic Minority Oversampling Technique (SMOTE) for data balancing, and three feature selection methods: Minimum Redundancy Maximum Relevance (mRMR), Lasso Regression, and XGBoost. The results demonstrate a 0.38% increase in accuracy compared to previous studies. The finding suggests opportunities for further exploration. Future studies can consider alternative feature selection techniques, such as Wrapper Methods or Heuristic/Metaheuristic approaches, which may produce more optimal feature combinations. Other data balancing methods, such as Undersampling techniques (e.g., Random Undersampling, Tomek Links) or Hybrid Methods (e.g., SMOTE combined with Tomek Links), could be explored to address imbalanced datasets effectively. These approaches are expected to provide better combinations and to improve overall prediction performance, enabling researchers to develop more robust and accurate models for customer churn prediction in subsequent studies.**

Keywords—**Customer churn; classification; ANN; SMOTE; ROS; XGBoost; LassoRegression; mRMR.**

Manuscript received 2 Sep. 2024; revised 15 Dec. 2024; accepted 5 Feb. 2025. Date of publication 31 May 2025.

I. Introduction

Customers are essential assets in the dynamic business world and competitive market. They can move to other services quickly, which is usually called customer churn [1]–[3]. For a lot of companies, especially in the telecommunications sector, customer churn is a big problem since it directly impacts company revenue [4]–[6]. In addition, customer churn also influences other customers in their social network to do the same [7]. One of the most profitable strategic plans is to retain customers to continue subscribing [8]. Maintaining the loyalty of existing customers to continue subscribing is a reasonably affordable strategy compared to getting new customers, as getting new customers costs 6 to 10 times more than retaining customers, thereby eliminating much profit [3], [4], [7], [9]–[13]. Maintaining customer loyalty can be more effective if we can predict which customers are at risk of churn; hence, it poses a challenge to determine a suitable churn prediction method. One method that has proven to be reasonably practical is machine learning techniques, which are very effective, especially for creating customer churn prediction models that can predict information based on previous data [5], [10], [14], [15].

This study aims to develop an optimized customer churn prediction model using an artificial neural network (ANN) that integrates advanced feature selection and dataset balancing techniques to achieve higher accuracy compared to previous models. Machine learning techniques have been widely employed in developing customer churn prediction models, yielding effective results. However, with the increasing volume of data, there has been a shift towards using deep learning techniques in recent years, due to their ability to manage data on a larger scale than conventional machine learning techniques. The term 'deep' in deep learning refers to using complex neural networks.

Deep learning is part of the Artificial Neural Network (ANN), whose characteristics include various layers in data

processing [16]. On the methodological level, deep learning is an extension of ANN architectures that are well known in forecasting literature since ANNs have specific layers that can be stacked on top of each other to create a "deeper" ANN [17]. Several previous studies have widely used Artificial Neural Networks to create customer churn prediction models. For example, the research for creating customer churn prediction models using Artificial Neural Networks achieved 79% accuracy [9]. Another study that confers a customer churn prediction model using deep learning obtained an accuracy of 88.12% with an Artificial Neural Network [16].

Apart from Artificial Neural Networks, there is research discussing customer churn prediction models using K-Nearest Neighbor (KNN), which gets an accuracy of 83.97%, Support Vector Machine (SVM) with an accuracy of 79.63%, Decision Tree (DT) gets 91.98%, Random Forest (RF) obtained an accuracy of 95.74% [18]. However, there are problems in the development process of customer churn prediction. Although prior studies have explored various feature selection and dataset balancing methods, the research in investigating their combined impact on ANN models for customer churn prediction is limited. This gap further highlights the need for further analysis to identify optimal technique combinations for achieving higher accuracy.

The model development process consists of Data Preparation, Data Preprocessing, Model Training, and Model Evaluation. The problem in the model development process is that the accuracy results could be more optimal due to the complex and varied dataset [19]. To overcome this problem, the customer churn prediction model needs to be optimized to become more accurate in processing complex and varied datasets in the future. Therefore, in this research, several additional processes carried out, such as Feature Selection to find out what features affect the prediction results and Dataset Balancing to overcome the problem of dataset inequality and to get maximum results in the process of developing the model, so that the model could be more optimal and has better accuracy compare to the previous one.

One of the studies [10], the Gravitational Search Algorithm (GSA) was employed for Feature Selection without dataset balancing, yielding accuracy results of approximately 80-81%. Still, there was no mention of comparison results between using and not using Feature Selection. Another research study [18] used mRMR and Relief-F for Feature Selection and oversampling for balancing the Dataset. The accuracy results increased from 81.65% (without Feature Selection) to 83.97% using mRMR and 82.15% using Relief-F in the KNN algorithm.

In contrast, several algorithms, such as SVM, showed no change in accuracy for those two Feature Selections used. Decision Tree and Random Forest accuracy experienced a decrease for the two Feature Selections used. Meanwhile, the results of the Dataset Balancing process using the Oversampling technique, both before and after Feature Selection, decreased the accuracy of all the used algorithms. There is also research by [16] using Lasso Regression for Feature Selection and Random Oversampling for Balancing Datasets on ANN algorithm, with accuracy results of around 86-88%. However, the comparison of results using feature selection and those without feature selection should still be mentioned. Therefore, the researchers' analysis was conducted by utilizing and combining several feature selection and dataset balancing techniques. The aim is to determine the accuracy of the best model from several combinations using the ANN algorithm and to compare it with the model from previous research. The feature selection techniques used were not only Lasso Regression but also mRMR and XGBoost. The used dataset balancing technique was not only ROS but also SMOTE, to determine which combination of methods was most prudent. Some combinations of methods were obtained as follows:

- Lasso Regression + ROS
- Lasso Regression + SMOTE
- mRMR + ROS
- mRMR + SMOTE
- XGBoost + ROS
- XGBoost + SMOTE

This would be combined with different evaluation methods, namely Holdout and 10-Fold Cross-Validation (CV). The following were combinations of evaluation methods:

- Holdout + Lasso Regression + ROS
- Holdout + Lasso Regression + SMOTE
- Holdout + mRMR + ROS
- Holdout + mRMR + SMOTE
- Holdout + XGBoost + ROS
- Holdout + XGBoost + SMOTE
- 10-Fold CV + Lasso Regression + ROS
- 10-Fold CV + Lasso Regression + SMOTE
- 10-Fold CV + mRMR + ROS
- 10-Fold CV + mRMR + SMOTE
- 10-Fold CV + XGBoost + ROS
- 10-Fold CV + XGBoost + SMOTE

In this research, a customer churn model was created using ANN. In general, ANN could be represented as a system of interconnected neurons since it was inspired by biological neural networks, specifically the human brain [20], [21]. ANN could be used for classification by using features or attributes as input and creating a non-linear function that functions to predict the dependent variable [22]. To put it simply, the Artificial Neural Network (ANN) JST Backpropagation process can be divided into two phases: training and testing.

Training, as mentioned above, was a learning process of the artificial neural network system, to understand the input values and how to map them to the output until an appropriate model was obtained [23]–[25]. Proper dataset processing was necessary to enhance model performance and achieve high-accuracy results. Therefore, this research used the SMOTE dataset balancing technique to take samples from the minority class and to create new samples synthetically [26], also ROS to reproduce data randomly [16]. Thus, the number of courses between the majority class and the minority class would be balanced [27].

This research utilized XGBoost, Lasso Regression, and mRMR as feature selection methods. XGBoost was used to identify essential features by selecting those with the highest feature importance values. Lasso Regression selected features by identifying those with coefficient values whom not equal to zero on the features [28]. Meanwhile, mRMR was applied to prioritize features with the lowest redundancy and highest relevance values [18], [29]. These methods were employed to improve model accuracy by ensuring that the most significant

and relevant features are utilized.

This research employed two evaluation methods: the Holdout Method and the 10-fold CV Method. These methods are divided into 90% training data and 10% testing data, and a Confusion Matrix is used to calculate Accuracy, Precision, Recall, and F-1 Score.

II. Materials and Methods

The research methodology is systematically outlined in Fig. 1, detailing the key stages involved in developing the customer churn prediction model, including Data Preparation, Data Preprocessing, Model Training, and Model Evaluation.

A. Data Preparation

This research used the IBM Telco dataset, which contains 7043 samples of customer information and 21 attributes, namely: customerID, gender, SeniorCitizen, Partner, Dependents, tenure, PhoneService, MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovie, Contact, PaperlessBilling, PaymentMethod, MonthlyCharges, TotalCharges, and Churn (dependent attribute/class). The dataset used was secondary and had been widely used in several previous studies [16]. This dataset was available on Kaggle [1] : The dataset was selected for its relevance to the research objectives, as it focuses specifically on customer churn in the telecommunications industry. It provided a comprehensive set of features, including demographic, usage, and payment information, which are critical for building robust churn prediction models.

Additionally, the dataset was widely recognized and had been used in prior research, enabling comparative analysis with existing studies. It was publicly available, and its anonymized nature ensures compliance with ethical standards, allowing for the reproducibility of the results. Furthermore, the dataset presented a significant technical challenge due to its imbalanced class distribution, making it an ideal testbed for evaluating data balancing techniques such as SMOTE and ROS. SMOTE and ROS are utilized to address class imbalance since they have demonstrated effectiveness in prior churn prediction research [16], [26]. SMOTE synthesized new minority class samples while maintaining dataset diversity, whereas ROS balanced the dataset by replicating existing minority class samples without introducing noise, ensuring robust model training.

B. Data Preprocessing

The IBM Telco dataset contained categorical attributes with two and three unique values. Therefore, five stages of data processing would be performed to create a model that requires numeric attribute values and the process flow. Before processing these five stages, we would remove the attribute ”customerID” as it did not have a significant impact creating the model. We would also change the data type of ”TotalCharges” data type from object to float because this attribute was numeric.

```
def DataPreparation():
dataset = pd.readcsv('dataset/IBM Telco.csv')
del dataset['customerID']
dataset['TotalCharges'] = pd.to_numeric(dataset['TotalCharges'],
errors = 'coerce')
return dataset
```

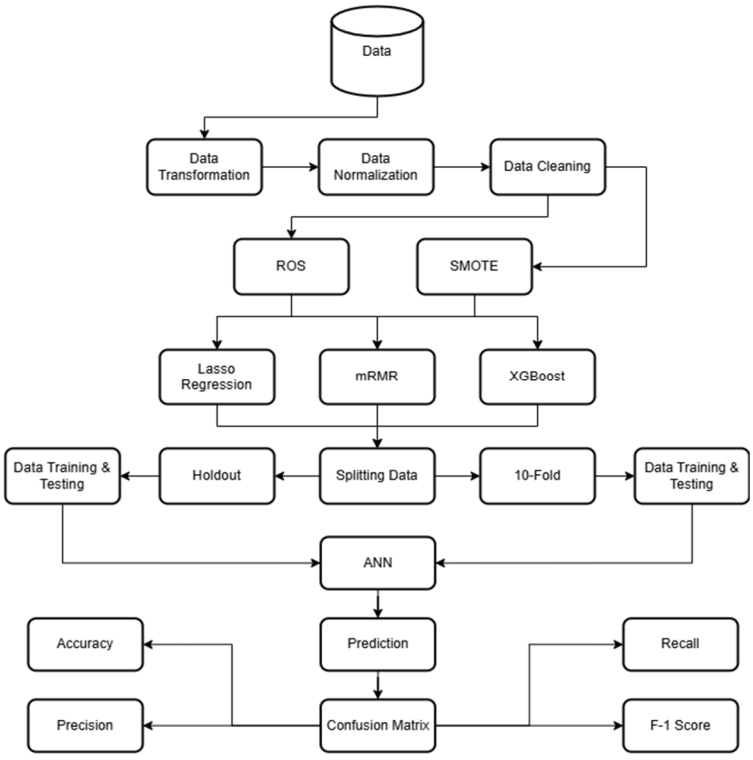

Fig. 1 Research Methodology Framework

[1] https://www.kaggle.com/code/ahmedshahriarsakib/churn-predictioni-eda-statistical-analysis/input

1) Data Transformation: Creating a Customer Churn Prediction model using ANN required a dataset containing numerical values, so a data transformation process was needed. Two methods would be used in the data transformation process: One-Hot Encoding and Label Encoding.

The first stage, the One-Hot Encoding method from the Python library, used pd.get_dummies to convert categorical attributes into numerical value representations. However, in this experiment, the results were in Boolean form. Therefore, changing the numerical form would be made at the next stage. The process flow of the One-Hot Encoding method begins by selecting attributes that have more than two unique values and a data type other than int64 or float64. This allows for changing the attribute according to the number of unique values and their respective names. The selected attributes were as follows: MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, Contract, and PaymentMethod. The results of the One-Hot Encoding method process are shown in Table I, and the total attributes are 40 (independent).

TABLE I
ONE-HOT ENCODING PROCESS RESULT

No	Gender	…	PaymentMethod_Mailed check
1	Female	…	False
2	Male	…	True
…	…	…	…
7042	Male	…	True
7043	Male	…	False

In the second stage, the Label Encoding method from the Python library, namely Scikit-learn, was used to convert non-numeric or Boolean values into numeric ones. In this case, the values were (Male = 1 and Female = 0), (Yes = 1 and No = 0), (True = 1 and False = 0). The process flow of the Label Encoding method begins by selecting attributes that meet the condition where the attribute has a unique value of 2 and the data type is neither int64 nor float64. This step changes the attribute value from Boolean or categorical to numeric (0 or 1). All attributes were selected except the following 4: SeniorCitizen, tenure, MonthlyCharges, and TotalCharges. The Label Encoding method's process results are shown in Table II, which displays all attribute values, including Boolean and categorical data types, represented as numeric values (0 or 1).

TABLE II
LABEL ENCODING PROCESS RESULT

No	Gender	…	PaymentMethod_Mailed check
1	0	…	0
2	1	…	1
…	…	…	…
7042	1	…	1
7043	1	…	0

2) *Data Normalization:* Based on information from previous research, the IBM Telco dataset contained several attributes with a relatively high value range. Therefore, Normalization was carried out using the MinMaxScaler method from the Python library, namely Scikit-learn. This method would change attribute values to values in the range [0, 1], which was only applied to attributes with a high-value range.

The process flow of the MinMaxScaler method begins by selecting attributes that have more than two unique values, with the data type being either int64 or float64. It then changes the attribute value to a value within the range [0, 1]. The selected attributes were tenure, MonthlyCharges, and TotalCharges. Data after processing using the MinMaxScaler method was shown in Table III.

TABLE III
MINMAXSCALER PROCESS RESULT

No	Tenure	Monthly Charges	Total Charges
1	0.013889	0.115423	0.001275
2	0.472222	0.385075	0.215867
…	…	...	…
7042	0.055556	0.558706	0.03321
7043	0.916667	0.869652	0.787641

3) Missing Value: Based on information from previous research, more values were needed for the IBM Telco dataset. Therefore, we checked and used the SimpleImputer method from the Python library, namely Scikit-learn, to replace it with the mean value. The SimpleImputer method begins by checking for missing values and filling them in with the average value (mean). The attribute with missing values was TotalCharges with 11 missing values. Data after processing using the SimpleImputer method was presented in Table IV.

TABLE IV
SIMPLEIMPUTER PROCESS RESULT

No	Gender	…	MonthlyCharges	TotalCharges	Churn
488	0	…	0.341294	0.261309	0
753	1	…	0.019900	0.261309	0
936	0	…	0.622886	0.261309	0
1082	1	…	0.074627	0.261309	0
3826	1	…	0.070647	0.261309	0
4380	0	…	0.017413	0.261309	0
5218	1	…	0.014428	0.261309	0
6670	0	…	0.548259	0.261309	0
6754	1	…	0.434328	0.261309	0

4) Balancing Data: The IBM Telco dataset had a relatively high ratio of positive and negative classes, specifically 73.46% (Churn = No) and 26.54% (Churn = Yes). Therefore, data balancing was carried out using the SMOTE method from the Python library, namely Imbalanced-learn, to handle data imbalance problems by generating synthetic samples for minority classes, thereby avoiding overfitting, which often occurred in simple oversampling methods, and the ROS method from the Python library, namely Imbalanced-learn to handle data imbalance problems by randomly reproducing data in a dataset.

5) Feature Selection: To reduce data dimensionality, model complexity, and improve overall performance, a feature selection process was implemented using three methods: XGBoost, Lasso Regression, and mRMR. These methods were chosen for their complementary strengths: XGBoost efficiently ranks feature importance while handling high-dimensional datasets [19], Lasso Regression applies regularization to mitigate overfitting [28], and mRMR prioritizes features with high relevance and minimal redundancy [29], ensuring the identification of an optimal feature set. All feature selection techniques were implemented

using Python libraries, facilitating the selection of features that are most relevant to and influential on the class values within the dataset.

C. Hyperparameter Tuning

After data preprocessing was complete, it was necessary to configure the hyperparameters in the ANN method, the data balancing method, and the feature selection method to produce high-accuracy values.

TABLE V
HYPERPARAMETERS

Method	Hyperparameter	Value
ANN	hidden_layer_sizes	200, 250
	activation	ReLu
	solver	Adam
	max_iter	10000
SMOTE	random_state	42
ROS	sampling_strategy	auto
	random_state	42
Lasso Regression	alpha	0.01
mRMR	K	28
	return_score	TRUE
	show_progress	FALSE
XGBoost	Default	

Table V summarizes the key hyperparameters used for various machine learning methods in this study. The explanation was as follows:

1) *ANN (MLPClassifier):* The ANN architecture was configured with two hidden layers consisting of 200 and 250 neurons, as these configurations were identified through empirical tuning to balance model complexity and training efficiency. The ReLU activation function and Adam optimizer were chosen for their widespread application and proven effectiveness in handling gradient-based optimization challenges in deep learning models [16]. Additionally, the MLPClassifier was set to perform a maximum of 10,000 iterations to ensure convergence and achieve optimal performance through extensive experimentation.

2) SMOTE: The Synthetic Minority Oversampling Technique (SMOTE) was configured with a random_state of 42 to ensure reproducibility, adhering to best practices in machine learning experiments. This approach has been demonstrated to effectively reduce bias in imbalanced datasets and improve model performance without introducing noise [26].

3) ROS: Random Over-Sampling (ROS) was selected as a data balancing method due to its ability to replicate minority class samples without altering the original data distribution, ensuring a balanced dataset for training. The method was configured to automatically determine the sampling strategy (sampling_strategy=auto) to balance class proportions, while random_state=42 was set to ensure consistent and reproducible results. The technique has been widely used in predictive modelling tasks, particularly due to its simplicity and effectiveness in handling class imbalance without introducing significant bias [16].

4) Lasso Regression: Lasso Regression was chosen for feature selection due to its ability to perform both dimensionality reduction and regularization by penalizing less significant features, improving model interpretability and efficiency. The regularization parameter (α = 0.01) was optimized to balance bias and variance, effectively retaining the most relevant features while reducing overfitting and maintaining model accuracy. This method has been widely adopted in high-dimensional data scenarios for its robust performance [28].

5) mRMR: The mRMR method was employed for feature selection to identify attributes with high relevance to the target variable while minimizing redundancy among selected features. The parameter K = 28 was determined experimentally to optimize accuracy, selecting 28 features that significantly contribute to the model. The setting return_scores=True was used to provide insights into feature importance, while show_progress=False suppressed execution progress updates for computational efficiency. This method was particularly effective in datasets with correlated features, ensuring the selection of non-redundant and highly informative attributes [29].

6) XGBoost: XGBoost was selected for feature selection due to its ability to rank feature importance while efficiently handling high-dimensional datasets. The method leverages its optimized gradient boosting algorithm to enhance predictive accuracy while retaining the library's default parameters to ensure robust and computationally efficient performance. XGBoost's effectiveness and scalability have established it as a standard in feature selection for classification tasks, including churn prediction [19].

D. Evaluation

In this research, a confusion matrix was employed as a model evaluation method because it provided a comprehensive summary of classification prediction results, detailing the number of correct and incorrect predictions across classes. It included key terms such as True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN), which are used to calculate essential metrics like accuracy, precision, recall, and F1-score. To ensure a thorough and balanced assessment of the model's performance, the study employed two standard evaluation methods: Holdout and 10-fold cross-validation [8]–[11], [16], [28]. The Holdout method, which splits the dataset into 90% training and 10% testing, was favored for its simplicity and computational efficiency, making it suitable for quick benchmarking. However, as Holdout relied on a single dataset split, it might introduce bias and overfitting. To address this limitation, 10-Fold Cross-Validation was employed, iteratively training and testing the model across multiple splits to reduce variance and provide a more generalized performance evaluation. By integrating these two methods, the research struck a balance between computational efficiency and robust validation, ensuring that the proposed ANN-based churn prediction model was rigorously and reliably evaluated for real-world applicability.

1) Accuracy: A measurement was being done to determine how well the model predicts a classification, with the following equation:

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

2) Precision: A measurement was being done to determine well the model made positive predictions in classification, namely how many of all positive predictions are truly positive, with equation as follows:

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

3) Recall: A measurement was being done to determine well the model makes positive predictions on truly positive data, with equation as follows:

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

4) F-1 Score: The harmonic mean of precision and recall, with equation as follows:

$$F-1Score = 2 * \left(\frac{Precision*Recall}{Precision+Recall}\right) \quad (4)$$

III. Result and Discussion

Model evaluation was performed using the Holdout and 10-Fold Cross Validation methods by dividing the dataset into 90% for training data and 10% for testing data. Evaluation was performed on the ANN+SMOTE and ANN+ROS models using each feature selection method. After conducting several model evaluation experiments by changing the number of features in each feature selection, XGBoost was obtained the best accuracy results from all feature selections with 28 features. Therefore, a comparison would be made between models with the same number of features. Model testing results were shown in Tables VI, VII, VIII and IX.

A. ANN (Without Data Balancing + Feature Selection)

Table VI and Fig. 2 above presented the baseline performance of ANN model without data balancing or feature selection, showing that the Holdout method achieves slightly higher accuracy (75.60%) compared to 10-Fold Cross-Validation (75.32%), along with marginally better Precision and F1-Score. However, Recall was nearly identical. These results indicated limited generalization capabilities with raw data, reflected in modest performance across metrics.

TABLE VI
HOLDOUT AND 10-FOLD CROSS VALIDATION MODEL RESULTS

Evaluation Metric	Data Splitting Method	
	Holdout	10-Fold CV
Accuracy	0.75603	0.75323
Precision	0.56707	0.53501
Recall	0.47938	0.53558
F-1 Score	0.51955	0.53529

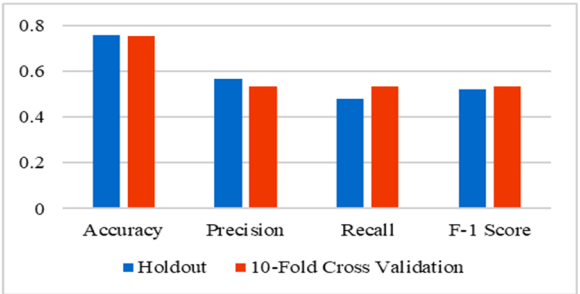

Fig. 2 Comparison of Holdout and 10-Fold Cross Validation Model Results

B. ANN + Data Balancing

Table VII and Fig. 3 evaluated the impact of data balancing techniques (SMOTE and ROS) on the ANN model, showing significant performance improvements compared to the baseline (Table VI). Accuracy improved to 87.05% with ROS and 85.41% with SMOTE (Holdout). ROS consistently outperformed SMOTE in Recall and F1-Score, highlighting its effectiveness in handling class imbalance.

TABLE VII
ANN + DATA BALANCING MODEL RESULTS

Evaluation Metric	Data Splitting Method	Data Balancing Technique	
		SMOTE	ROS
Accuracy	Holdout	0.85411	0.87053
	10-Fold CV	0.83903	0.85844
Precision	Holdout	0.81426	0.83673
	10-Fold CV	0.81672	0.81289
Recall	Holdout	0.92642	0.92830
	10-Fold CV	0.87418	0.93119
F-1 Score	Holdout	0.86673	0.88014
	10-Fold CV	0.84447	0.86803

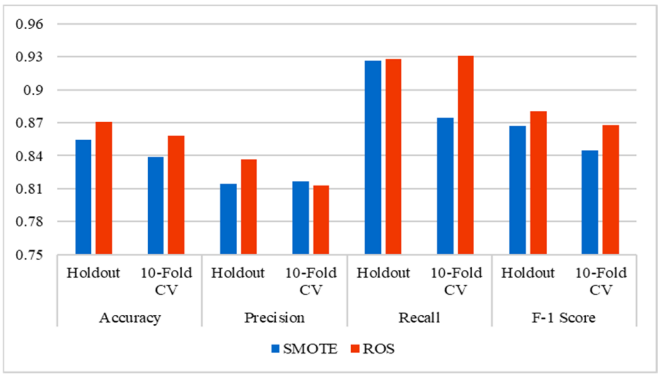

Fig. 3 Comparison of ANN + Data Balancing Model Results

C. ANN + Feature Selection

Table VIII and Fig. 4 highlight the effects of feature selection methods (Lasso Regression, mRMR, and XGBoost) on the ANN model, showing that XGBoost achieved the highest accuracy (77.73%) under Holdout, surpassing other methods. Similar trends were observed in Precision and F1-Score, but the improvements remained modest compared to Table VII, where data balancing had a more significant impact.

TABLE VIII
ANN + FEATURE SELECTION MODEL RESULTS

Evaluation Metric	Data Splitting Method	Feature Selection Method		
		Lasso Regression	mRMR	XGBoost
Accuracy	Holdout	0.74894	0.76170	0.77730
	10-Fold CV	0.74457	0.75706	0.75408
Precision	Holdout	0.54359	0.57386	0.59686
	10-Fold CV	0.51949	0.54350	0.53816
Recall	Holdout	0.54639	0.52062	0.58763
	10-Fold CV	0.4992	0.52809	0.51685
F-1 Score	Holdout	0.54499	0.54595	0.59221
	10-Fold CV	0.50914	0.53569	0.52729

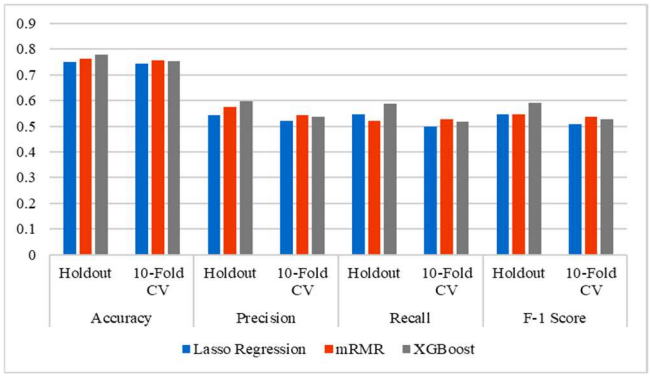

Fig. 4 Comparison of ANN + Feature Selection Model Results

D. ANN + Feature Selection + Data Balancing

Table IX, Fig. 5, and Fig. 6 collectively illustrate the impact of integrating feature selection and data balancing techniques on the performance of the ANN model. This analysis highlighted the effectiveness of advanced methodologies, with Fig. 5 providing a detailed breakdown of key metrics—accuracy, precision, recall, and F1-score—across various combinations, and Fig. 6 comparing the outcomes of Holdout and 10-Fold Cross-Validation. Among feature selection methods (Lasso Regression, mRMR, and XGBoost), XGBoost yields the best accuracy (77.73%) under Holdout, outperforming other techniques. However, improvements from feature selection alone remain modest compared to data balancing, as seen in Table VII.

Notably, the combination of XGBoost and ROS consistently delivered superior results across all metrics under both validation methods. Based on the results in Tables VI through IX, while Holdout showed higher accuracy, 10-Fold Cross-Validation provided more generalized results, emphasizing the importance of robust evaluation methods to ensure model reliability across varied datasets. These findings underscore that feature selection, while reducing dimensionality and enhancing interpretability, achieves its full potential when combined with data balancing, ensuring robust and generalized model performance. Accordingly, the following points lead further into these findings.

TABLE IX
ANN + FEATURE SELECTION + DATA BALANCING MODEL RESULTS

Evaluation Metric	Data Splitting Method	Feature Selection Method + Data Balancing Technique					
		Lasso Regression		mRMR		XGBoost	
		SMOTE	ROS	SMOTE	ROS	SMOTE	ROS
Accuracy	Holdout	0.83092	0.84348	0.83961	0.85024	0.86184	0.88502
	10-Fold CV	0.81554	0.82007	0.82985	0.83669	0.84260	0.84550
Precision	Holdout	0.80136	0.80263	0.82500	0.82496	0.83888	0.85128
	10-Fold CV	0.79199	0.77490	0.79829	0.80370	0.81545	0.80613
Recall	Holdout	0.89057	0.92075	0.87170	0.89811	0.90377	0.93962
	10-Fold CV	0.85582	0.90220	0.88268	0.89099	0.88558	0.90974
F-1 Score	Holdout	0.84361	0.85764	0.84771	0.85998	0.87012	0.89327
	10-Fold CV	0.82267	0.83372	0.83837	0.84510	0.84907	0.85481

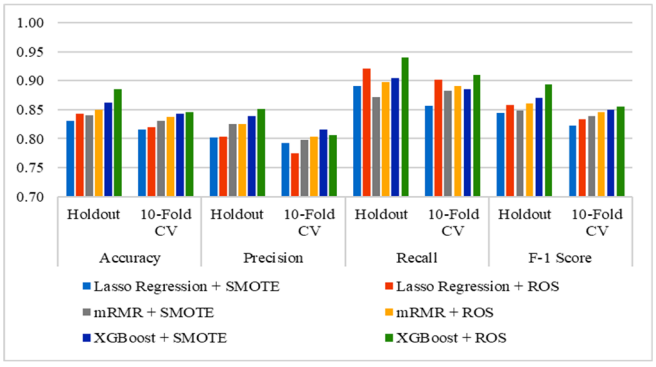

Fig. 5 Comparison of ANN + Feature Selection + Data Balancing Model Results

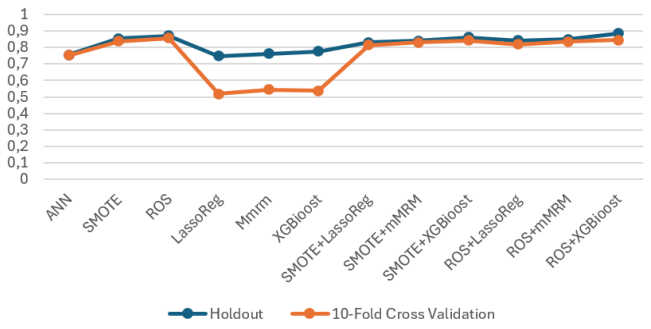

Fig. 6 Comparison Holdout and 10-Fold Cross Validation

1) Data Balancing: The application of data balancing techniques, such as SMOTE and ROS, significantly enhanced model accuracy by more than 10% for both Holdout and 10-Fold Cross Validation methods compared to models that did not employ data balancing. Among these techniques, ROS (Random Over-Sampling) demonstrated superior performance across all evaluation metrics. It was likely attributed to its capability to effectively address class imbalances while preserving the diversity and distribution of the original dataset.

2) Feature Selection: Feature selection led to modest performance enhancements, with approximately a 2% increase in accuracy for the Holdout method and a 0.2% improvement for the 10-Fold Cross Validation method. Among the evaluated feature selection techniques, XGBoost achieved the highest improvement, which is attributed to its ability to rank features by importance and minimize redundancy effectively. In comparison, Lasso Regression and mRMR delivered consistent but slightly lower enhancements in model performance.

3) Data Balancing + Feature Selection: The combination of data balancing and feature selection produced the highest overall performance, with an improvement of approximately 12% in Holdout accuracy and 9% in 10-Fold Cross Validation accuracy compared to models that do not employ these techniques. Among the combinations tested, XGBoost with

ROS achieved the best results across all evaluation metrics, particularly excelling in Recall and F-1 Score. As illustrated in Fig. 6, the ROS + XGBoost combination consistently delivered superior results, by emphasizing its effectiveness for applications requiring high recall, such as customer churn prediction.

4) Implications: The choice between the Holdout and 10-Fold Cross Validation (CV) methods had significant implications for model performance and applicability. While the Holdout method often achieves higher accuracy, it is prone to overfitting due to its dependency on a single data split, which may only partially represent the dataset's diversity. In contrast, the 10-Fold CV method provided more robust and consistent performance across varying datasets, making it the preferred choice for real-world scenarios that involve diverse or evolving data. For business-critical applications, such as customer churn prediction, the combination of ROS and XGBoost offers an optimal balance of high performance, interpretability, and computational efficiency. Moreover, models validated through 10-Fold CV were recommended for production deployment, as this approach ensures greater generalizability and reliability when applied to unseen datasets.

E. Comparison of Studies

In previous research [16], there were several results of previous research and comparisons with this research on the IBM Telco dataset.

TABLE X
RESULTS OF PREVIOUS RESEARCH

No	Authors	Dataset	Classifiers	Evaluation Model	Accuracy (%)
1	Agrawal et al. [30]	IBM Telco	Multi-layered ANN	Holdout 80% for training and 20% for testing	80.03%
2	Mohammad et al. [31]	IBM Telco	ANN	Holdout 70% for training and 30% for testing	85.55%
3	Momin et al. [32]	IBM Telco	ANN	Holdout 90% for training and 10% for testing	82.83%
4	Fujo et al. [16]	IBM Telco	ANN	Holdout 90% for training and 10% for testing	88.12%

Table X compares the results of ANN models that were created previously with different evaluation methods and models on the IBM Telco dataset. Based on the results from Table X, the research obtained the best results compared to other studies and provides recommendations for using feature importance techniques, such as XGBoost, and dataset balancing techniques, such as ROS, which can be implemented into web applications using Flask. Therefore, this research could contribute to the optimization of the previously created ANN model by utilizing different feature selection and dataset balancing techniques, and following the recommendations provided. A comparison of model evaluation results is shown in Table XI.

TABLE XI
COMPARISON OF MODEL EVALUATION RESULTS

No	Authors	Holdout	10-Fold CV
1	Agrawal et al. [30]	80.03%	Not Applied
2	Mohammad et al. [31]	85.55%	Not Applied
3	Momin et al. [32]	82.83%	Not Applied
4	Fujo et al. [16]	88.12%	86.57%
5	Proposed Method	88.50%	84.54%

F. Data Privacy and Compliance

Data privacy was a top priority in this research. All data used had undergone anonymization, where personally identifiable information had been removed or encrypted to prevent unauthorized disclosure. Furthermore, this research complied with applicable data privacy regulations, as the data had already been anonymized.

G. Responsible Implementation and Risk Management

By considering the technical aspect, this research also emphasized the importance of the responsible application of the model in real-world environments. Before implementing, a risk assessment would be conducted to identify and mitigate potential negative impacts, such as prediction errors that could affect business decisions or customer experiences. Long-term impact monitoring would also be implemented to ensure the model continues functioning as intended without causing unintended side effects. As part of these efforts, internal controls would be introduced to prevent model misuse, including transparency policies, regular audits, and oversight mechanisms to ensure the model is used fairly and ethically.

H. External Validation and Research Limitations

External validation was an important step that needed to be carried out to generalize the proposed model. Although internal validation using Holdout and 10-Fold CV had shown good model performance, testing this model on different datasets was essential to assess its robustness and generalization. Plans included testing the model on other telecom datasets available in public repositories such as Kaggle and the UCI Machine Learning Repository.

Additionally, we would apply the model in real-world scenarios in collaboration with local telecommunications companies to evaluate the model's performance in natural operational environments. This implementation would involve integrating the model into existing Customer Relationship Management (CRM) systems to identify high-risk customers in real-time and facilitate proactive retention strategies, such as offering personalized service plans or discounts. The model's ability to handle diverse datasets with varying levels of imbalance was evaluated using scalable techniques, such as ROS and XGBoost, to ensure consistent performance in practical settings. This pilot would include an analysis of newer and more diverse customer data, which would enable us to assess the model's ability to handle a wider range of data.

Although this research has shown promising results in optimizing Artificial Neural Network (ANN) models for customer churn prediction, several limitations must be noted. First, the dataset used in this research came from a single source, IBM Telco, which might only partially represent the variety of customer data and behaviors in the broader

telecommunications industry. Thus, the findings of this study might lack generalizability to telecommunication companies with different customer demographics, services, or geographic markets. Second, this research did not account for changes in customer behavior over time, which could impact the model's long-term accuracy. Incorporating temporal and behavioral trends into the model could further enhance its predictive capabilities, particularly in dynamic customer environments. Third, although data balancing methods such as SMOTE and ROS were used to overcome class imbalance, they could also introduce bias.

Nevertheless, this research made a unique contribution by integrating advanced feature selection techniques, such as XGBoost, with data balancing methods like ROS, significantly improving model performance across accuracy, precision, recall, and F1-score. The proposed model offered practical value to the telecommunications industry by enabling the real-time identification of high-risk customers through integration into existing CRM systems, thereby facilitating proactive retention strategies. Additionally, testing the model on external datasets from different geographic regions and market segments would provide insights into its scalability and robustness, ensuring its practical applicability in diverse telecommunication settings. Its scalability and robustness across diverse customer segments ensured that the model remains a valuable tool for long-term churn management despite these limitations.

IV. Conclusion

This research successfully optimized Artificial Neural Network (ANN) models for customer churn prediction, achieving a superior accuracy of 88.50% using the Holdout method. Although the 10-Fold Cross Validation method showed slightly lower accuracy at 84.54%, the findings highlighted the critical role of feature selection techniques, dataset balancing, and data preprocessing in enhancing model performance. The integration of advanced techniques, such as XGBoost for feature selection and ROS for data balancing, significantly improved prediction accuracy, providing a scalable and adaptable framework for applications in the telecommunications industry. Future research could enhance this model by incorporating temporal dynamics and customer behavior trends to improve long-term adaptability and accuracy. Validating the model on datasets from diverse geographic regions and market segments would also be essential to ensure generalizability and robustness. By addressing these areas and exploring additional features of engineering and data balancing methods, this research laid the foundation for developing even more effective and scalable churn prediction solutions in dynamic, real-world scenarios.

Acknowledgment

We thank Universitas Buddhi Dharma, especially the Department of Informatics Engineering, for their support and resources. Thanks to colleagues and faculty members for their valuable feedback and all who contributed to this research.

References

[1] A. Amin, F. Al-Obeidat, B. Shah, A. Adnan, J. Loo, and S. Anwar, "Customer churn prediction in the telecommunication industry using data certainty," *J. Bus. Res.*, vol. 94, pp. 290–301, Jan. 2019, doi:10.1016/j.jbusres.2018.03.003.

[2] Q. Bi, "Cultivating loyal customers through online customer communities: A psychological contract perspective," *J. Bus. Res.*, vol. 103, pp. 34–44, Oct. 2019, doi: 10.1016/j.jbusres.2019.06.005.

[3] X. Xiahou and Y. Harada, "B2C E-Commerce customer churn prediction based on K-means and SVM," *J. Theor. Appl. Electron. Commer. Res.*, vol. 17, no. 2, pp. 458–475, Jun. 2022, doi:10.3390/jtaer17020024.

[4] S. O. Abdulsalam, M. O. Arowolo, Y. K. Saheed, and J. O. Afolayan, "Customer churn prediction in telecommunication industry using classification and regression trees and artificial neural network algorithms," *Indones. J. Electr. Eng. Inform.*, vol. 10, no. 2, pp. 431–440, Jun. 2022, doi: 10.52549/ijeei.v10i2.2985.

[5] S. M. Mirabdolbaghi and B. Amiri, "Model optimization analysis of customer churn prediction using machine learning algorithms with focus on feature reductions," *Discrete Dyn. Nat. Soc.*, vol. 2022, Art. no. 5134356, 2022, doi: 10.1155/2022/5134356.

[6] M. Vaudevan, R. S. Narayanan, S. F. Nakeeb, and Abhishek, "Customer churn analysis using XGBoosted decision trees," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 25, no. 1, pp. 488–495, Jan. 2022, doi: 10.11591/ijeecs.v25.i1.pp488-495.

[7] A. De Caigny, K. Coussement, and K. W. De Bock, "A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees," *Eur. J. Oper. Res.*, vol. 269, no. 2, pp. 760–772, Sep. 2018, doi: 10.1016/j.ejor.2018.02.009.

[8] A. K. Ahmad, A. Jafar, and K. Aljoumaa, "Customer churn prediction in telecom using machine learning in big data platform," *J. Big Data*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40537-019-0191-6.

[9] Y. Khan, S. Shafiq, A. Naeem, S. Ahmed, N. Safwan, and S. Hussain, "Customers churn prediction using artificial neural networks (ANN) in telecom industry," Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 9, pp. 1–10, 2019, doi: 10.14569/ijacsa.2019.0100918.

[10] 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, Feb. 2022, doi: 10.1007/s00607-021-00908-y.

[11] 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, doi:10.1109/access.2021.3073776.

[12] S. Kumar and M. Kumar, "Predicting customer churn using artificial neural network," in *Commun. Comput. Inf. Sci.*, vol. 1000, pp. 299–306, 2019, doi: 10.1007/978-3-030-20257-6_25.

[13] G. Torkzadeh, J. C. J. Chang, and G. W. Hansen, "Identifying issues in customer relationship management at Merck-Medco," *Decis. Support Syst.*, vol. 42, no. 2, pp. 1116–1130, Nov. 2006, doi:10.1016/j.dss.2005.10.003.

[14] A. Amin et al., "Just-in-time customer churn prediction in the telecommunication sector," *J. Supercomput.*, vol. 76, no. 6, pp. 3924–3948, Jun. 2020, doi: 10.1007/s11227-017-2149-9.

[15] C. Wang, D. Han, W. Fan, and Q. Liu, "Customer churn prediction with feature embedded convolutional neural network: An empirical study in the internet funds industry," *Int. J. Comput. Intell. Appl.*, vol. 18, no. 1, Art. no. 1950003, 2019, doi: 10.1142/S1469026819500032.

[16] S. W. Fujo, S. Subramanian, and M. A. Khder, "Customer churn prediction in telecommunication industry using deep learning," *Inf. Sci. Lett.*, vol. 11, no. 1, pp. 185–198, Jan. 2022, doi: 10.18576/isl/110120.

[17] A. De Caigny, K. Coussement, K. W. De Bock, and S. Lessmann, "Incorporating textual information in customer churn prediction models based on a convolutional neural network," *Int. J. Forecast.*, vol. 36, no. 4, pp. 1563–1578, Oct. 2020, doi:10.1016/j.ijforecast.2019.03.029.

[18] M. Rahman and V. Kumar, "Machine learning based customer churn prediction in banking," in *Proc. 4th Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA)*, Nov. 2020, pp. 1196–1201, doi:10.1109/iceca49313.2020.9297529.

[19] P. Swetha and R. B. Dayananda, "Improvised XGBoost machine learning algorithm for customer churn prediction," *EAI Endorsed Trans. Energy Web*, vol. 7, no. 30, pp. 1–7, 2020.

[20] Y. Chen, L. Song, Y. Liu, L. Yang, and D. Li, "A review of the artificial neural network models for water quality prediction," *Appl. Sci.*, vol. 10, no. 17, Art. no. 5776, 2020, doi: 10.3390/app10175776.

[21] R. Dastres, M. Soori, and A. Neural, "Artificial neural network systems," *Int. J. Imaging Robot.*, vol. 21, no. 2, pp. 13–25, 2021.

[22] A. A. H. Adam Khatir and M. Bee, "Machine learning models and data-balancing techniques for credit scoring: What is the best combination?," *Risks*, vol. 10, no. 9, Art. no. 169, Aug. 2022, doi:10.3390/risks10090169.

[23] Y. Antonisfia, R. Susanti, Efendi, S. Anderson, and F. Anisa, "Design and development of a coffee blending device with carbon monoxide (CO) level identification based on artificial neural networks," *Int. J. Adv. Sci. Comput. Eng.*, vol. 5, no. 3, pp. 239–246, 2023, doi:10.62527/ijasce.5.3.171.

[24] R. Susanti, R. Nofendra, Z. Zaini, M. S. A. bin Suhaimi, and M. I. Rusydi, "The use of artificial neural networks in agricultural plants," *Andalas J. Electr. Electron. Eng. Technol.*, vol. 2, no. 2, pp. 62–68, Jan. 2023, doi: 10.25077/ajeeet.v2i2.32.

[25] R. Susanti, Z. R. Aidha, M. Yuliza, Suryadi, and S. Yondri, "Artificial neural network application for aroma monitoring on the coffee beans blending process," *Int. J. Informatics Vis.*, vol. 2, no. 3, pp. 147–152, 2018, doi: 10.30630/joiv.2.3.86.

[26] H. Zhang, L. Huang, C. Q. Wu, and Z. Li, "An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset," *Comput. Netw.*, vol. 177, Art. no. 107315, Aug. 2020, doi: 10.1016/j.comnet.2020.107315.

[27] F. Zulfikri, D. Tryanda, A. Syarif, and H. Patria, "Predicting peer to peer lending loan risk using classification approach," *Int. J. Adv. Sci. Comput. Eng.*, vol. 3, no. 2, pp. 94–100, 2021, doi:10.62527/ijasce.3.2.57.

[28] S. Shafiee, L. M. Lied, I. Burud, J. A. Dieseth, M. Alsheikh, and M. Lillemo, "Sequential forward selection and support vector regression in comparison to LASSO regression for spring wheat yield prediction based on UAV imagery," *Comput. Electron. Agric.*, vol. 183, Art. no. 106036, Jun. 2021, doi: 10.1016/j.compag.2021.106036.

[29] M. Billah and S. Waheed, "Minimum redundancy maximum relevance (mRMR) based feature selection from endoscopic images for automatic gastrointestinal polyp detection," *Multimed. Tools Appl.*, vol. 79, no. 33–34, pp. 23633–23643, Sep. 2020, doi: 10.1007/s11042-020-09151-7.

[30] S. Agrawal, A. Das, A. Gaikwad, and S. Dhage, "Customer churn prediction modelling based on behavioural patterns analysis using deep learning," in *Proc. Int. Conf. Smart Comput. Electron. Enterp. (ICSCEE)*, 2018, pp. 1–6, doi: 10.1109/icscee.2018.8538420.

[31] N. Muhammad, S. A. Ismail, M. Kama, O. Mohd Yusop, and A. Azmi, *Customer Churn Prediction in Telecommunication Industry Using Machine Learning Classifiers*, 2019, doi:10.1145/3387168.3387219.

[32] S. Momin, T. Bohra, and P. Raut, "Prediction of customer churn using machine learning," in *Proc. EAI Int. Conf. Big Data Innov. Sustain. Cognit. Comput.*, 2020, pp. 203–212, doi: 10.1007/978-3-030-19562-5_20.