BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **24**, No 3
Sofia • 2024 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2024-0022

# Multi-Level Machine Learning Model to Improve the Effectiveness of Predicting Customers Churn Banks

*Van-Binh Ngo*[1], *Van-Hieu Vu*[2]

[1]*FPT University, Ha Noi, Viet Nam*
[2]*Institute of Information Technology, Vietnam Academy of Science and Technology, Cau Giay, Ha Noi, Viet Nam*
*E-mails: binhnv11@fe.edu.vn vvhieu@ioit.ac.vn*

***Abstract*:** *This study presents a novel multi-level Stacking model designed to enhance the accuracy of customer churn prediction in the banking sector, a critical aspect for improving customer retention. Our approach integrates four distinct machine-learning algorithms – K-Nearest Neighbor* (*KNN*), *XGBoost*, *Random Forest* (*RF*), *and Support Vector Machine* (*SVM*) *– at the first level* (*Level 0*). *These algorithms generate initial predictions, which are then combined and fed into higher-level models* (*Level 1*) *comprising Logistic Regression, Recurrent Neural Network* (*RNN*), *and Deep Neural Network* (*DNN*).

*We evaluated the model through three scenarios: Scenario 1 uses Logistic Regression at Level 1, Scenario 2 employs a Deep Convolutional Neural Network* (*DNN*), *and Scenario 3 utilizes a Deep Recurrent Neural Network* (*RNN*). *Our experiments on multiple datasets demonstrate significant improvements over traditional methods. In particular, Scenario 1 achieved an accuracy of 91.08%, a ROC-AUC of 98%, and an AUC-PR of 98.15%. Comparisons with existing research further underscore the enhanced performance of our proposed model.*

***Keywords*:** *Customer churn, Banking sector, Stacking model, Machine learning.*

## 1. Introduction

The problem of retaining customers at commercial banks is always a top concern in the bank's business. Many methods have been developed [33], however, it is still not possible to confirm which model is suitable for detecting customer churn.

Convolutional Neural Networks (CNNs) have established their efficacy across various fields, notably in image and video recognition [14, 24], natural language processing [37], and speech recognition [20]. Their ability to extract high-level features from extensive datasets is a cornerstone of their success. Furthermore, CNNs excel in image processing tasks, as evidenced by significant achievements in this domain [23, 28]. Given these accomplishments, CNNs emerge as a promising approach for customer churn prediction. The advancements in deep learning [18] have further opened avenues for utilizing these sophisticated models in churn

prediction, suggesting their potential to enhance the accuracy and efficiency of such predictive analyses significantly.

Deep learning is renowned for its superior accuracy in handling big data challenges, primarily due to its ability to extract features automatically. Unlike traditional methods that require manual feature extraction, deep learning networks are trained on large sets of labeled data and learn features directly from this data. However, in customer churn prediction, the prevalent approach is to use data records based on attributes from a pre-constructed customer data table, which typically bypasses the feature extraction stage.

Given this context, machine learning techniques remain a viable and effective method. In our proposal, we adopt a traditional machine-learning approach. To enhance performance, we combine several machine-learning techniques in a two-tiered model, commonly known as a stacking model. This approach leverages the strengths of different algorithms at various levels, aiming to capitalize on the synergistic effect of the combined methods for more accurate churn prediction.

The proposed Stacking model is evaluated through three scenarios to demonstrate its effectiveness:

**Scenario 1.** Logistic Regression at Level 1.

**Scenario 2.** Deep Convolutional Neural Network (DNN) at Level 1.

**Scenario 3.** Deep Recurrent Neural Network (RNN) at Level 1.

Our experiments on multiple datasets, including a supplemented dataset from [34], and comparisons with research from [33], showcase the superior performance of our model. Additionally, practical implementations in three banking institutions illustrate the model's significant impact on customer retention.

The structure of the article is as follows: In Section 2 we introduce some related work, Section 3 presents some basic models, Section 4 is the proposed model and Section 5 is the experimental results. Finally, the conclusion in Section 6.

## 2. Related work

The study by Benlan He et al. [9] on customer churn prediction using SVM, logistic regression, and RBF SVM models reveals the challenge of data imbalance, where models achieved high accuracy but low recall, indicating a bias towards the majority class [13]. Authors in [8] compared K-Nearest Neighbor (KNN) and Decision Tree models for churn prediction, finding that while both models had high accuracy, there was a significant disparity in F1-scores, with DT outperforming KNN due to better recall [12].

Xu, Ma and Kim [30] developed a Stacking and Voting model integrating XGBoost, logistic regression, Decision Tree, and a naive classifier, achieving notable accuracy improvements [35]. Veningston et al. [29] explored various classification models, finding a diverse range of effectiveness with logistic regression, Decision Tree, Random Forest, KNN, and AdaBoost [34]. Renato, Silva and Tabak [6] conducted a detailed evaluation of various algorithms using AUC-ROC, highlighting the strong performance of Random Forests and ensemble

methods but also pointing out the variability in effectiveness across different models [10].

Kumar and Chandrakala [13] analyzed various methods for churn prediction, suggesting that combining SVM with boosting algorithms could improve performance [22]. Kumar and Chandrakala [13] presented a hybrid approach combining SVM with Adaboost for higher classification accuracy, although the complexity and computational demand were noted as limitations [17]. Hoang, Le and Nguyen [28] evaluated various models for customer churn prediction, showing that Random Forests and SVM generally outperformed others, while KNN and Logistic Regression exhibited lower but respectable performance, highlighting the complexity of finding a universally effective model [32].

These studies underscore the need for more nuanced model selection and continuous refinement of prediction models to enhance their effectiveness in various contexts.

## 3. Proposed model

In this paper, we propose a Stacking model to predict bank customer churn. Stacking or Stacked Generalization [16] is a method of combining multiple base models into a model meta-model to achieve higher prediction accuracy [29]. The base models will be trained on the training data set and finally, the meta-model will be trained based on the predictions from the base models as the features. This method can also be used for regression [5] and unsupervised learning [25].

### 3.1. Problem statement

The problem is stated as follows: for Dataset $D = \{X_{i=1}^{N}, y_{i=1}^{N}\}$, $i = 1, \ldots, N$, where $X_i$ are the observations, and $y_i$ are the corresponding labels. Let $M = \{M\}_{j=1}^{n}$, be the set of classification models for the $D$ dataset. Divide data set $X$ into $k$ parts as follows equation:

$$\begin{cases} D = D_1 \bigcup D_2 \bigcup \cdots \bigcup D_k = (X, y), \\ D_1 \bigcap D_2 \bigcap D_3 \bigcap \cdots \bigcap D_k = 0. \end{cases} \tag{1}$$

Training on set **M** of classification models obtains a new feature set, called Level 0:

$$\text{Level 0} = \begin{cases} D_i' \leftarrow \text{train}(M_j, \{D \setminus D_i\}), \\ i = 1, ..., (k-1) \text{ folds}, \\ j = 1, ..., n \text{ models}, \\ \text{get one fold for Test.} \end{cases} \tag{2}$$

Initialize an **M** classification model and perform training with a feature set that is the union of the predicted probabilities obtained in all Level 0 and k-Folds models.

$$\text{Level 1} = \begin{cases} D' = \bigcup D'_i,\ i = 1, ..., k, \\ \text{predictions} \leftarrow \text{train}(\mathbf{M}, D'), \\ \text{evaluate}\{\text{predictions}, y\}. \end{cases} \tag{3}$$

Level 0 models and Level 1 models will be modeled in Subsection 4.2 and Algorithm 1.

### 3.2. Problem modelling

In Fig. 1, the Stacking model consists of two levels with Level 0 (Level 0) including *n* models called *base models* (Model Stack), at this level the basic models have the task of make predictions that feature the model at Level 1 (Level). At Level 1, there will be a model called meta-model; this model is responsible for making final predictions from the features that are the predictions of the underlying models.

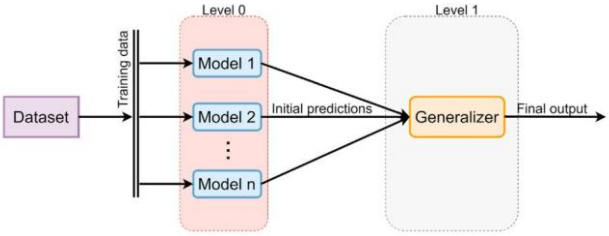



Fig. 1. Graphical representation of the Stacking method [19]

In this paper, we propose a Stacking model for predicting bank customer churn, which consists of two levels. The first level (Level 0: Model Stack) comprises four base classifiers: K-nearest neighbors, XGBoost, Random Forest, and Support Vector Machine. At the second level (Level 1: Meta Model), Logistic Regression, Deep Neural Networks (DNN), and Recurrent Neural Networks (RNN) are employed. The architecture of the proposed model is illustrated in Fig. 2.

In a two-level Stacked model, the results from the Level 0 models will be included in the Level 1 model in a specific way: Level 0 model output: Each selected machine learning model (KNN, XGBoost, Random Forest, and Support Vector Machine) is trained on the dataset. After training, these models generate predictions or output probabilities for each observation in the data set.

Combine Level 0 output: Predictions from all Level 0 models are combined to form a new dataset (called *new features*). This dataset includes the output probabilities from each Level 0 model, capturing the insights or predictions from these models.

Level 1 Input: The new dataset, consisting of the combined predictions from Level 0, is then used as input to the Level 1 model. This dataset provides a new set of features for Level 1 because it includes all the observations predicted in the learned Level 0 models.

Training Level 1 Model: Can use Logistic Regression, RNN or Deep Learning Neural Network (DNN). Level 1 models learn to make final predictions by efficiently interpreting the aggregated insights from Level 0 models (according to regression knowledge).

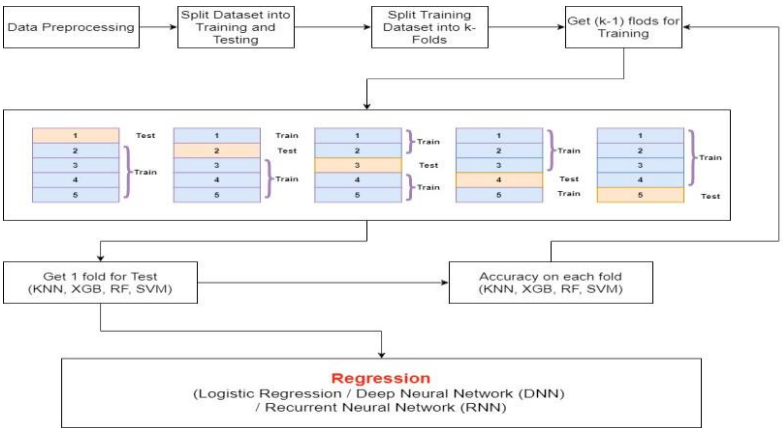



Fig. 2. System architecture model

Algorithm 1 will execute Level 0 and Level 1 explicitly.

**Algorithm 1. Stacking Model with *k*-Folds for Customer Churn Prediction**

**Require:** Dataset $D = \{\mathbf{X}_i, y_i\}_{i=1}^{N}$

**Ensure:** Predicted customer churn

**Step 1. Preprocess Dataset** $D$

**Step 2.** Initialize Level 0 Models: KNN, XGB, RF, SVM

**Step 3.** Initialize Level 1 Models: LR, RNN, DNN 4:

**Step 4.** Define the number of folds $k$

**Step 5. for** each fold in k-Folds **do**

**Step 6.** Partition D into the training set $D_{\text{train}}$ and test set $D_{\text{test}}$

**Step 7. for** each Level 0 Model $M$ **do**

**Step 8.** Train $M$ on $D_{\text{train}}$

**Step 9.** Predict on $D_{\text{test}}$ to get $P_M$

**Step 10. end for**

**Step 11.** Combine predictions $P = \{P_{\text{KNN}}, P_{\text{XGB}}, P_{\text{RF}}, P_{\text{SVM}}\}$

**Step 12. end for**

**Step 13.** Combine all fold predictions for Level 1 training

**Step 14. for** each Level 1 Model $M$ **do**

**Step 15.**Train $M$ using combined predictions as features

**Step 16. end for**

**Step 17.** Evaluate Level 1 Models on the combined test set

**Step 18. Return** best-performing model predictions

## 4. Experiments and results

### 4.1. Evaluation method

As for the evaluation method in the classification problem. In this study, we will use the methods to evaluate the performance of the model based on the confusion matrix (Confusion Matrix) by the following indicators: accuracy, precision, recall, and F1 score (Table 1).

Table 1. The confusion matrix

| Measure | Positive | Negative |
|---|---|---|
| Positive | TP | FN |
| Negative | FP | TN |

- TP (True Positive) is the total number of cases where the prediction model matches the correct pattern.
- TN (True Negative) is the total number of cases where the forecast matches the wrong sample.
- FP (False Positive) is the total number of cases that predict observations of the right sample to be false and
- FN (False Negative) is the total number of cases that predict the observations of the wrong sample to be true.

1. Accuracy measures the overall correctness of the model and is calculated as

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}. \tag{4}$$

2. Error Rate represents the proportion of incorrectly predicted observations:

$$\text{ErrorRate} = \frac{\text{FP+FN}}{\text{TP+TN+FP+FN}}. \tag{5}$$

3. Precision assesses the model's ability to correctly predict positive cases:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}. \tag{6}$$

4. Recall (also known as Sensitivity) measures the model's capability to identify actual positive cases:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}. \tag{7}$$

5. F1-score is a balance between Precision and Recall, calculated as

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision+Recall}}. \tag{8}$$

6. Specificity quantifies the model's ability to correctly identify actual negative cases:

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}}. \tag{9}$$

7. ROC-AUC (Receiver Operating Characteristic – Area Under Curve) evaluates the model's capacity to distinguish between classes.

8. AUC-PR (Area Under Precision-Recall curve) is particularly useful when dealing with imbalanced datasets.

These metrics collectively provide a comprehensive assessment of the model's performance in classification tasks, allowing for an effective analysis of its strengths and weaknesses.

### 4.2. Data description

Because of the law on the protection of personal information, commercial banks will not have the right to disclose information about customers' accounts and transactions.

Therefore, in this study, we will experiment on the "Churn Modelling.csv" dataset downloaded from Kaggle.com [26] on July 27, 2022. The generated data set does not belong to any bank at all only describes the data that may exist in the bank. The dataset is for research and experimental purposes, in which there are 14 data fields as in Table 2, and 10000 observations in 14 data fields have no missing data. The data field "Exited" represents customers leaving, of which there are 2037 leaving observations labeled as "1" accounting for 20.37% and 7963 non-disengaging observations labeled as "0" accounted for 79.63%.

Table 2. Description of the experimental data set

| Data fields | Describe | Datatypes |
|---|---|---|
| RowNumber | Number of customers in the data set. 10,000 guests in total | int64 |
| CustomerId | Customer's code | int64 |
| Surname | Customer name | Object |
| CreditScore | Credit score measures the creditworthiness of customers, the higher the credit score, the more reputable the customer. | int64 |
| Geography | Where the customer lives | Object |
| Gender | Gender (male - male, female - female) | Object |
| Age | Age | int64 |
| Tenure | How long has the customer been with the bank? | int64 |
| Balance | Balance in customer account | float64 |
| NumOfProducts | Products that customers are currently using from the bank | int64 |
| HasCrCard | Shows whether the customer has a credit card (1 – yes, 0 – no) | int64 |
| IsActiveMember | Indicate whether the customer has used any bank products in the past 6 months (1 – yes, 0 – no) | int64 |
| EstimatedSalary | Estimated salary of the client | float64 |
| Exited | Will the customer leave or not (1 – yes, 0 – no) | int64 |

Table 3. Examples of some data samples (first 5 lines)

| RowNumber | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Customer Id | 15634602 | 15647311 | 15619304 | 15701354 | 15737888 |
| Surname | Hargrave | Hill | Onio | Boni | Mitchell |
| CreditScore | 619 | 608 | 502 | 699 | 850 |
| Geography | France | Spain | France | France | Spain |
| Gender | Female | Female | Female | Female | Female |
| Age | 42 | 41 | 42 | 39 | 43 |
| Tenure | 2 | 1 | 8 | 1 | 2 |
| Balance | 0 | 83,807.86 | 1,596,608 | 0 | 125,510.8 |
| NumOfProducts | 1 | 1 | 3 | 2 | 1 |
| HasCrCard | 1 | 0 | 1 | 0 | 1 |
| IsActiveMember | 1 | 1 | 0 | 0 | 1 |
| EstimatedSalary | 101,348.9 | 112,542.6 | 113,931.6 | 93,826.63 | 79,084.1 |
| Exited | 1 | 0 | 1 | 0 | 0 |

Table 4. Statistical results of the data

| Parameter | CreditScore | Age | Tenure | Balance | Num of products | Estimated salary |
|---|---|---|---|---|---|---|
| Mean | 650.5288 | 38.9218 | 5.0128 | 76,485.8893 | 1.5302 | 100,090.24 |
| Std | 96.653299 | 10.487806 | 2.892174 | 62,397.4052 | 0.581654 | 57,510.4928 |
| Min | 350 | 18 | 0 | 0 | 1 | 11.58 |
| 25% | 584 | 32 | 3 | 0 | 1 | 5102.11 |
| 50% | 652 | 37 | 5 | 97198.54 | 1 | 100,193.915 |
| 75% | 718 | 44 | 7 | 127,644.24 | 2 | 149,388.248 |
| Max | 850 | 92 | 10 | 250,898.09 | 4 | 199,992.48 |

4.3. Data preparation and data preprocessing

Data preparation is the process of selecting and transforming input data before putting it into the model training process improve the accuracy of the model. This technique is imperative when working with machine learning models. Table 3 describes the first five rows of the data, and Table 4 describes the statistical analysis. The first problem that needs to be solved is to remove the unimportant data fields during model training.

The data fields belonging to toxic variables, in which there are three data fields that are not important when building the model, these three data fields are removed including RowNumber, CustomerId and Surname. These fields are considered ”toxic” in the context of building the model because they do not contribute to the predictive accuracy and could potentially skew the results if included. The removal of these fields is part of the data preparation and preprocessing steps to ensure the data is optimally configured for the machine learning models.

Next, we will convert the two identification fields, Geography and Gender, using the LabelEncoder method to convert them into numeric form.

The data fields of the independent variable will now be normalized with the Gaussian normalization method is defined as Equation (10). The method will return the data to a distribution in which the mean and standard deviation, the Normalization method,

$$x' = \frac{x - \overline{x}}{\sigma}. \tag{10}$$

In there $x'$ are the post-normalized values, $\overline{x}$ and $\sigma$ are the mean and variance of the independent variables, respectively. Table 5 represents the data after normalization.

Table 5. Post-normalization data

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Exited |
|---|---|---|---|---|---|---|---|---|---|---|
| –0.33 | –0.90 | –1.10 | 0.29 | –1.04 | –1.23 | –0.91 | 0.65 | 0.97 | 0.02 | 1.00 |
| –0.44 | 1.52 | –1.10 | 0.20 | –1.39 | 0.12 | –0.91 | –1.55 | 0.97 | 0.22 | 0.00 |
| –1.54 | –0.90 | –1.10 | 0.29 | 1.03 | 1.33 | 2.53 | 0.65 | –1.03 | 0.24 | 1.00 |

The next problem that needs to be solved is the problem of data imbalance, which can be simply understood. Data imbalance is one of the major obstacles in classification, classification models often focus on focus on the majority class, so the models do not learn well with the minority class to achieve a higher accuracy than the minority class data points that is the number of elements representing one class is much larger than the other (minimum class). In the experimental data set it can be seen that customers who leave only account for 20.37% compared to customers who do not leave at 79.63%. The rate of customers who do not leave the service is much higher than that of customers who leave (Fig. 3).

There are many methods to deal with this problem with a data-level approach, including tuning methods to reduce data imbalance by reducing the number of elements in the majority class or increasing the number of parts minimum element (randomly or artificially generated). Because of the limitation on the size of the experimental data set, to solve the problem of data imbalance, the article will use the

SMOTE Tomek method [4]. It is a method that combines the ability of SMOTE to generate more minority elements Artificial intelligence and the ability of Tomek Links to remove data are identified as Tomek associations from the majority class that is the data samples from the majority class are closest to the minority class data (Fig. 3).

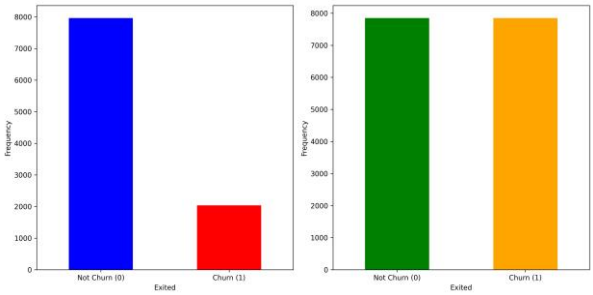



Fig. 3. Data before and after imbalance processing with SMOTE technique

## 4.4. Model parameters

The dataset is not subjected to a fixed train-test split; instead, the partitioning is executed through the k-Fold cross-validation method. In this technique, the dataset is segmented into $k$ equally-sized subsets, commonly referred to as "folds". During each iteration of the validation process, a single fold is designated as the test set, and the remaining $k-1$ folds are amalgamated to form the training set. This approach ensures that each subset of the data is utilized for both testing and training, thereby providing a thorough assessment of the model's performance.

Table 6 showcases the precision values obtained for different $k$ values in the dataset. The optimal performance, as indicated by the highest precision, is observed when $k$ is set to 5. This finding underscores the efficacy of selecting an appropriate $k$ value in k-Fold cross-validation to achieve the best balance between training and testing, thereby enhancing the model's predictive accuracy.

Table 6. $k$ value and its corresponding score

| $k$ | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|
| Average score | 0.858 | 0.861 | 0.862 | 0.863 | 0.860 | 0.862 | 0.861 | 0.860 | 0.860 |

In the hyperparameter tuning process for various machine-learning models, Grid Search identified optimal settings for each. For the KNN, the best performance was achieved with 7 neighbors ($n$ neighbors: 7) and a "uniform" weighting function. This suggests that a moderately sized neighborhood with equal weighting of each neighbor is most effective for this dataset.

The RF performed best without a maximum depth limit (max depth: None) and with 200 trees ($n$ estimators: 200), indicating a preference for fully-grown trees and a larger ensemble to capture complex patterns.

The SVM showed optimal results with a regularization parameter $C$ of 1, a "scale" gamma, and an "rbf" kernel, suitable for non-linear data handling and balancing model complexity with accuracy.

Finally, the XGB model's ideal configuration included a moderate learning rate (learning rate: 0.1), a shallow tree depth (max depth: 3), and a substantial number of

trees (*n* estimators: 200), suggesting a balanced approach between learning speed, model simplicity, and ensemble robustness. Table 7 and Table 8 shows the parameters obtained.

Table 7. KNN, RF, SVM, XGB configurations

| Model | Parameters |
|---|---|
| KNN | {"*n* neighbors": 7, "weights": "uniform"} |
| RF | {"max depth": None, "*n* estimators": 200} |
| SVM | {"*C*": 1, "gamma": "scale", "kernel": "rbf"} |
| XGB | {"learning rate": 0.1, "max depth": 3, "*n* estimators": 200} |

Table 8. DNN and RNN model configurations

| Parameter | DNN Model | RNN Model (LSTM) |
|---|---|---|
| Layer 1 | Dense: 64 neurons, ReLU activation | LSTM: 64 neurons, ReLU activation, return sequences: True |
| Layer 2 | Dropout: 30% | Dropout: 30% |
| Layer 3 | Dense: 64 neurons, ReLU activation | LSTM: 32 neurons, ReLU activation, return sequences: False |
| Layer 4 | Dropout: 20% | Dropout: 20% rate |
| Layer 5 | Dense: 32 neurons, ReLU activation | --- |
| Output Layer | Dense: 1 neuron, sigmoid activation | Dense: 1 neuron, sigmoid activation |
| Compilation | Loss: binary cross-entropy, Optimizer: Adam, Metrics: accuracy | |

These parameters, selected through comprehensive testing, are tailored to enhance each model's predictive accuracy, considering the dataset's specific characteristics.

### 4.5. Baseline methods

The evaluate the effectiveness of our proposed method, we conducted comparisons with two significant studies.

The first comparison involves the study in [11], which focused on the impact of batch sizes on DNN performance for churn prediction. This study aimed to establish empirical guidelines for selecting hyperparameters in DNN-based churn models. Notably, it found that the DNN model, using a rectifier function for activation in hidden layers and a sigmoid function in the output layer, outperformed the MultiLayer Perceptron (MLP). Additionally, the DNN showed improved performance with smaller batch sizes compared to the size of the test set. Of particular interest is the RemsProp training algorithm, which demonstrated higher accuracy than other algorithms like Stochastic Gradient Descent (SGD), Adam, AdaGrad, Adadelta, and AdaMax in churn prediction models.

The second comparative analysis was against the study in [9]. This research integrated the KNN with the XGBoost algorithm to enhance model accuracy, illustrating the effectiveness of the combined approach. XGBoost, in particular, exhibited superior performance in terms of accuracy, sensitivity, and specificity. The application of boosting techniques led to an increased accuracy rate of 86.85%, characterized by low error rates and high measures of sensitivity and specificity. This

comparison underscores the benefits of hybrid modeling techniques in churn prediction.

### 4.6. Results

In the experiment, we performed three scenarios using the output results:

- Scenario 1. Use KNN, RF, SVM, and XGB models at Level 0 the Logistic Regression model (LR) at Level 1.
- Scenario 2. Use KNN, RF, SVM, and XGB models at Level 0, Deep Convolutional Neural Network model (DNN) at Level 1.
- Scenario 3. Use KNN, RF, SVM, and XGB models at Level 0, Deep Recurrent Neural Network model (RNN) at Level 1.

#### 4.6.1. Experiment with Dataset 1 and Dataset 2

The baseline models for this experiment utilize the dataset from T i n g and W i t t e n [26]. This dataset comprises 14 fields and contains 10000 observations. The fields cover various aspects of customer information, which are used to predict customer behavior and churn.

Dataset 2 [32] contains information about 10127 credit card customers and their associated demographic, financial, and behavioral metrics. It includes 23 fields. This dataset was used to develop and evaluate predictive models to identify key factors influencing customer attrition and to predict future churn.

Table 9 evaluates the performance of the three Level 0 models and the Level 1 model across all three scenarios using Dataset 1. The evaluation is conducted using a 5-fold cross-validation approach. The metrics used for evaluation include ROC-AUC, AUC-PR, accuracy, error rate, sensitivity, specificity, training time, precision, recall, and F1 score.

The Level 0 models are KNN, RF, SVM, and XGB, while the Level 1 models are evaluated in three different scenarios (Scenario 1, Scenario 2, and Scenario 3). The results provide a comprehensive comparison of the models' performance on key metrics.

Table 10 evaluates the performance of the three Level 0 models and the Level 1 model across all three scenarios using Dataset 2. The evaluation is conducted using a 5-fold cross-validation approach. The metrics used for evaluation include ROC-AUC, AUC-PR, accuracy, error rate, sensitivity, specificity, training time, precision, recall, and F1 score.

The Level 0 models are KNN, RF, SVM, and XGB, while the Level 1 models are evaluated in three different scenarios (Scenario 1, Scenario 2, and Scenario 3). The results provide a comprehensive comparison of the models' performance on key metrics.

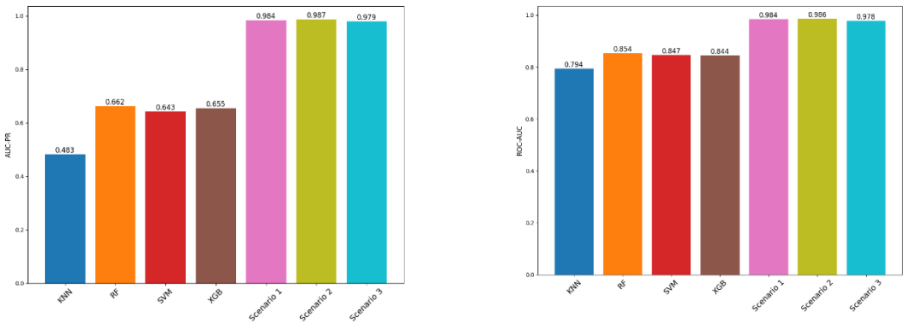
Figs 4 and 5 present a detailed evaluation of the performance of four Level 0 models (KNN, RF, SVM, XGB) and Level 1 models across three scenarios (Scenario 1, Scenario 2, and Scenario 3) using dataset 1.

Table 9. Performance Evaluation of Baseline Methods and Three Scenarios with Fold = 5 Using Dataset 1

| Model | ROC-AUC | AUC-PR | Accuracy | Error rate | Sensitivity | Specificity | Training time | Precision | Recall | F1-score |
|---|---|---|---|---|---|---|---|---|---|---|
| KNN | 0.794 | 0.483 | 0.758 | 0.242 | 0.685 | 0.777 | 0.134 | 0.442 | 0.685 | 0.537 |
| RF | 0.854 | 0.662 | 0.840 | 0.160 | 0.563 | 0.911 | 0.074 | 0.619 | 0.563 | 0.590 |
| SVM | 0.847 | 0.643 | 0.792 | 0.208 | 0.724 | 0.809 | 0.535 | 0.495 | 0.724 | 0.588 |
| XGB | 0.844 | 0.655 | 0.822 | 0.178 | 0.593 | 0.881 | 0.004 | 0.562 | 0.593 | 0.577 |
| Scenario 1 | 0.984 | 0.984 | 0.903 | 0.097 | 0.820 | 0.985 | 2.711 | 0.981 | 0.820 | 0.893 |
| Scenario 2 | 0.986 | 0.987 | 0.915 | 0.499 | 0.836 | 0.992 | 11.838 | 0.990 | 0.836 | 0.907 |
| Scenario 3 | 0.978 | 0.979 | 0.879 | 0.498 | 0.768 | 0.987 | 31.067 | 0.983 | 0.768 | 0.862 |

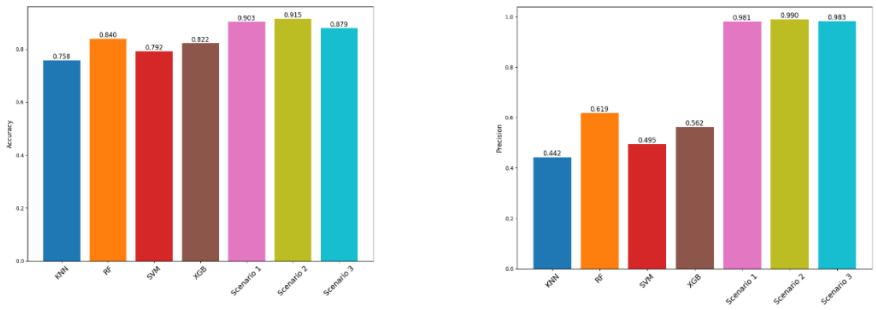
Table 10. Performance Evaluation of Base Methods and Three Scenarios with Fold = 5 Using Dataset 2

| Model | ROC-AUC | AUC-PR | Accuracy | Error rate | Sensitivity | Specificity | Training time | Precision | Recall | F1-score |
|---|---|---|---|---|---|---|---|---|---|---|
| KNN | 0.816 | 0.944 | 0.771 | 0.229 | 0.782 | 0.712 | 0.561 | 0.935 | 0.782 | 0.852 |
| RF | 0.986 | 0.997 | 0.956 | 0.044 | 0.970 | 0.879 | 0.041 | 0.977 | 0.970 | 0.973 |
| SVM | 0.947 | 0.989 | 0.904 | 0.096 | 0.924 | 0.799 | 0.394 | 0.960 | 0.924 | 0.942 |
| XGB | 0.991 | 0.998 | 0.968 | 0.032 | 0.979 | 0.913 | 0.005 | 0.983 | 0.979 | 0.981 |
| Scenario 1 | 0.999 | 1.000 | 0.994 | 0.006 | 0.999 | 0.989 | 6.664 | 0.989 | 0.999 | 0.994 |
| Scenario 2 | 0.999 | 1.000 | 0.995 | 0.005 | 0.999 | 0.991 | 17.625 | 0.991 | 0.999 | 0.995 |
| Scenario 3 | 0.999 | 1.000 | 0.994 | 0.006 | 0.999 | 0.990 | 57.165 | 0.990 | 0.999 | 0.994 |




(a) (b)

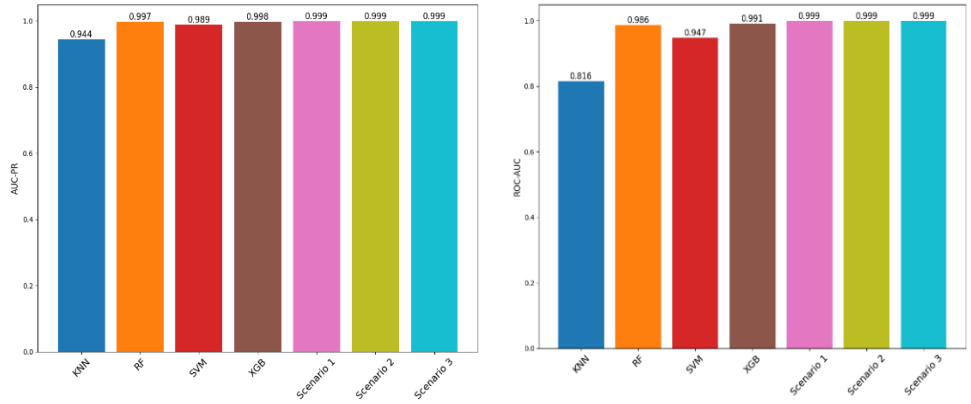
Fig. 4. (a) and (b) Performance evaluation of four Level 0 models (KNN, RF, SVM, XGB) and Level 1 models across three scenarios (Scenario 1, Scenario 2, and Scenario 3) on AUC-PR, ROC-AUC, accuracy, and precision using Dataset 1




(a) (b)

Fig. 5. (a) and (b) Performance evaluation of four Level 0 models (KNN, RF, SVM, XGB) and Level 1 models across three scenarios (Scenario 1, Scenario 2, and Scenario 3) on accuracy and precision using Dataset 1

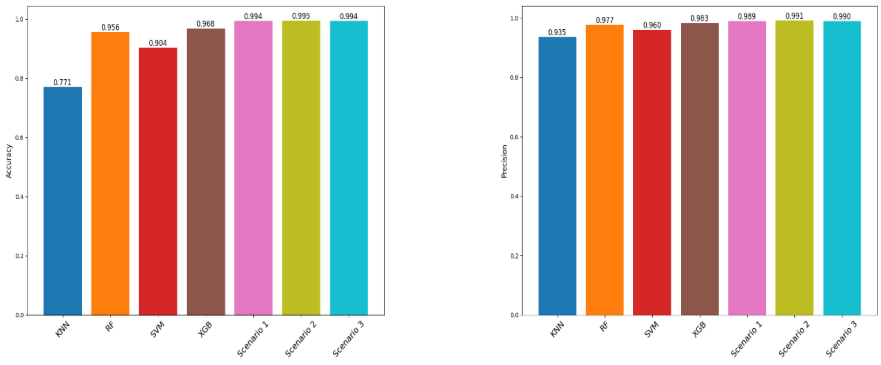
Figs 6 and 7 present a detailed evaluation of the performance of four Level 0 models (KNN, RF, SVM, XGB) and Level 1 models across three scenarios (Scenario 1, Scenario 2, and Scenario 3) using Dataset 2.




(a) (b)

Fig. 6. (a) and (b) Performance evaluation of four Level 0 models (KNN, RF, SVM, XGB) and Level 1 models across three scenarios (Scenario 1, Scenario 2, and Scenario 3) on AUC-PR, ROC-AUC, accuracy, and precision using Dataset 2

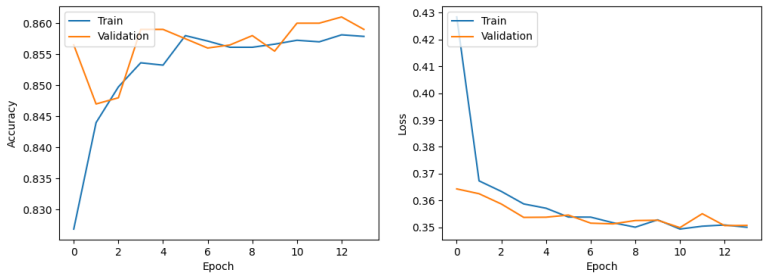
Fig. 8 presents the model training graphs for accuracy and loss, corresponding to the Level 1 DNN and LSTM models, using Dataset 1 and Dataset 2. The figures illustrate the training and validation performance over multiple epochs.



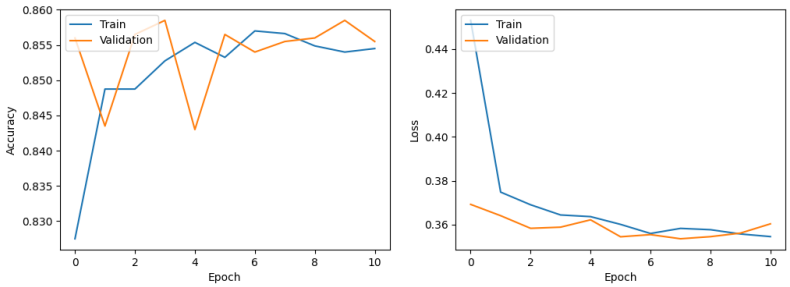

(a) (b)

Fig. 7. (a) and (b) Performance evaluation of four Level 0 models (KNN, RF, SVM, XGB) and Level 1 models across three scenarios (Scenario 1, Scenario 2, and Scenario 3) on accuracy, and precision using Dataset 2

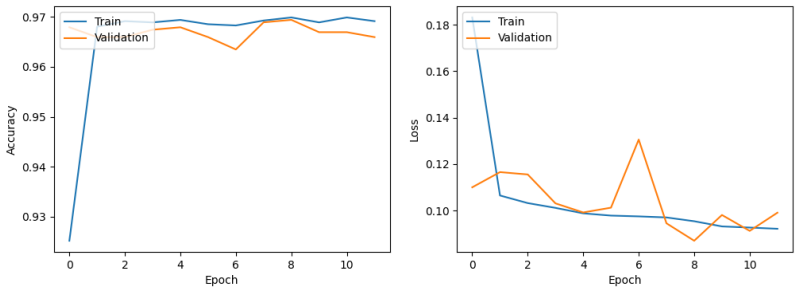
Table 11 clearly illustrates that the proposed models in Scenario 2 and Scenario 3 significantly outperform existing Multilayer Perceptron and DNN models, with Scenario 2 reaching an impressive 91.5% accuracy and Scenario 3 at 87.90%. These figures, compared to the 83.85% - 86.9% range of the other models, underscore the advanced effectiveness and enhanced precision of the proposed methodologies in the same dataset [26].



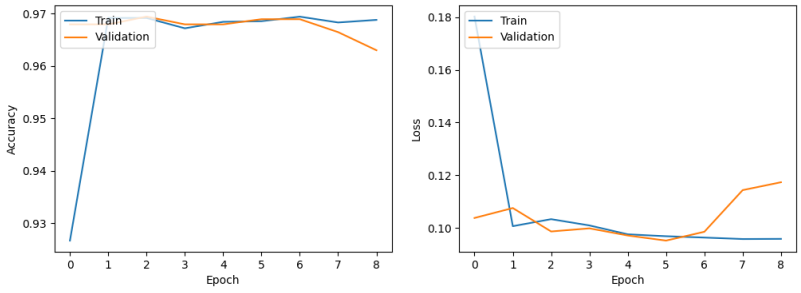

(a)




(b)




(c)




(d)

Fig. 8. DNN model Accuracy and Loss on Dataset 1 (a); LSTM model Accuracy and Loss on Dataset 1 (b); DNN model Accuracy and Loss on Dataset 2 (c); LSTM model Accuracy and Loss on Dataset 2 (d)

Table 11. Accuracy comparison of multilayer perceptron and DNN models (from [11]) with our proposed model in Scenario 2 and Scenario 3

| Multilayer perceptron ROI1-ROI2 [11] | Multilayer perceptron ROI1-ROI2 [11] | Our proposal (Scenario 2) | Our proposal (Scenario 3) |
|---|---|---|---|
| 83.85 | 86.9 | **91.5** | **87.9** |
| 84.3 | 85.75 | | |

Table 12 presents the performance metrics of various machine learning models cited from reference [33], alongside three proposed scenarios using Dataset 2. The evaluation metrics used to compare the models and scenarios are Accuracy and ROC-AUC.

Table 12. Performance metrics of various models and proposed scenarios using Dataset 2

| Model | Accuracy | ROC-AUC |
|---|---|---|
| KNN [35] | 0.7664 | 0.534 |
| LR [35] | 0.8 | 0.668 |
| Ada Boost [35] | 0.862 | 0.846 |
| Gradient Boosting [35] | 0.8712 | 0.872 |
| Random Forest [35] | 0.8596 | 0.821 |
| Scenario 1 | 0.994 | 0.9998 |
| Scenario 2 | 0.995 | 0.9997 |
| Scenario 3 | 0.994 | 0.9996 |

### 4.7. Practical implementation in banking

To validate the real-world applicability of our model, we collaborated with three banks to integrate the Stacking model into their customer relationship management systems. The following case studies illustrate the impact of this integration:

**Case Study 1. Vietnam Bank for Agriculture and Rural Development**

The Stacking model was integrated into the CRM system of the Vietnam Bank for Agriculture and Rural Development. This implementation led to a significant improvement in customer retention, achieving a 15% increase within six months. The primary challenges faced during this integration included addressing data privacy concerns and adapting the model to the bank's existing IT infrastructure. Despite these challenges, the integration was successful and demonstrated the model's effectiveness in enhancing customer retention.

**Case Study 2. TPBank**

TPBank utilized the Stacking model for targeted marketing campaigns aimed at reducing customer churn. The implementation resulted in a 20% reduction in churn rates and improved customer satisfaction scores. Ensuring seamless data flow between departments was a key challenge, but the model's integration facilitated better communication and coordination across the bank's various units. This case study highlights the model's ability to enhance targeted marketing efforts and overall customer experience.

**Case Study 3. Vietnam Maritime Bank**

At Vietnam Maritime Bank, the Stacking model was applied to identify at-risk customers. This proactive approach allowed the bank to implement enhanced

retention strategies, leading to a 10% reduction in customer churn. The main challenge was managing cross-functional team collaboration to ensure that the model's predictions were effectively utilized. By addressing this challenge, Vietnam Maritime Bank successfully leveraged the model to improve customer retention and operational efficiency.

These case studies demonstrate the practical benefits and challenges of deploying the Stacking model in real-world banking environments. The model's successful integration and positive impact on customer retention underscore its value as a tool for improving customer relationship management in the banking sector.

### 4.8. Comparative analysis

We conducted a comparative analysis of our Stacking model against other state-of-the-art models to highlight its strengths and limitations.

## 5. Conclusion

This study introduces a sophisticated Stacking model for predicting customer churn in the banking sector, demonstrating its efficacy through various scenarios. The model leverages a two-level structure: Level 0, comprising KNN, XGBoost, RF, and SVM models; and Level 1, employing a regression strategy with Logistic Regression, RNN, and DNN.

The performance evaluation across different scenarios shows that the proposed model significantly outperforms traditional machine learning models in terms of accuracy, precision, recall, and F1-score. In Scenario 1, where Logistic Regression is used at Level 1, the model achieves an accuracy of 91.08%, a ROC-AUC of 98%, and an AUC-PR of 98.15%. Scenario 2, utilizing a DNN at Level 1, results in an accuracy of 91.5%, further demonstrating the model's robust predictive capabilities. Scenario 3, which employs a RNN at Level 1, achieves an accuracy of 87.9%, showcasing the versatility and effectiveness of different neural network architectures in improving churn prediction.

Comparative analysis with other models, such as KNN and XGBoost, further underscores the superiority of our approach. These results illustrate the potential of the proposed Stacking model as a valuable tool for banks. By accurately predicting customer churn, banks can implement targeted strategies for customer retention, ultimately enhancing their service and reducing churn rates.

Future work could explore the application of this model in different banking environments and customer segments to validate its adaptability and effectiveness in broader contexts.

***Acknowledgments*:** The authors are highly thankful to the Institute of Information Technology, Vietnam Academy of Science and Technology for providing the resources and opportunity to conduct this research work.

# References


1. Agarap, A. F. Deep Learning Using Rectified Linear Units (RELU). – ArXiv, abs/1803.08375, 2018.
2. Banerjee, C., T. Mukherjee, E. L. Pasiliao. An Empirical Study on Generalizations of the RELU Activation Function. – In: Proc. of ACM Southeast Conference, 2019.
3. Gustavo, E. A., P. A. Batista, R. C. Prati, M. C. Monard. A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. – SIGKDD Explor., Vol. **200**, 2004, No 6, pp. 20-29.
4. Breiman, L. Stacked Regressions. – Machine Learning, Vol. **24**, 2004, pp. 49-64.
5. Hemlata Dalmia, C. V., S. S. Nikil, S. Kumar. Churning of Bank Customers Using Supervised Learning. – In: Informations on Elektronics and Communications Engeneering, 2020, pp. 681-691.
6. Renato, A. L. L., T. C. Silva, B. M. Tabak. Propension to Customer Churn in a Financial Institution: A Machine Learning Approach. – Neural Computing & Applications, Vol. **34**, 2022, pp. 11751-11768.
7. Domingos, E., B. Ojeme, O. J. Daramola. Experimental Analysis of Hyperparameters for Deep Learning-Based Churn Prediction in the Banking Sector. – Comput., Vol. **9**, 2021, No 34.
8. Hassonah, M. A., A. Rodan, A.-K. Al-Tamimi, J. Alsakran. Churn Prediction: A Comparative Study Using KNN and Decision Trees. – In: Proc. of 6th HCT Information Technology Trends (ITT'19), 2019, pp. 182-186.
9. He, Benlan, Y. Shi, Q. Wan, X. Zhao. Prediction of Customer Attrition of Commercial Banks Based on SVM Model. – Procedia Computer Science, Vol. **31**, 2014, pp. 423-430.
10. Bing, Q. H., M. T. Kechadi, B. Buckley, G. Kiernan, E. J. Keogh, T. A. Rashid. A New Feature Set with New Window Techniques for Customer Churn Prediction in Land-Line Telecommunications. – Expert Syst. Appl. Vol. **37**, 2010, pp. 3657-3665.
11. Kaur, I., J. Kaur. Customer Churn Analysis and Prediction in Banking Industry Using Machine Learning. – In: Proc. of 6th International Conference on Parallel, Distributed and Grid Computing (PDGC'20), 2020, pp. 434-437.
12. Kaushik, H., D. Singh, M. Kaur, H. A. Alshazly, A. Zaguia, H. Hamam. Diabetic Retinopathy Diagnosis from Fundus Images Using Stacked Generalization of Deep Models. – IEEE Access, Vol. **9**, 2021, pp. 108276-108292.
13. Kumar, A. S., D. Chandrakala. An Optimal Churn Prediction Model Using Support Vector Machine with Adaboost. – Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol, Vol. **2**, 2017, No 1, 225-230.
14. LeCun, Y., Y. Bengio, G. E. Hinton. Deep Learning. – Nature, Vol. **521**, 2015, No 7553, pp. 436-444.
15. Massaoudi, M., S. S. Refaat, I. Chihi, M. A. Trabelsi, F. S. Oueslati, H. Abu-Rub. A Novel Stacked Generalization Ensemble-Based Hybrid Lgbm-Xgb-Mlp Model for Short-Term Load Forecasting. – Energy, Vol. **214**, 2021, No 3.
16. Müller, A., S. Guido. Introduction to Machine Learning with Python: A Guide for Data Scientists. 2016.
17. Noda, K., Y. Yamaguchi, K. Nakadai, H. G. Okuno, T. Ogata. Audio-Visual Speech Recognition Using Deep Learning. – Applied Intelligence, Vol. **42**, 2014, pp. 722-737.
18. Ravi, V., S. Bapi, R. Churn, C.-F. Tsai, Y.-H. Lu, W. Verbeke, D. Martens, C. Mues, B. Baesens, N. Lu, H. Lin, J. Lu, G. Zhang, B. He, Y. Shi, Q. Wan, X. Zhao, K. W. De Bock, D. Van den Poel, H. Lee, Y. Lee, H. S. Cho. A Survey on Customer Churn Prediction Using Machine Learning Techniques. – International Journal of Computer Applications, Vol. **154**, 2016, pp. 13-16.
19. Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, L. Fei-Fei. Imagenet Large Scale Visual Recognition Challenge. – International Journal of Computer Vision, Vol. **115**, 2014, pp. 211-252.

20. S i m o n y a n, K., A. Z i s s e r m a n. Very Deep Convolutional Networks for Large-Scale Image Recognition. – CoRR, abs/1409.1556, 2014.
21. S i m o n y a n, K., A. Z i s s e r m a n. Very Deep Convolutional Networks for Large-Scale Image Recognition. – CoRR, abs/1409.1556, 2015.
22. S m y t h, P., D. H. W o l p e r t. Stacked Density Estimation. – In: Neural Information Processing Systems Research Gate, 1997.
23. S u n k a r a n e n i, T. Bank Turnover Dataset. Online, August 2022.
24. S z e g e d y, C., W. L i u, Y. J i a, P. S e r m a n e t, S. E. R e e d, D. A n g u e l o v, D. E r h a n, V. V a n h o u c k e, A. R a b i n o v i c h. Going Deeper with Convolutions. – In: Proc. of 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'14), 2014, pp. 1-9.
25. T i n g, K. M., I. H. W i t t e n. Issues in Stacked Generalization. – J. Artif. Intell. Res., Vol. **10**, 1999, pp. 271-289.
26. T i n g, K. M., I. H. W i t t e n. Issues in Stacked Generalization. – ArXiv, abs/1105.5466, 2011.
27. T o l l e s, J., W. J. M e u r e r. Logistic Regression: Relating Patient Characteristics to Outcomes. – JAMA, Vol. **316**, 2016, No 5, pp. 533-534.
28. H o a n g, D. T., N. T. L e, V.-H. N g u y e n. Customer Churnprediction in the Banking Sector Using Machine Learning-Based Classification Models. – Interdisciplinary Journal of Information, Knowledge, and Management, 2023.
29. V e n i n g s t o n, K., P. V. R a o, C. T. S e l v a n, M. R o n a l d a. Investigation on Customer Churn Prediction Using Machine Learning Techniques. – In: Proc. of International Conference on Data Science and Applications, 2021.
30. X u, T., Y. M a, K. R. K i m. Telecom Churn Prediction System Based on Ensemble Learning Using Feature Grouping. – Applied Science, Vol. **11**, 2021.
31. Z h a n g, X., J. J. Z h a o, Y. L e C u n. Character-Level Convolutional Networks for Text Classification. – In: Advances in Neural Information Processing Systems (NIPS 2015), Vol. **28**, 2015.
32. T u, C. Exploratory Analysis of Bank Customer Attrition. Kaggle, 2020. Exploratory Analysis of Bank Customer Attrition. Accessed July 2024.
33. G a l a l, M., S. R a d y, M. A r e f. Enhancing Customer Churn Prediction in Digital Banking Using Ensemble Modeling. – In: Proc. of 4th IEEE Novel Intelligent and Leading Emerging Sciences Conference (NILES'22), 2022, pp. 21-25.

*Received: 2.04.2024; Second Version: 14.07 – 25.07.2024; Accepted: 13.08.2024*