ABUAD Journal of Engineering Research and Development (AJERD)
ISSN (online): 2645-2685; ISSN (print): 2756-6811

Volume 8, Issue 3, 69-80

Detection of Bank Customer Churn Using Neural Network and Voting Classifier Ensembles

Abdulrahman OLAJIDE, Isaac ELESEMOYO, Habeeb ADEROGBA, Edimaobong ISAAC

Department of Computer Engineering, Elizade University, Ilara-Mokin, Nigeria
abdulrahman.olajide@elizadeuniversity.edu.ng/isaac.elesemoyo@elizadeuniversity.edu.ng/habeeb.aderogba@elizadeuniversity.edu.ng/edimaobong.isaac@elizadeuniversity.edu.ng

Corresponding Author: abdulrahman.olajide@elizadeuniversity.edu.ng, +2347062066898
Received: 16/04/2025
Revised: 30/09/2025
Accepted: 12/10/2025
Available online: 25/10/2025

Abstract: *Customer churn is the loss of business clients to a competitor. Since keeping current clients is more economical than finding new ones, customer retention measures such as churn detection are now essential aspects of modern banking strategy. However, many existing studies rely heavily on conventional machine learning approaches such as Support Vector Machines, Logistic Regression, Random Forest, etc., often neglecting the deeper learning capabilities of neural networks. Also, the repeated use of the same small dataset by the banking studies may limit the improvement of the models' generalisation. To address these gaps, this study presents a method that integrates deep learning for customer churn detection and a soft and hard voting classifier ensemble embedded with the best performing models over the years for results comparison, supported by a synthetic data augmentation method for model improvement. The study utilised a secondary banking churn dataset from Kaggle, which contained 10,000 unique customer records. To address the dataset limitations, a Conditional Tabular Generative Adversarial Network (CTGAN) model was used to generate an additional 10,000 records, expanding the dataset used for the study to 20,000 rows. Data preprocessing steps were done before training, including oversampling using Synthetic Minority Oversampling Technique (SMOTE). Model development and analysis processes were implemented using Python programming language with prominent libraries and frameworks on Google Colab. In this study, a Feedforward neural network and a soft and hard voting classifier were developed. The voting classifier ensembles integrated three prominent classifiers: Random Forest, XGBoost, and Logistic Regression. The performances were evaluated using Accuracy, F1 Score, and Area Under ROC Curve as metrics. Results show that while the Feedforward Neural Network achieved strong predictive performance with an accuracy of 88.23%, an F1 Score of 87.83% and an AUC of 94.73%, the ensemble approaches performed slightly better as the soft voting classifier delivered the best results, obtaining an accuracy of 89.46%, F1 Score of 88.92% and AUC of 95.40% showing the advantage of combining multiple models to leverage complementary strengths. After comparison with past studies, the proposed models did not surpass the very best outcomes. However, they remain highly competitive, achieving performance levels that are on par with or exceed many earlier works. The contribution of this work is to show how synthetic data augmentation, enhanced preprocessing, deep learning techniques, and machine learning ensembles can improve churn detection in banking studies. Banking institutions can utilise the results from this study to accurately detect churn, supporting proactive customer retention strategies, targeted marketing, and personalised financial services, thereby reducing revenue losses.*

Keywords: *Churn in Banks, Churn Detection, Ensemble Machine Learning, Neural Networks, Voting Classifiers, CTGAN*

1. INTRODUCTION

The term "churn" refers to the number of loyal clients who abandon their relationship with a business or service provider. It presents a big obstacle in the banking sector, affecting their market share and profitability [1]. Churn studies and modelling have become significantly prevalent since it is far more costly to bring on new clients than to keep the existing ones, making it a critical focus area for financial institutions [2]. The ability to predict customer churn accurately allows banks to implement retention measures and personalise customer service, ultimately reducing revenue loss [2]. In the realm of predictive analytics, machine learning methods have become the preferred instrument for identifying patterns and predicting future outcomes, making them highly viable in churn predictions [3, 4]. The rise of diverse data (data sourced and merged from different sources), which includes manually entered customer information, operational interactions and external data sources, has changed how banks use deep learning and machine learning for churn prediction, making it possible to now make more reliable models [4].

For years, customer churn and retention studies have made machine learning methods a prominent focus in churn prediction strategies, comparing algorithms such as k-nearest neighbours, support vector machines, and logistic regression, while ignoring the intricate learning abilities of neural networks. Existing solutions seldom explore deep learning methods in churn studies, this can be seen in research by Imani *et al.*, [5], showing that only about five out of forty studies published annually between 2020 and 2024 employed deep learning methods. New studies have shown that better results can be achieved using neural networks due to the availability of more structured data.

Additionally, many existing papers employ individual models, some of which have already achieved exceptional results. These traditional rule-based and statistical churn detection models often don't capture customer behaviour complexities adequately. Contrarily, decision trees, boosting algorithms, random forest, support vector machines, and other ensemble learning approaches all have shown a better predictive capability when modelled with structured datasets [5, 6]. These methods reduce overfitting and variance, which are limitations of single statistical models [7]. Better results can be achieved if these models are nested into a hybrid model, leveraging the advantages of each. Another observation is the repeated use of relatively the same dataset by the banking churn studies.

Considering these, this study proposes a comparative approach that addresses these limitations highlighted. The major objective is to develop a Conditional Tabular Generative Adversarial Network (CTGAN) for data augmentation then develop a multi layered Feedforward deep neural network and compare the results with a hard and soft voting classifier embedded with the best performing individual models (XGBoost classifier, logistic regression and random forest classifier) over the recent years from existing research. The performance will be analysed using F1 score, Area Under the Receiver Operating Characteristics Curve (AUC), and accuracy classification metrics. The geographic focus of the study is not limited to a specific region, commercial banks that operate in highly competitive markets, where customer retention is critical, will find the paper relevant.

The article's remaining content is organised as follows: Section 2 contains the literature review section, where we review existing approaches to churn prediction and compare individual results, outlining each strategy's strengths and drawbacks to show the expected advantages over existing studies. In Section 3, a detailed explanation of the methodology used is provided, including the properties of the datasets, steps in preprocessing, and characteristics of the models. Section 4 shows the results obtained in terms of the evaluation metrics, a discussion of these results and a detailed comparison with previous works. Section 5 concludes the work, showing the importance, recommendations and limitations of the proposed model

2. LITERATURE REVIEW

2.1 Related Works

Several techniques have been employed in churn prediction, including the use of individual algorithms and a combination of distinct algorithms. In a study by Rahman & Kumar [8], the paper proposed a model that predicts bank clients' churn using some machine learning techniques. The findings show that customer retention is more cost effective compared to customer acquisition, which highlights the need for a churn prediction system. The study used a dataset obtained from Kaggle, which contained 10,000 unique bank clients' information with 13 input features and one output label. The dataset was pre-processed using a minimum redundancy maximum relevance filter and a relief algorithm filter before random oversampling. The data was split into two parts, with 70% set out for training and 30% for testing. Four classification algorithms were used: Support Vector Machine, Decision Tree, K-Nearest Neighbour, and Random Forest, for an adequate comparison of results. Accuracy was used to evaluate the models, and the Random Forest model had the best results, achieving 95.74% accuracy after oversampling. The study's results also concluded that feature selection algorithms have little to no impact on tree classifiers, as they decrease the prediction score of these classifiers.

In another study by Muneer *et al.* [9], the authors developed a credit card customer churn detection procedure using three astute machine learning algorithms: Adaboost, Support Vector Machine and Random Forest. A dataset of 10,127 records (rows) was used, which contained 83.9% non-churners and 16.1% churners, with 20 input features and one output label. To address the high dataset imbalance, a combination of oversampling and undersampling techniques was employed. The classifiers were evaluated using recall, accuracy, F1 score, and fold cross-validation. The Random Forest classifier performed best, leading the other models in three of the four metrics. It achieved the highest F1 score of 91%, an accuracy of 88.7%, and the highest F1 validation score after a 5-fold cross-validation. A perfect recall was attained with the SVM model. The study's results aim to calculate credit card customers' churn periodically from various perspectives, providing valuable insights into the future. It is suggested that further research be conducted on a deep learning model to enhance the accuracy of the proposed study.

Singh *et al.* [10] underscore the drawbacks of customer attrition in the financial sector, highlighting the need for detection and retention strategies, as well as the advantages of their implementation, such as customised banking solutions. The study proposed a concise preprocessing technique that effectively captures and unifies data in a consistent format. Support Vector Machine, Random Forest, XGBoost and Logistic Regression models were trained on this dataset to predict churn. The results were examined using recall, F1 score, accuracy, the Area under the ROC curve and precision. The XGBoost performed best among the other models, with 83.9% accuracy, 81.3% F1 score, and 84.7% AUC after oversampling, followed closely by the Random Forest classifier. A functional application that provides extensive visualisations and insights on data to ensure improved decision making was also designed.

Another study by Lalwani *et al.* [11] also attempted to capture churning using several machine learning algorithms, including Naive Bayes, Support Vector Machines, Random Forest classifier, Decision Trees, Logistic Regression, and some boosting algorithms, such as XGBoost, AdaBoost, and CatBoost, and compared the results. The research was carried out to analyse each model's performance on a telecommunications customer dataset pre-processed using a gravitational search algorithm. The models were evaluated using a confusion matrix and the Area under the ROC curve. The Adaboost classifier performed best, with an accuracy of 81.71%, an F1 score of 80.28%, and a tied AUC score of 84% with the XGBoost classifier. Ullah *et al.* [12] also proposed a unique model using clustering and classification techniques

specifically for factor identification and churn detection. Models employed include Random Tree, Random Forest, Logistic Regression, Naive Bayes, AdaBoost, and others. The models were evaluated using confusion matrix parameters and the Area under the ROC curve. Through this evaluation, the random forest method fared best, with an accuracy of 88.63%, an 88% F1 score, and a 95.9% AUC. Table 1 summarises selected works, highlighting their datasets, applied models, and the best performing ones.

Customer churn detection/prediction has been studied extensively over the years in various important industries, such as banking, telecommunications, insurance, gaming, and e-commerce, due to its benefits. The application of artificial intelligence methods (deep learning and machine learning) has been invaluable in these studies. Typical approaches involve a heavy reliance on supervised statistical machine learning methods, such as Naive Bayes, Logistic Regression, Support Vector Machines, and K-nearest neighbours, among others [5, 8, 12, 13]. As customer data became more complex, researchers began to explore ensemble techniques, such as random forest and gradient boosting methods, including Extreme Gradient Boost (XGBoost), AdaBoost, and CatBoost [6, 9, 10, 11], which demonstrate superior performance compared to statistical models. In parallel, deep learning/neural network architectures have been neglected and are now gaining more attention due to the availability of more data, and they now achieve improved success rates when compared to their machine learning counterparts [14]. In a study by Çelik & Osmanoğlu [13] where several churn detection papers were reviewed, the authors concluded that improved and higher success rates can be achieved with deep learning techniques if the training data is balanced, larger, and error-free (sufficient preprocessing should be done). In another study by Geiler *et al*., [6], after a survey on the popular best performing machine learning models, a conclusion was derived that nested ensemble approaches are better for classification tasks (e.g. churn detection) as these methods outperform traditional ensemble and individual statistical models. These nested ensembles (e.g., voting classifier ensembles) provide a hybrid approach that creates a supermodel by leveraging the strengths of all the individual models it is composed of. Despite the advances, many studies still emphasise the class imbalance problem, often leading to a biased model, as the majority of datasets are lopsided, having larger percentages of non-churned customers than churned customers. Techniques such as Adaptive Synthetic Sampling (ADASYN), Synthetic Minority Over-sampling Technique (SMOTE), resampling, and stratified sampling have been employed to tackle this issue. The results comparison has been conclusive so far, indicating the need for an oversampling technique during data processing in every study. As shown in Table 1, a common limitation identified in the banking churn studies is the repeated use of relatively the same dataset consisting of approximately 10,000 customer records. While these datasets have provided useful insights, their restricted size limits the robustness and generalizability of the predictive models. To tackle this issue, this research proposes the application of a Conditional Tabular Generative Adversarial Network (CTGAN) to supplement data synthetically and evaluate whether it improves the performance of the models.

From the review, it is evident that both deep learning and nested ensemble-based approaches offer significant advantages in churn prediction. Working on these insights, the proposed study will build a multilayered Feedforward neural network and a soft and hard voting classifier ensemble containing the best performing individual machine learning models from the review of past studies to leverage the individual strengths of these models, and a comparison study will be done. The two approaches are expected to deliver improved predictive accuracy and a more generalised model across various datasets.

Table 1: Related works comparison

Author	Year	Sector	Dataset Size	Models Used	Best Performing Model
Rahman & Kumar [8]	2020	Banking	10,000 rows	KNN, SVM, DT, RF	Random Forest
Muneer *et al.*, [9]	2022	Banking	10,127 rows	RF, SVM, AdaBoost	Random Forest
Singh *et al.*, [10]	2024	Banking	10,000 rows	RF, LR, SVM, XGBoost	XGBoost
Lalwani *et al.*, [11]	2021	Telecoms	7,000 rows	LR, DT, KNN, RF, NB, SVM, AdaBoost, XGBoost, CatBoost	AdaBoost & XGBoost
Ullah *et al.*, [12]	2019	Telecoms	64,107 + 3,333 rows	RT, J48, RF, NB, LR, AdaBoost…	Random Forest

2.2 Advantages of the Proposed Technique over Existing Techniques

i. Development of a Conditional Tabular Generative Adversarial Network (CTGAN) to supplement data synthetically, expanding the initial dataset of 10,000 rows to 20,000 rows, thereby enriching the training data available for model development
ii. Implementation of a careful feature selection method and adequate preprocessing, including removing high correlation features to avoid overfitting, in contrast to most studies, where this was not considered.
iii. Use of a deep learning technique for models' comparison as opposed to most studies, where only machine learning models are used.
iv. Use of a nested ensemble approach to combine the strengths of the best performing models in past studies to make a better performing hybrid model.

3. METHODOLOGY

The base dataset used in this work was a secondary dataset obtained from Kaggle for churn modelling applications [15]. The dataset contains 10,000 bank customer records containing demographic, behavioural, and transactional attributes with 17 columns of distinct features, and one target label column 'exited' that indicates if the customer left or not. The dataset description is presented in Table 2. All the model development and analysis were implemented using Python programming language on a free tier Google Colab IDE, which is a cloud based Jupyter notebook. The free tier option grants access to a cloud server with approximately 100 Gigabytes of ROM storage, 12 Gigabytes of RAM storage, and 15 Gigabytes of Nvidia T4 tensor core GPU storage, providing an average of 4 hours of runtime. The study leveraged well known machine learning libraries and frameworks, including Scikit-learn for model training and evaluation, Pandas and NumPy for data preprocessing and manipulation, Matplotlib and Seaborn for visualisation, and CTGAN (Conditional Tabular Generative Adversarial Network) to supplement data synthetically. CTGAN is a generative model designed specifically for tabular datasets with mixed data types, enabling the creation of synthetic data samples that preserve both the statistical distributions and complex dependencies of the original data. The system architecture, which shows the order of steps taken, is shown in Figure 1.

In this study, after the dataset was imported, the identifier columns were removed, and a list with column names for categorical variables was defined. This list will be passed to the model, allowing it to determine how to process these fields. The CTGAN model was then initialised with suitable hyperparameters and trained iteratively for 200 epochs until convergence, allowing the generator to learn the underlying feature distributions. After successful training, the model was used to generate a sample of 10,000 additional records, which were then merged with the original dataset to form an expanded dataset of 20,000 rows for the study. To assess the resulting dataset's quality, the TableEvaluator framework was employed, which provides multiple visualisation techniques for comparing the real and synthetic data to check if the statistical properties and feature distributions are preserved. The Principal Component Analysis (PCA) plots shown in Figure 2 project both the real and synthetic datasets into two principal components, providing a visual comparison of their global structures. A similar scatter distribution is observed between the two plots, indicating that the overall data structure has been preserved. The absolute log mean and standard deviation plots illustrated in Figure 3 evaluate how well the model captures the central tendencies and variability of the numeric features. Each point on the plots represents the features, and the diagonal line indicates perfect agreement between the real and synthetic statistics. The plots show the points close to the diagonal, indicating strong alignment with little inconsistencies. The data was also evaluated using a correlation heatmap difference plot, distribution per feature plots and cumulative sum per feature plots, and the data performed well on all these metrics.

Table 2: Dataset description

Feature Name	Description	Type
RowNumber	Corresponds to the record (row) number	Number
CustomerId	A unique number assigned to identify each customer.	Number
Surname	The customer's family name.	Text
CreditScore	A numeric value that reflects the customer's credit reliability.	Number
Geography	The customer's location	Text
Gender	Specifies whether the customer is male or female.	Text
Age	The customer's age, measured in years.	Number
Tenure	Represents the years that the client has held an account with the bank.	Number
Balance	The amount of money currently in the customer's account.	Number
NumOfProducts	Number of banking products held by the customer (e.g., cards, loans).	Number
HasCrCard	Indicates if the customer possesses a credit card.	Number
IsActiveMember	Indicates if the client is making use of the bank's services.	Number
EstimatedSalary	The customer's projected yearly income.	Number
Complain	Whether the customer has filed a complaint or not.	Number
Satisfaction Score	A rating that reflects the customer's satisfaction level.	Number
Card Type	Specifies the type of card the customer uses.	Text
Points Earned	Loyalty or reward points accumulated through card usage.	Number
Exited	The target variable. Indicates if the client has departed the bank. (1 = Yes, 0 = No).	Number

3.1 Data Cleaning and Pre-processing

Data cleaning and preprocessing stages were carried out to ensure optimisation of the model. The steps include:

1. Handling Duplicates: Eliminating duplicates is an important step, as duplicated values can cause bias in deep learning/machine learning models. The first step was to check the dataset for duplicate rows.
2. Feature Selection: This procedure was done to remove irrelevant or redundant variables that do not have predictive properties to reduce noise and prevent unintended model memorisation.

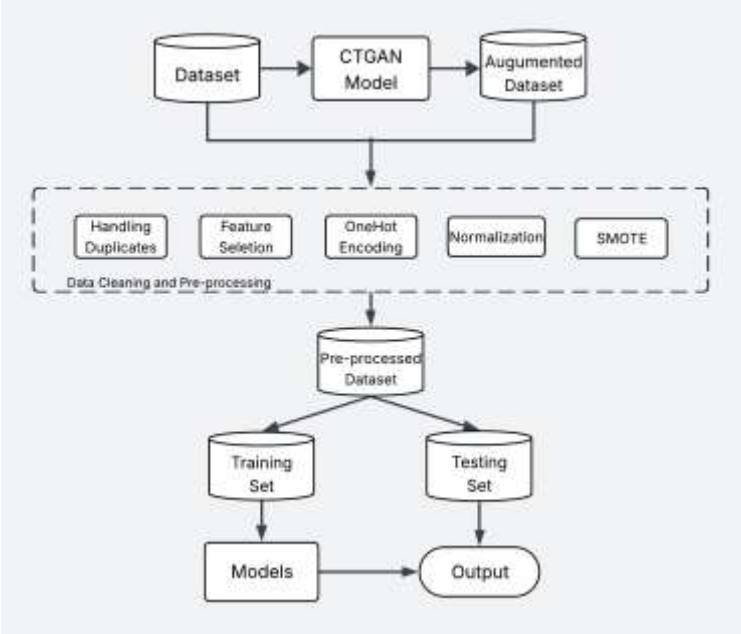

Figure 1: System architecture

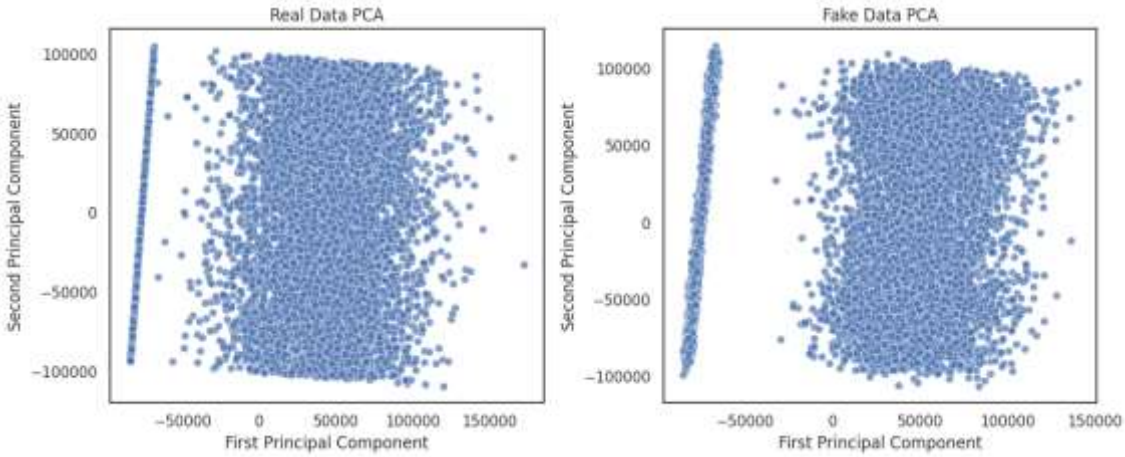

Figure 2: Principal component analysis plot of the real and synthetic data

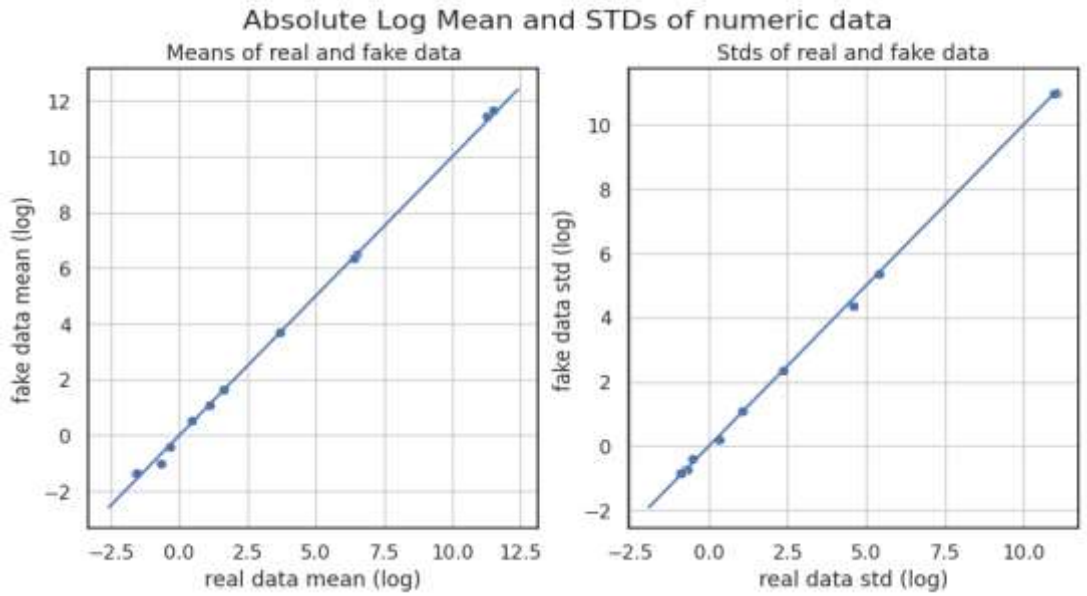

Figure 3: Absolute log mean and standard deviation plot comparison of the real and synthetic data

The features "RowNumber, Surname and CustomerID" were taken out as they are merely unique identifiers that do not contribute to churn prediction and will not affect the model's decision. Removing these columns reduces the dataset's dimensionality, thereby speeding up model training time.

3. Column Renaming: Renaming column headings is a crucial stage in the data preparation phase when working with data that has unclear and inconsistent column naming. The column names were standardised to a consistent code friendly convention, and the columns with spaces, inconsistent capitalisation and special characters were adjusted to avoid coding errors
4. Encoding Categorical Variables: Machine learning algorithms cannot process categorical text variables directly; they must be converted into a numerical format. The datatypes of the features were checked, noting the unique variables of each feature. In this study, the gender column was converted with a label encoder because it has two categories, and the mapping is straightforward, while the geography and card_type columns, which have more than two categories, were converted using OneHot encoding. OneHot encoding was applied with "drop='first'" to avoid perfect multicollinearity.
5. Normalization: The numerical features were standardised using MinMaxScaler with the objective of decreasing dimensionality and enhancing model functionality.
6. Advanced Feature Selection: To reduce redundancy and the risk of overfitting, a pairwise correlation between the features was computed and visualised with a heatmap. where two features showed high correlation (threshold ≥ 0.85), one of the pair was removed. Additionally, the Variance Inflation Factor (VIF) was computed to identify multicollinearity, and features with very high VIF values were considered for removal to facilitate faster model convergence.
7. Handling Class Imbalance (Oversampling): The dataset is highly skewed, and training the model with this can severely bias the models in favour of the majority class, hence causing underfitting and yielding a bad sensitivity and F1 Score. To address this, SMOTE (Synthetic Minority Over-sampling Technique) was applied to the dataset. SMOTE creates new examples of minority classes by combining current minority instances. After SMOTE, the dataset contained equal proportions of churners and non-churners.
8. Data Splitting into Training and Test Sets: To examine the model's generalisation performance, the dataset was split into training and test sets. A stratified split with a test_size of 0.3 was used to divide the data into a 70:30 ratio for training and testing, respectively. This proportion was chosen because it offers a decent balance between evaluating and training the model. Training using 70% guarantees that the model has sufficient samples to learn the complex patterns within the data while also having enough for testing. Using a smaller test set, e.g. 10% could make the evaluation less reliable, while a large test set leaves small samples for training.

This study employed both deep learning and ensemble machine learning approaches to build a robust churn detection system. The models used include a Feed Forward Neural Network (FNN) and Voting Classifier Ensembles (hard and soft voting). The ensemble models combined three base classifiers: Random Forest, Extreme Gradient Boosting (XGBoost) and Logistic Regression.

3.2 Feedforward Neural Network

A feedforward neural network is one of the foundational architectures of artificial neural networks, and it is widely used for classification tasks. In this, the node connections do not form cycles, the information travels in a forward path from the input layer through the hidden layers and finally to the output layer. In this study, the neural network was implemented using Python's TensorFlow/Keras framework. The train split of the processed dataset was received by the input layer, followed by the hidden layer neurons, utilising Rectified Linear Unit (ReLU) activation functions, which enabled the model to capture the complex interactions among the features. To prevent overfitting, dropout regularisation and batch normalisation were applied. Sigmoid activation was employed for binary classification in the output layer, producing a probability between 0 and 1, and a decision boundary threshold was applied to classify the result. Equation (1) depicts the decision boundary equation [16]. After that, the model was trained using the binary cross-entropy loss function and the Adam optimiser for 10 epochs, ensuring rapid convergence in learning the nonlinear patterns.

$$decision = \begin{cases} 1, if\ \hat{y} \geq 0.5\ (Churn) \\ 0, if\ \hat{y} < 0.5\ (Normal) \end{cases} \quad (1)$$

3.3 Voting Classifier Ensemble

This is an ensemble learning technique that generates an improved result from merging the results of multiple base models. Voting classifiers are mostly used when the base models have varying strengths and weaknesses because they group the diverse characteristics and make a super model that leverages all their advantages. There are two types

1. Hard Voting: The hard voting classifier makes its final predictions based on the majority votes. Each base model in the classifier forecasts a class category, and the class that receives the highest number of votes from the three models is assigned as the final output prediction. Hard voting is mostly used in scenarios where individual classifiers have high accuracy, as it relies on the collective decision of multiple strong models. It is modelled according to Equation (2) [17].

$$\hat{y} = \text{mode}\{h_1(x), h_2(x), \ldots, h_n(x)\} \quad (2)$$

Where:

$h_i(x)$ is the class prediction of the i^{th} base model

$\hat{y}$ is the final predicted class (either 0 or 1)

2. Soft Voting: The soft voting classifier makes its predictions using weighted probability averaging. Instead of selecting the most common class label, it sums the predicted probabilities from each classifier for each class label and chooses the class that has the highest average likelihood. When the base models output well calibrated probability scores, soft voting tends to perform better as it considers the confidence level of each model's predictions. Equation (3) defines the mathematical model of the soft voting classifier [17].

$$\hat{y} = \arg\max\left(\frac{1}{n}\sum_{i=1}^{n} P_{i,k}(x)\right) \quad (3)$$

Where $P_{i,k}(x)$ is the predicted probability of class $k \in \{0, 1\}$ by the i^{th} classifier.

The final prediction will now be calculated by

$$\text{decision} = \begin{cases} 1, \text{if } \hat{y} \geq 0.5 \text{ (Churn)} \\ 0, \text{if } \hat{y} < 0.5 \text{ (Normal)} \end{cases} \quad (4)$$

The ensemble voting classifiers in this study combined three machine learning models:

1. Random Forest: A reliable and popular ensemble learning method is random forest, which consists of a couple of decision trees. It is prevalent in classification problems due to its high accuracy, making it suitable for churn detection. It works by assembling a couple of decision trees. Samples of the dataset are trained individually with these trees, and the class prediction with the highest outcome from these trees is selected as the final output. Each decision tree is trained using a different bootstrap sample (selecting a specified number of objects with replacements) from the training dataset and a random selection of each node's split features. This randomness ensures little to no correlation among the trees, reducing overfitting and improving generalisation of the model [18].
2. Logistic Regression: This is an approach that is widely used for binary classification problems, making it ideal for churn detection where the outcome is either true or false. In this study, logistic regression was used as one of the base models in the voting classifier. The model estimates the probability $P(y = 1|X)$ that a given customer will churn, and $P(y = 0|X)$ is the probability that the customer will not churn. Logistic regression works by learning through a set of vectors which contains real numbers of weights and biases. Using the logistic regression model, the probability of churn is expressed by Equations (5-9) [19]. The term z, which is a function of the weight and bias, is given as:
    a. $z = (\sum_{i=1}^{n} w_i x_i) + b$ (5)
    b. $z = \mathbf{w} \bullet \mathbf{x} + b$ (6)
    c. The sigmoid/logistic function σ(z) is used to calculate the probability since it's a binary classification task. The sigmoid function is represented by:
    $$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (7)$$
    d. If the weighted feature total is subjected to the logistic function, a value from 0 and 1 is obtained. The final equation is given as:
    $$P(y = 1 \mid X) = \frac{1}{1 + e^{-(w \bullet x + b)}} \quad (8)$$
    e. To decide which class to assign the probability values gotten from the logistic function, a decision boundary is specified such that the classes are specified if the value is greater or less than the boundary. Equation (9) shows the decision boundary:
    $$\text{decision} = \begin{cases} 1, \text{if } P(y = 1 \mid X) \geq 0.5 \text{ (Churn)} \\ 0, \text{if } P(y = 1 \mid X) < 0.5 \text{ (Normal)} \end{cases} \quad (9)$$
3. Extreme Gradient Boosting (XGBoost): This is an enhanced and scalable version of gradient boosted decision trees, which are designed for speed and better performance. It is highly effective for structured or tabular datasets and is used by data scientists to achieve exceptional results [6]. Unlike Random Forest, which builds decision trees simultaneously, XGBoost constructs the trees one after the other, attempting to correct the mistakes of the

earlier trees. To train the model, XGBoost uses a second-order Taylor approximation of the loss and finds the optimal structure of the next tree to minimise the regularised objective function [20].

3.6 Evaluation Metrics

The model's effectiveness was assessed using accuracy and F1 score as performance metrics, all of which were derived from the values of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) in the confusion matrix. Additionally, the Area Under the Receiver Operating Characteristic (ROC) curve was employed to further evaluate the models' generalisation performance.

Accuracy measures the ratio of correctly predicted instances to the total instances. It is calculated as shown in Equation (10) [11]:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (10)$$

Precision is a metric that measures the proportion of correctly predicted positive predictions to the total number of positive predictions made. It is defined as shown in Equation (11) [11]:

$$Precision = \frac{TP}{(TP + FP)} \quad (11)$$

Recall measures how many actual positives are correctly classified by the model. It is defined as shown in Equation (12) [11]:

$$Recall = \frac{TP}{(TN + FP)} \quad (12)$$

F1 Score is calculated as the harmonic mean of precision and recall, it reflects how well the model balances these two aspects of performance. A strong F1 score indicates effective and consistent predictions. It is represented as shown in Equation (13) [11]:

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (13)$$

The Receiver Operating Characteristics (ROC) curve is a visual depiction of the true positive and the false positive rates. It is a classification metric often used with binary classification models that helps to evaluate the model's ability to differentiate between the classes. The closer the ROC curve is to the top left corner, the better the model performs.

The Area Under the ROC Curve (AUC) serves as a summary measure of the model's ability to differentiate between positive and negative instances correctly. A higher AUC value means the model has better discrimination ability, making it useful for imbalanced classification tasks

4. RESULTS AND DISCUSSION

This section presents the outcome from evaluating the three churn prediction models on the test split of the dataset. Performances were evaluated using accuracy, F1 score, and the Area Under the ROC curve, guaranteeing a thorough evaluation of the models. After augmentation, the dataset contained 20,000 unique records with 15,372 churned customers and 4,628 non-churned customers. After oversampling was performed using SMOTE to mitigate bias towards the majority class, the dataset comprised 30,744 total records, evenly split between the two classes (i.e., 15,372 records each). Table 3 summarises the models' performance, and Figure 4 illustrates the ROC curves of the models after the first conclusive training. At this stage, all models achieved extremely high performance, indicating the presence of redundant features which may have contributed to overfitting, causing the models to memorise patterns rather than generalise. The perfect scores suggested that the models could detect churn perfectly, but such ideal performance is often unrealistic in real world scenarios.

Advanced feature selection was done by removing features with high correlation to achieve a more realistic and generalised model. The correlation heatmap of the dataset's features is displayed in Figure 5. The 'complain' feature was seen to have a high correlation with the target label feature 'churn', and upon removal and re-training the models, the results obtained are summarised in Table 4, and Figure 6 shows the ROC curves.

Table 3: Results after first training

Model	Accuracy (%)	F1 Score (%)	AUC (%)
FNN	99.87	99.90	100
Soft Voting Classifier	99.92	99.98	100
Hard Voting Classifier	99.90	99.98	-

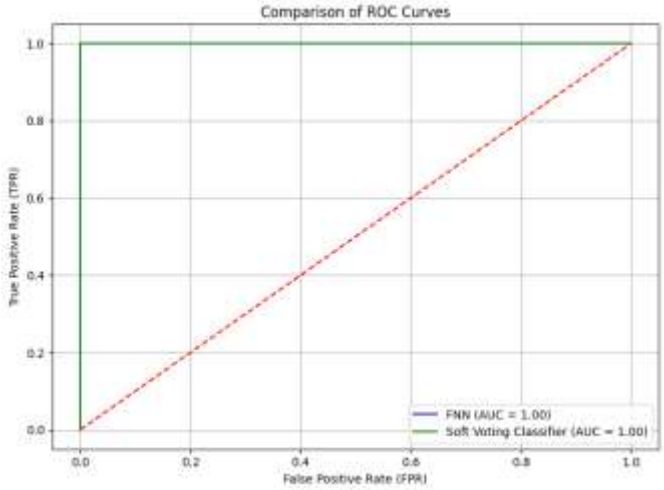

Figure 4: ROC after first training

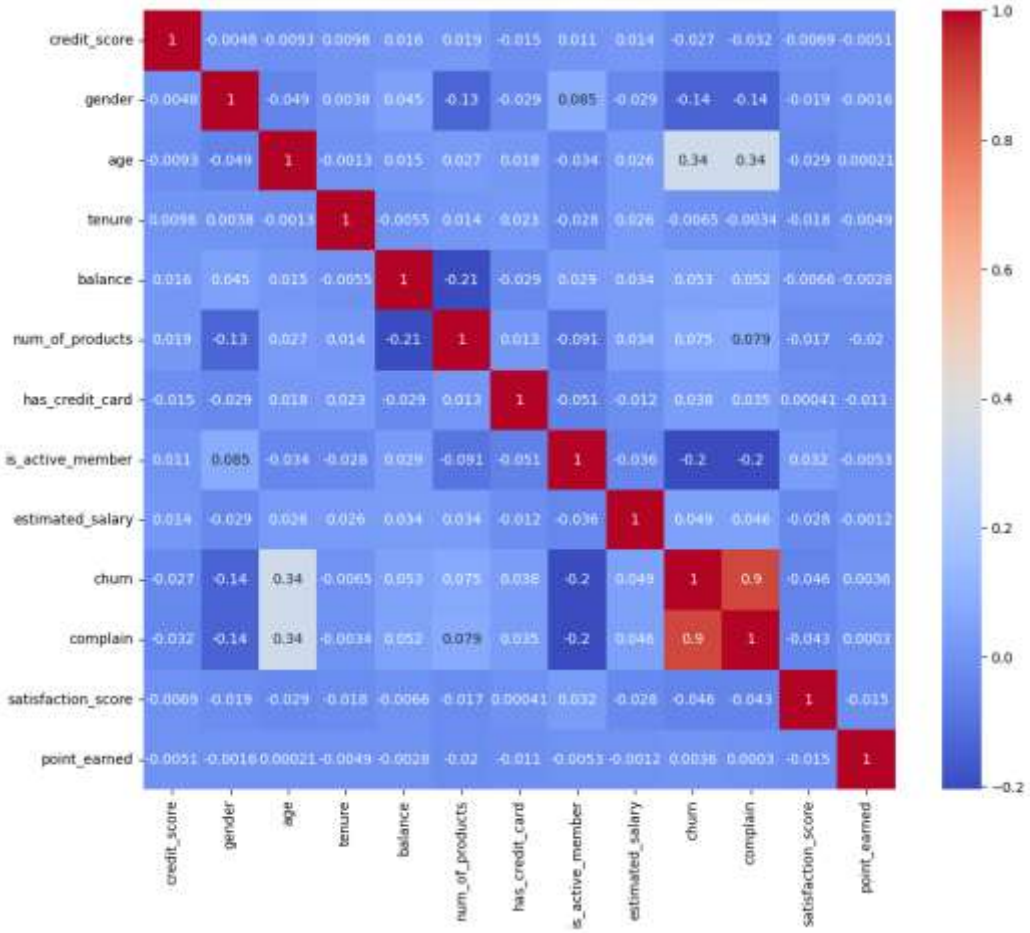

Figure 5: Correlation heatmap

Table 4: Results after feature selection and oversampling

Model	Accuracy (%)	F1 Score (%)	AUC (%)
FNN	88.23	87.83	94.73
Soft Voting Classifier	89.46	88.92	95.40
Hard Voting Classifier	89.04	88.48	-

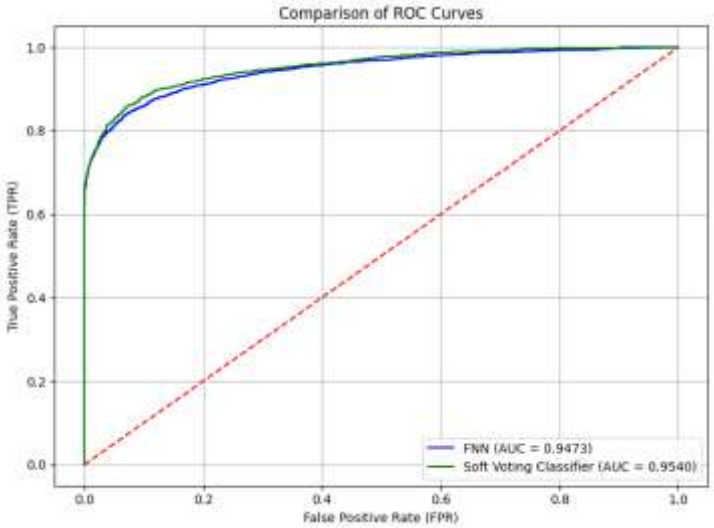

Figure 6: ROC after feature selection

The hard voting classifier ROC was not evaluated because hard voting classifiers generally output the final class prediction, and this will produce an ROC curve of two straight lines that represent the sensitivity and specificity of the model [21].

The proposed outcomes show that ensemble approaches, particularly the soft voting classifier, outperform the feedforward neural network. While the neural network model provided competitive results, the collaborative strength of the ensemble produced higher accuracy and a better balance of performance with a higher F1 score and AUC, making it more suitable for practical churn detection applications in the banking industry. Table 5 evaluates the outlined models' performance against that of the best models from previous studies. Random forest performed best, with accuracy scores above 88% in all studies. The proposed models did not surpass the very best outcomes, however, they remain highly competitive, achieving performance levels that are on par with or exceed the majority of earlier works, confirming the practical utility of the approach.

To validate the proposed results, an additional test was conducted using five random variables from the test set. The models performed exceptionally well, with the voting classifiers predicting five out of five correctly, and the neural network predicting four out of five. The results are summarised in Table 6.

Table 5: Evaluation of proposed models in relation to earlier research

Model	**Accuracy (%)**	**F1 Score (%)**	**AUC (%)**
Random Forest [8]	95.74	Not Reported	Not Reported
Random Forest [9]	88.70	91	Not Reported
XGBoost [10]	83.90	81.30	84.70
AdaBoost [11]	81.71	80.28	84.
Random Forest [12]	88.63	88	95.90
Proposed Feedforward Neural Network	88.23	87.83	94.73
Proposed Soft Voting Classifier	89.46	88.92	95.40
Proposed Hard Voting Classifier	89.04	88.48	-

Table 6: Results after testing with five random variables

Y Test	**FNN Prediction**	**Soft Voting Classifier Prediction**	**Hard Voting Classifier Prediction**
1	1	1	1
0	0	0	0
1	0	1	1
1	1	1	1
1	1	1	1

5. CONCLUSION

In today's competitive financial landscape, understanding customers and predicting their behaviour has become an essential task for banks that aim to maintain long-term relationships. As customer expectations keep evolving, so does the complexity of the strategies needed for retention, making predictive models made from machine learning a top tier choice.

This research examined how well neural networks work in comparison to ensemble machine learning approaches in predicting the likelihood of a customer switching banks. After applying Conditional Tabular Generative Adversarial Network (CTGAN) to enhance the dataset and addressing class imbalance using SMOTE, the three models demonstrated strong predictive capabilities. However, the soft voting ensemble classifier achieved the best overall performance, with an accuracy of 89.46%, an F1 score of 88.92%, and an AUC of 95.40%, closely rivalling or surpassing most models reported in previous studies.

In view of these, several suggestions are advised for future studies. Firstly, the deep learning component of the study could be improved by using more sophisticated neural network architectures, such as models based on transformers or Long Short-Term Memory (LSTM), which can better capture sequential customer behavioural patterns over time. Secondly, future work should consider training and evaluation of the models on primary data collected from a local bank to assess the real world applicability of the study, as expanding the dataset size with more diverse features could lead to improved performance. Additionally, future works may consider experimenting with more complex ensemble strategies.

REFERENCES

[1] Lemmens, A. & Gupta, S. (2020). Managing Churn to Maximize Profits. *Marketing Science*, 39(5), 956-973. https://doi.org/10.1287/mksc.2020.1229.

[2] Peng, K., Peng, Y., & Li, W. (2023). Research on customer churn prediction and model interpretability analysis. *Plos one*, 18(12). https://doi.org/10.1371/journal.pone.0289724

[3] Ebrah, K., & Elnasir, S. (2019). Churn prediction using machine learning and recommendations plans for telecoms. *Journal of Computer and Communications*, 7(11), 33-53. https://doi.org/10.4236/jcc.2019.711003.

[4] Tekouabou, S. C. K., Cherif, W., & Silkan, H. (2019). A data modeling approach for classification problems: application to bank telemarketing prediction. *in Proceedings of the 2nd international conference on networking, information systems & security*. https://dl.acm.org/doi/abs/10.1145/3320326.3320389.

[5] Imani, M., Joudaki, M., Beikmohamadi, A., & Arabnia, H. R. (2025). Customer Churn Prediction: A Review of Recent Advances, Trends, and Challenges in Conventional Machine Learning and Deep Learning. https://doi.org/10.20944/preprints202503.1969.v1

[6] Geiler, L., Affeldt, S., & Nadif, M. (2022). A survey on machine learning methods for churn prediction. *International Journal of Data Science and Analytics*, 14(3), 217-242. https://doi.org/10.1007/s41060-022-00312-5.

[7] Obiora, N. C., & Uchenna, N. D. (2022). PREDICTING CUSTOMER CHURN IN THE TELECOMMUNICATION INDUSTRY USING MACHINE LEARNING ALGORITHMS: Performance comparison with logistic regression, random forest, and gradient boosting techniques.

[8] Rahman, M., & Kumar, V. (2020). Machine learning based customer churn prediction in banking, *in 4th international conference on electronics, communication and aerospace technology (ICECA)*. https://doi.org/10.1109/ICECA49313.2020.9297529.

[9] Muneer, A., Ali, R. F., Alghamdi, A., Taib, S. M., Almaghthawi, A., & Ghaleb, E. A. A. (2022). Predicting customers churning in banking industry: A machine learning approach. *Indonesian Journal of Electrical Engineering and Computer Science*, 26(1), 539-549. http://doi.org/10.11591/ijeecs.v26.i1.pp539-549.

[10] Singh, P. P., Anik, F. I., Senapati, R., Sinha, A., & Sakib, N. (2024). Investigating customer churn in banking: A machine learning approach and visualization app for data science and management. *Data Science and Management*, 7(1), 7-16. https://doi.org/10.1016/j.dsm.2023.09.002.

[11] Lalwani, P., Mishra, M. K., Chadha, J. S., & Sethi, P. (2021). Customer churn prediction system: a machine learning approach. *Computing*, 104(2), 271-294. https://doi.org/10.1007/s00607-021-00908-y.

[12] Ullah, I., Raza, B., Malik, A. K., Imran, M., Islam, S. U., & Kim, S. W. (2019). A churn prediction model using random forest: analysis of machine learning techniques for churn prediction and factor identification in telecom sector. *IEEE access*, 7, 60134-60149. https://doi.org/10.1109/ACCESS.2019.2914999.

[13] Çelik, Ö., & Osmanoğlu, U. Ö. (2019). Comparing to techniques used in customer churn analysis. *Journal of Multidisciplinary Developments*, 4(1), 30-38.

[14] Ahn, J., Hwang, J., Kim, D., Choi, H., & Kang, S. (2020). A survey on churn analysis in various business domains, *IEEE access*, 8, 220816-220839. https://doi.org/10.1109/ACCESS.2020.3042657.

[15] Kollipara, R. (n.d.). Bank Customer Data for Customer Churn on Kaggle. [Online]. Available: https://www.kaggle.com/datasets/radheshyamkollipara/bank-customer-churn/data.

[16] Muruganandam, S., Joshi, R., Suresh, P., Balakrishna, N., Kishore, K. H., & Manikanthan, S. V. (2023). A deep learning based feed forward artificial neural network to predict the K-barriers for intrusion detection using a wireless sensor network. *Measurement: Sensors*, 25. https://doi.org/10.1016/j.measen.2022.100613.

[17] Aruna, S., Divya, M., & Sahu, P. K. (2025). Feature-Based Child Mortality Prediction Using Ensemble and Traditional Machine Learning Models. *Journal of Applied Science and Technology Trends*, 6(2), 169-182.

https://doi.org/10.38094/jastt62264.
[18] Salman, H. A., Kalakech, A., & Steiti, A. (2024). Random Forest Algorithm Overview. *Babylonian Journal of Machine Learning*, 69-79. https://doi.org/10.58496/BJML/2024/007.
[19] Jurafsky, D., & Martin, J. H. (2025). Speech and Language Processing, 3rd ed draft. Available: https://web.stanford.edu/~jurafsky/slp3.
[20] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System, *in Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785-794. https://doi.org/10.1145/2939672.2939785
[21] Reiniger, B. (2020). Is there any way to plot ROC curve for Ensemble hard voting classifier? On Stack Exchange. [Online]. Available: https://datascience.stackexchange.com/questions/77327/is-there-any-way-to-plot-roc-curve-for-ensemble-hard-voting-classifier.