machine learning & knowledge extraction

MDPI

Article

Enhancing Performance of Credit Card Model by Utilizing LSTM Networks and XGBoost Algorithms

Kianeh Kandi [1] and Antonio García-Dopico [1,2,*]

[1] Departamento de Arquitectura y Tecnología de Sistemas Informáticos (DATSI) Computer Science, Escuelta Técnica Superior de Ingenieros Informáticos, Universidad Politécnica de Madrid, 28660 Madrid, Spain; kianeh.kandi@alumnos.upm.es

[2] Centro de Investigación en Simulación Computacional, Universidad Politécnica de Madrid, 28660 Madrid, Spain

* Correspondence: antonio.garcia.dopico@upm.es

Abstract: This research paper presents novel approaches for detecting credit card risk through the utilization of Long Short-Term Memory (LSTM) networks and XGBoost algorithms. Facing the challenge of securing credit card transactions, this study explores the potential of LSTM networks for their ability to understand sequential dependencies in transaction data. This research sheds light on which model is more effective in addressing the challenges posed by imbalanced datasets in credit risk assessment. The methodology utilized for imbalanced datasets includes the use of the Synthetic Minority Oversampling Technique (SMOTE) to address any imbalance in class distribution. This paper conducts an extensive literature review, comparing various machine learning methods, and proposes an innovative framework that compares LSTM with XGBoost to improve fraud detection accuracy. LSTM, a recurrent neural network renowned for its ability to capture temporal dependencies within sequences of transactions, is compared with XGBoost, a formidable ensemble learning algorithm that enhances feature-based classification. By meticulously carrying out preprocessing tasks, constructing competent training models, and implementing ensemble techniques, our proposed framework demonstrates unwavering performance in accurately identifying fraudulent transactions. The comparison of LSTM and XGBoost shows that LSTM is more effective for our imbalanced dataset. Compared with XGBOOST's 97% accuracy, LSTM's accuracy is 99%. The final result emphasizes how crucial it is to select the optimal algorithm based on particular criteria within financial concerns, which will ultimately result in more reliable and knowledgeable credit score decisions.

Keywords: recurrent neural network (RNN); long short-term memory (LSTM) network; extreme gradient boosting (XGBoost); synthetic minority oversampling technique (SMOTE); imbalanced dataset

Academic Editor: Peter Kieseberg

Received: 14 November 2024
Revised: 28 January 2025
Accepted: 2 February 2025
Published: 21 February 2025

Citation: Kandi, K.; García-Dopico, A. Enhancing Performance of Credit Card Model by Utilizing LSTM Networks and XGBoost Algorithms. *Mach. Learn. Knowl. Extr.* **2025**, *7*, 20. https://doi.org/10.3390/make7010020

1. Introduction

The advent of digital transactions has made credit card fraud a significant concern for financial institutions worldwide. With the increasing sophistication of fraudulent activities, traditional detection methods are proving to be inadequate.

Traditional models like Probit and Logit struggle with complex datasets due to their reliance on linear assumptions, limited ability to capture feature interactions, and poor handling of high-dimensional, sequential, or imbalanced data. They also lack built-in mechanisms like regularization and scalability for large datasets. In contrast, modern techniques like XGBoost and LSTM excel in addressing these challenges, offering superior performance by effectively modeling non-linear relationships and sequential data and handling

imbalanced datasets with greater robustness. This makes modern models better suited for tasks such as credit card default prediction.

This paper explores the potential of using Long Short-Term Memory (LSTM) networks and Extreme Gradient Boosting (XGBoost) algorithms to enhance the performance of credit card fraud detection models. LSTM networks, a subset of recurrent neural networks known for their prowess in capturing intricate temporal relationships within sequential data, stand as formidable tools for the analysis of time-series credit card transactions. On the other hand, XGBoost, a powerful gradient boosting algorithm, excels at handling structured data and offers robust performance in classification tasks. The deployment of sophisticated deep learning models plays a crucial role in closing the gap previously filled with outdated financial tools [1].

One of the most renowned and widely utilized contemporary payment methods worldwide is the credit card, which offers exceptional services for both purchasing and selling transactions. Nevertheless, it is plagued by fraudulent activities that lead to substantial monetary losses for financial institutions, enterprises, and individuals annually amounting to billions of dollars. Machine learning (ML) techniques represent a commonly used approach by scholars in the field. Various machine learning methods, such as Logistic Regression (LR), Linear Discriminant Analysis (LDA), Naïve Bayes (NB), and the boosting technique XGBoost, were employed to develop models capable of detecting fraudulent activities. Evaluation of model performance involves the utilization of metrics such as accuracy, precision, recall, F1 score, and AUC confusion matrix. The XGBoost model demonstrated superior outcomes when compared with alternative models [2].

The exponential growth in the demand for credit cards can be attributed to the many benefits and enhanced security measures they offer to customers. Consequently, credit card companies exercise caution and thorough consideration before granting approval to potential customers. Only those individuals who satisfy the predetermined criteria established by these companies are eligible for a credit card. With more than 50% US citizens reporting fraudulent transactions, credit card fraud is a serious problem for economic institutions. Using conventional approaches, it is far more difficult and time-consuming to locate such fraud. Using AI techniques such as Random Forest, Logistic Regression, SVM, Naïve Bayes, XGBoost, and KNN, technology can assist in the development of automatic fraud detection methods. This look addresses machine learning approaches including XGBoost and Random Forest and evaluates the work of several researchers on credit card fraud detection on highly skewed datasets. The consequences reveal that integration methods like Random Forest and XGBoost perform better when figuring out credit card transactions as fraudulent [3].

To minimize losses, an effective fraud detection machine method should be designed and implemented. However, fraud sequences or behavioral changes that could motivate false alarms are not taken into account by the machine-mastering techniques used to mechanically discover card fraud. Static machine learning models fail to consider shifts and patterns in consumer spending patterns, such as those that occur during specific seasons and geographical areas. In these circumstances, financial institutions must put in place an accurate fraud detection system that continuously evolves and adapts to new fraud behaviors in order to stop fraud before it starts, protect the interests of their customers, and reduce the harm that fraud causes. The complex architecture of Long Short-Term Memory (LSTM) allows neural networks to intertwine connections between nodes over a series of time steps. With the use of this novel framework, the model is able to store and utilize information from previous inputs, allowing it to identify temporal correlations between different events that are contained within a specific input sequence. Essentially, Long Short-Term Memory (LSTM) is an advanced technique to identify and decipher

succession patterns within sequential data points, where the occurrence of one event may depend intricately on the interactions of multiple earlier events that are dispersed over time [4].

Throughout credit card operations, enormous amounts of data are produced in a variety of ways, including primary customer information, invoice, installment, and reimbursement statistics, transaction flows, and past-due information. Thanks to recent advancements in the maturity and applicability of technologies like artificial intelligence and large statistics evaluation, massive transaction facts can now also be mined and analyzed. In our paper, two popular models have been selected: XGBoost (utilized in monetary categorization models) and Long-Short Term Memory (LSTM) (utilized in time series statistics). While the LSTM set of rules can reach better accuracy without characteristic extraction, the accuracy of the XGBoost model is dependent on the extent of characteristic extraction talent. In default prediction, the generated XGBoost and LSTM models validated an appropriate classification performance. The findings can serve as a guide for the use of deep learning algorithms inside the finance enterprise. Ref. [5] proposed the issue of imbalanced data using two strategies and demonstrated effective prediction performance on a dataset with limited interface residues. The XGBoost-based method achieved a prediction accuracy of 0.807. The proposed feature extraction method, based on the evolutionary conservatism of proteins, significantly contributed to the prediction accuracy and overall performance of the model. By considering the influence of overlapping regions of positive and negative samples, the computational model showed enhanced prediction capability, addressing the challenge of imbalanced data distribution in protein–protein interaction site prediction. Experimental results confirm the effectiveness of the XGBoost algorithm in predicting protein–protein interaction sites, highlighting its potential for applications in drug development and understanding cell biological activities.

The structure of this paper is organized as follows. Section 2 provides a comprehensive review of the relevant literature, focusing on applying advanced machine learning techniques, such as Long Short-Term Memory (LSTM) and Extreme Gradient Boosting (XGBoost), in credit risk prediction. Section 3 introduces the proposed methodology, including detailed preprocessing steps, the application of the Synthetic Minority Oversampling Technique (SMOTE) for addressing class imbalance, and the architectural frameworks of the LSTM and XGBoost models. Section 4 describes the experimental setup, outlining data preparation, hyperparameter configurations, and the evaluation metrics used to assess model performance. Section 5 presents the results and discussion, offering a comparative analysis of LSTM and XGBoost. This section emphasizes the relevance of precision, recall, and F1 score metrics for evaluating model effectiveness. Finally, it concludes the study, summarizing key findings and their implications and outlining future research directions, including the potential development of hybrid and ensemble approaches to improve the accuracy and robustness of credit risk predictions.

2. Literature Review

Traditional models like Probit and Logit are favored for their simplicity, interpretability, and theoretical foundations. However, their reliance on linearity assumptions may limit performance on complex datasets. Robit offers robustness against outliers, bridging the gap between traditional methods and more flexible machine learning techniques. In contrast, MLPs and RBF networks shine in scenarios involving high-dimensional data or intricate relationships. MLPs, with their deep architectures, can approximate any continuous function, while RBF networks focus on localized data representation. These models, however, require careful hyperparameter tuning and significant computational resources.

The choice of a binary classification model hinges on dataset characteristics, such as size, complexity, and the presence of outliers. Traditional methods like Probit, Logit, and Robit offer interpretability and computational efficiency, making them suitable for simpler problems. Meanwhile, MLPs and RBF networks provide advanced modeling capabilities for complex, non-linear data. While traditional models are valuable for their simplicity and interpretability, their rigid assumptions and limited scalability make them less effective in complex, real-world scenarios. ML models, with their flexibility, automatic feature extraction, and advanced learning capabilities, are better equipped to handle the diverse and complex datasets often encountered in modern binary classification problems.

Gao et al. presented a novel approach to improving the accuracy of credit risk prediction. Their study mined and analyzed transaction flow data, which best reflected customer behavior. They compared XGBoost, a widely used algorithm in financial classification tasks, with Long Short-Term Memory (LSTM), a model suitable for time-series data. The findings indicated that the accuracy of XGBoost depends heavily on the expertise involved in feature extraction, whereas LSTM achieves high accuracy without requiring feature extraction. These results highlight the superiority of LSTM in handling raw sequential data. Validation of their hypothesis demonstrated that transaction flow features held the highest importance. Moreover, the accuracy of default prediction in XGBoost primarily relied on feature extraction, while LSTM required only minimal preprocessing, such as data complementation and splicing, without the need for manual feature engineering [6].

Another study investigated the use of XGBoost to predict credit card approvals and compared its performance with the Random Forest (RF) algorithm. The analysis showed that XGBoost achieved an accuracy of 87.97%, significantly outperforming RF, which achieved an accuracy of only 82.86%. These findings suggest that XGBoost is more effective than RF in accurately predicting credit card approvals, while simultaneously reducing the loss percentage from 17.14% to 12.03% [7].

A further study focused on aiding bank management in evaluating credit card clients by forecasting consumer behavior. Using real-world credit card data, researchers trained an LSTM model and assessed its performance using metrics such as accuracy and area under the curve (AUC). The results conclusively demonstrate that LSTM scores effectively predicted both isolated and successive missed payments, outperforming conventional machine learning algorithms, including Support Vector Machines (SVMs), Random Forests, multilayer perceptrons, and Logistic Regression. Overall, the LSTM-based credit scoring method significantly improved the accuracy of credit risk predictions [8].

In another insightful study, researchers tackled the challenging task of categorizing credit card security issues by applying various machine learning methodologies. They compared prominent classifiers, including SVM, Random Forest, Bagged Tree, K-Nearest Neighbor (KNN), Naïve Bayes, and Extreme Gradient Boosting (EGB). The findings revealed that KNN achieved the highest classification accuracy, exceeding 97.50%, while the LSTM model also demonstrated strong performance, achieving an accuracy of over 96% [9].

A novel approach for credit risk prediction combining deep learning and SMOTE techniques was proposed to address imbalanced datasets. The researchers developed and analyzed stacked LSTM and stacked BiLSTM (Bidirectional Long Short-Term Memory) models, integrated with SMOTE oversampling. This approach proved especially effective for real-world credit scoring datasets, which often lack time dependence or correlation. Additionally, SMOTE applied to a three-dimensional array further refined the dataset, enhancing the accuracy of credit score predictions [10].

Mekruksavanich et al. introduced a novel deep learning model called the Long Short-Term Memory Neural Network with eXtreme Gradient Boosting (LSTM-XGB) for Human Activity Recognition (HAR). HAR tasks involve activities related to daily life and falls, which are critical in applications such as health monitoring, sports analytics, and smart home systems. The LSTM-XGB model combines LSTM layers to understand input characteristics and automatically learn features, followed by XGBoost in the final layer for class label prediction. After the LSTM extracts temporal features from input data, these features are fed into an XGBoost classifier, which utilizes them to categorize the data into various activity types. The model's efficiency can be enhanced by simplifying its parameters. Experiments on smartphone sensor data demonstrated that the LSTM-XGB model achieved superior identification capabilities, with the highest accuracy of 92.59%, outperforming existing best-practice models on the same dataset [11].

Another study aimed to optimize machine learning model performance by leveraging boosting algorithms. The model, built using XGBoost, was compared with Support Vector Machines (SVMs) and multilayer perceptron (MLP) neural networks. The experimental results show that the boosting model slightly surpassed the performance of SVM and MLP, indicating its effectiveness in optimizing accuracy and computational efficiency [12].

Raval et al. proposed a novel framework, "RaKShA", combining explainable artificial intelligence (XAI) with LSTM models to enhance the detection of credit card fraud. Traditional machine learning algorithms have been ineffective in addressing the growing complexities of fraud detection. Although LSTM models show promise, their black-box nature limits interpretability. RaKShA employs XAI techniques to extract relevant features from the credit card fraud dataset, improving LSTM model performance. The extracted features undergo further analysis, and the output classifications are stored in a smart contract for result integrity. Blockchain technology ensures secure and chronological ledger entries. The XAI-LSTM model demonstrated an impressive accuracy rate of 99.8%, compared with 85% without XAI [13].

Raj et al. highlighted the exceptional performance of the XGBoost algorithm in fraud detection, reporting an accuracy rate of 97% and a precision level of 94%. The model achieved a high area under the curve (AUC) score of 0.97, reflecting its robust capability to distinguish between fraudulent and legitimate transactions. Furthermore, the study explored combining machine learning techniques with blockchain technology to strengthen security and foster trust in online financial systems [14].

Priscilla and Prabha introduced an optimized XGBoost (OXGBoost) model that addresses class imbalances without resampling techniques. The OXGBoost approach integrates RandomizedSearchCV for hyperparameter optimization and data sampling methods to enhance model efficiency. Experiments on real-world credit card datasets showed that the OXGBoost approach outperformed traditional methods, achieving superior accuracy without sacrificing computational efficiency [15]. Additionally, they proposed a two-phase feature selection method combining filter and wrapper techniques. In the first phase, features are ranked based on mutual information (MI). The second phase employs recursive feature elimination (RFE) with five-fold cross-validation to eliminate redundant features. The selected features are applied to boosting algorithms like XGBoost, Gradient Boosting Machine (GBM), Classic Gradient Boosting (CatBoost), and Light Gradient Boosting Machine (LGBM). The results from a credit card fraud dataset demonstrate promising outcomes, with GMean values of 84.8% for XGBoost and 83.7% for LGBM, alongside an AUC improvement from 79.8% [16].

Alonso and Carbó evaluated the predictive performance of various machine learning (ML) models using a distinct and anonymized banking dataset in Spain. These models were compared with Logistic Regression (Logit), a standard benchmark model. While ML models outperformed Logit in both classification and calibration, the study noted that increased algorithm complexity does not always lead to superior predictions. The authors emphasized the significant economic benefits of ML models, advocating for further research to address the risks associated with these methods [17].

Since X (formally Twitter) is a widely used platform for news and interaction, novice investors often rely on it for financial insights. When addressing regression problems with small datasets, such as demand or weather forecasting, XGBoost has shown excellent promise. Similarly, LSTM has proven to be a highly effective deep learning algorithm for forecasting tasks. One study found that LSTM outperformed XGBoost in forecasting Ether prices due to its ability to capture long-term dependencies and manage sequential data. These findings suggest that sentiment analysis on Twitter data could enhance cryptocurrency price predictions, with LSTM demonstrating particular effectiveness in this scenario [18].

XGBoost is widely recognized for its efficacy in fraud detection and resolving class imbalance issues that could lead to overfitting. By eliminating noisy and unnecessary data, the model selects compelling features for training and modifies parameters to optimize performance. A hybrid LSTM-XGBoost model demonstrated superior results compared with standalone algorithms, achieving improved effectiveness through feature selection. The hybrid model reduced training time and minimized overfitting, resulting in enhanced overall performance [19].

In the Wisconsin Breast Cancer Database, class imbalance presented a challenge where the majority class dominated precision metrics. To address this, oversampling techniques like SMOTE and Random Oversampling were employed. Meanwhile, tree-based machine learning algorithms such as Random Forest, Adaptive Boosting, and XGBoost improved model performance. Notably, XGBoost achieved enhanced accuracy in breast cancer detection, with 10-fold cross-validation yielding a score of 0.98 [20].

Liu and Wang proposed a credit card default prediction model that leverages XGBoost combined with RandomUnderSampler to address data imbalance issues. The study evaluated multiple machine learning techniques, including Random Forest, Logistic Regression, AdaBoost, LightGBM, and Support Vector Machines, with XGBoost outperforming all others in terms of accuracy and ROC AUC score. The final model achieved a ROC AUC score of 0.7882, a precision rate of 21.67%, a recall rate of 79.14%, and an F1 score of 33.85%. They highlight the effectiveness of combining advanced machine learning algorithms like XGBoost with robust data preprocessing techniques, bridging theoretical innovation with practical implementation for credit risk management [21].

Traditional models like SVM and KNN, which depend on limited features and require manual feature selection, leverage advanced deep learning techniques to automate feature extraction directly from raw data. This eliminates the labor-intensive and time-consuming process of manual feature engineering, particularly for complex datasets like network traffic. By employing LSTM and attention mechanisms, the proposed model learns hierarchical feature representations, enhancing accuracy in binary classification tasks. Experimental evaluations against state-of-the-art ML/DL models demonstrate its superior performance. However, it is essential to acknowledge the limitations of the proposed model and similar approaches, ensuring balanced assessments and ongoing refinement [22].

Network intrusion detection systems (NIDSs) are critical for safeguarding enterprise assets and enhancing cybersecurity. Despite significant advancements in deep learning-based NIDS, the evolving nature of network intrusions necessitates continued research. This study investigates the influence of various feature selection methods on the performance of LSTM-based NIDSs using the benchmark UNSW-NB15 dataset. Eight LSTM models, combined with different feature selection techniques, were evaluated. Interestingly, the LSTM model without feature selection outperformed those using specific methods, demonstrating the potential for deep learning models to autonomously learn effective representations. High accuracy in classification underscores the system's ability to differentiate between malicious and normal network traffic while minimizing false alarms [23].

Jun Ma and Congying Li provide a detailed comparison of the Probit and Logit models, offering insights into their relative strengths and practical applications. Key findings reveal that the Probit model excels in scenarios involving smaller datasets across varying levels of kurtosis, while the Logit model demonstrates superior performance with larger datasets, particularly those with leptokurtic distributions. Moreover, the independence of independent variables does not significantly influence the performance of either model, simplifying their application in certain research contexts [24].

By exploiting the sequential nature of transaction data, the LSTM effectively distinguishes between fraudulent and legitimate transactions. The early stopping mechanism halts training at the optimal point, preventing overfitting and improving generalization. The analysis shows a balance between recall and accuracy during training, enhancing fraud detection, reducing false alarms, and supporting real-time decisions for financial institutions. The model's flexibility and generalization make it a valuable tool in combating credit card fraud [25].

3. Methodology

Traditional statistical models, such as Probit and Logit, have long been favored for binary outcome prediction due to their interpretability and robust theoretical underpinnings. However, these models face inherent challenges, including limited adaptability to complex data structures, sensitivity to data characteristics, and restrictions in modeling non-linear relationships.

In contrast, machine learning (ML) models, though inherently more complex, deliver enhanced flexibility and predictive accuracy, especially when applied to high-dimensional or intricate datasets. These findings highlight the importance of selecting the appropriate modeling approach based on specific dataset characteristics and research objectives. By bridging traditional and modern methodologies, this study offers valuable guidance for researchers navigating diverse analytical challenges.

This study delves into the realm of using machine learning (ML) models to power a credit card default prediction system. The primary objective at hand is to unearth the most efficacious ML model for a freshly proposed credit card scoring dataset. This novel dataset, which contains intricate details of credit card transaction histories and customer profiles, undergoes rigorous testing with an array of machine learning algorithms, including LSTM and XGBoost. Before subjecting this wealth of data to the scrutiny of our ML models, several steps are taken, including data preprocessing techniques to eliminate any inconsistencies or anomalies within the dataset. Additionally, feature extraction helps distill crucial information from the plethora of data available, while feature selection ensures that only relevant features make their way into our analysis. Moreover, efforts are made towards balancing out skewed datasets using specialized techniques tailor-made for such scenarios.

Utilizing LSTM networks and XGBoost algorithms separately provides financial institutions with distinct advantages in enhancing the performance of credit card fraud detection models. The sequential learning capabilities of LSTMs and the ensemble learning power of XGBoost contribute to more robust and adaptive systems. By considering the strengths of each approach, organizations can tailor their strategies to better understand the evolving landscape of credit cards.

3.1. Preprocessing Data

Class imbalance is a common challenge in machine learning datasets, where minority class samples are significantly fewer than those of the majority class. The Synthetic Minority Oversampling Technique (SMOTE) addresses this by generating synthetic samples for the minority class through interpolation. This dataset lacks sequential relationships, SMOTE combined with a machine learning model is likely a better choice than MIDAS-LSTM (Multivariate and Dynamic Adaptation via LSTMs). MIDAS-LSTM can be prone to overfitting due to several reasons inherent to its architecture and application in time series or sequential data problems.

The process involves the following steps:

1. Identify class imbalances by comparing the minority and majority class distributions.
2. For each minority class instance, find its k-nearest neighbors.
3. Generate synthetic samples by interpolating between a chosen instance and its neighbors.

Mathematically, the generation of synthetic examples can be expressed as

$$X_{new} = X_i + \lambda(X_j - X_i), \lambda \in [0, 1] \quad (1)$$

This method creates synthetic examples along the line segment between X_i and X_j, effectively balancing the dataset.

After balancing, the enriched dataset is used to train machine learning models, allowing them to better capture the underlying patterns and reduce bias caused by imbalanced distributions.

3.2. Long Short-Term Memory (LSTM)

LSTM is a specialized type of recurrent neural network (RNN) architecture designed to address the issue of vanishing gradients in traditional RNNs. LSTMs are particularly well suited for analyzing sequential data that exhibits long-term dependencies between data. They have found widespread application in various domains, including natural language processing, time series analysis, and credit card fraud detection. Figure 1 represents the architecture of a single LSTM unit, which is composed of three key gates, input, forget, and output, which work together to regulate the flow of information through the unit [26].

Long short-term memory (LSTM) networks extend the functionality of standard recurrent neural networks (RNNs) by replacing conventional recurrent nodes with memory cells. These memory cells feature an internal state maintained through a self-connected recurrent edge with a fixed weight of 1, enabling gradients to propagate effectively across multiple time steps without vanishing or exploding. The term "Long Short-Term Memory" highlights the model's ability to balance long-term and short-term memory. Traditional RNNs store long-term memory in weights, which evolve gradually during training to encode general patterns in the data. Short-term memory exists as transient activations passed from one node to the next. LSTMs introduce an intermediate memory structure through memory cells, which are complex units constructed with interconnected nodes and incorporate multiplicative gates for enhanced functionality.

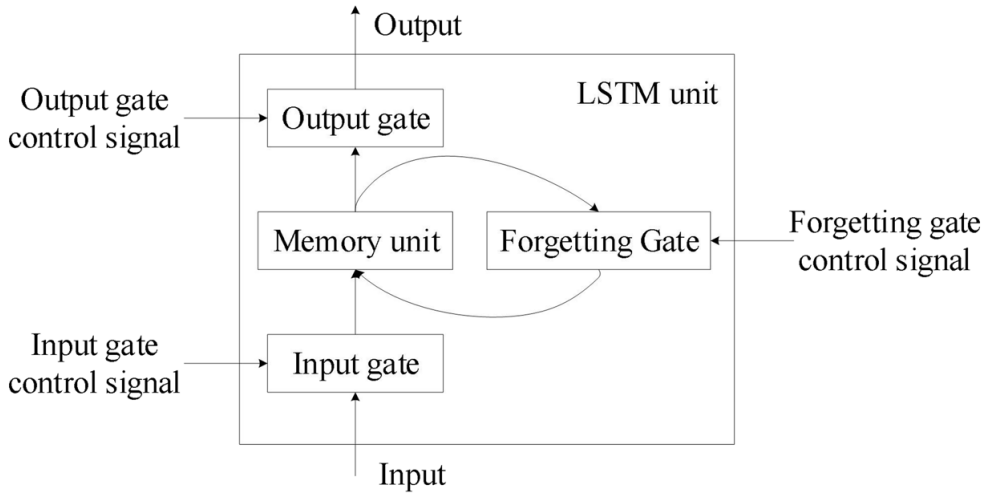

Figure 1. The structure of the LSTM.

3.2.1. Mathematical Representation of LSTM Components

Gated Mechanisms in Memory Cells

Each memory cell employs three types of gates, implemented as fully connected layers with sigmoid activations to control data flow within the cell:

1. Input Gate (I_t): Regulates the extent to which incoming data influences the internal state. It determines how much new information, represented by the input node (C_t), contributes to updating the cell state.

$$I_t = \sigma(X_t W_{x_i} + H_{t-1} W_{h_i} + b_i) \qquad I_t \in R_n \times h \tag{2}$$

2. Forget Gate (F_t): The forget gate decides whether to retain or reset parts of the cell's previous internal state (C_{t-1}). It enables the model to decide how much of the historical information should be preserved.

$$F_t = \sigma(X_t W_{x_f} + H_{t-1} W_{h_f} + b_f) \qquad F_t \in R_n \times h \tag{3}$$

3. Output Gate (O_t): This gate determines how much of the updated internal state (C_t) contributes to the output (H_t) at the current time step.

$$O_t = \sigma(X_t W_{x_o} + H_{t-1} W_{h_o} + b_o) \qquad O_t \in R_n \times h \tag{4}$$

being in all the equations: $X_t \in R_n \times d$; $H_{t-1} \in R_n \times h$; $W_{x_i}, W_{x_f}, W_{x_o} \in R_d \times h$; $W_{h_i}, W_{h_f}, W_{h_o} \in R_h \times h$; $b_i, b_f, b_o \in R_1 \times h$; $H_{t-1} \in R_n \times h$.

Mathematically, suppose that the batch size is n, the input size is d, and the number of hidden units is h, while W is the weight parameter and b shows the bias parameter. These dimensions define the structure of the input data and the LSTM's hidden layers. To regulate the flow of information, LSTMs utilize sigmoid activation functions (σ) to map input values to the range (0, 1).

In addition to the gates, LSTMs incorporate an input node (C_t), which represents the candidate state for updating the memory cell. Unlike the gates, the input node uses the *tanh* activation function, mapping values to the range $(-1, 1)$. This allows the input node to introduce both positive and negative contributions to the internal state.

$$\begin{gathered} C_t = tanh(X_t W_{x_c} + H_{t-1} W_{h_c} + b_c) \\ C_t \in R_n \times h, \quad W_{x_c} \in R_d \times h, \quad W_{h_c} \in R_h \times h, \quad b_c \in R_1 \times h \end{gathered} \tag{5}$$

Memory Cell Internal State and Hidden State

The memory cell's internal state (C_t) in an LSTM is updated dynamically at each time step through a combination of past memory and new information. This update is controlled by the input gate (I_t) and the forget gate (F_t), as represented by the following equation:

$$C_t = F_t \odot C_{t-1} + I_t \odot C_t \tag{6}$$

The forget gate (F_t) determines how much of the previous internal state (C_t) is retained. The input gate (I_t) controls how much of the new candidate state (C_t) is incorporated into the updated memory cell.

If the forget gate is always active ($F_t = 1$) and the input gate is always inactive ($I_t = 0$), the memory cell's internal state will remain constant forever; this enables LSTMs to maintain long-term dependencies in sequential data. The interplay between the gates allows the model to selectively update or reset its memory in response to new inputs.

The hidden state (H_t) serves as the output of the LSTM at each time step and is derived from the updated memory cell (C_t). Its calculation involves two steps:

1. Apply the *tanh* activation function to scale the memory cell's internal state (C_t) to the range $(-1, 1)$.
2. Use the output gate (O_t) to control how much of this information is propagated to subsequent layers.

The hidden state is expressed mathematically as

$$H_t = O_t \odot tanh(C_t) \quad H_t \in R_n \times h \tag{7}$$

- When the output gate (O_t) is fully active ($O_t \approx 1$), the internal state significantly influences the output.
- When the output gate is inactive ($O_t \approx 0$), the memory cell's impact on the network's subsequent layers is minimal.

This gating mechanism ensures that the hidden state remains in the range $(-1, 1)$ offering controlled and flexible propagation of information to other layers of the model.

Figure 2 illustrates the architecture of an LSTM unit, emphasizing the flow of information through its components. This structure enables efficient information transfer across time steps in sequential data.

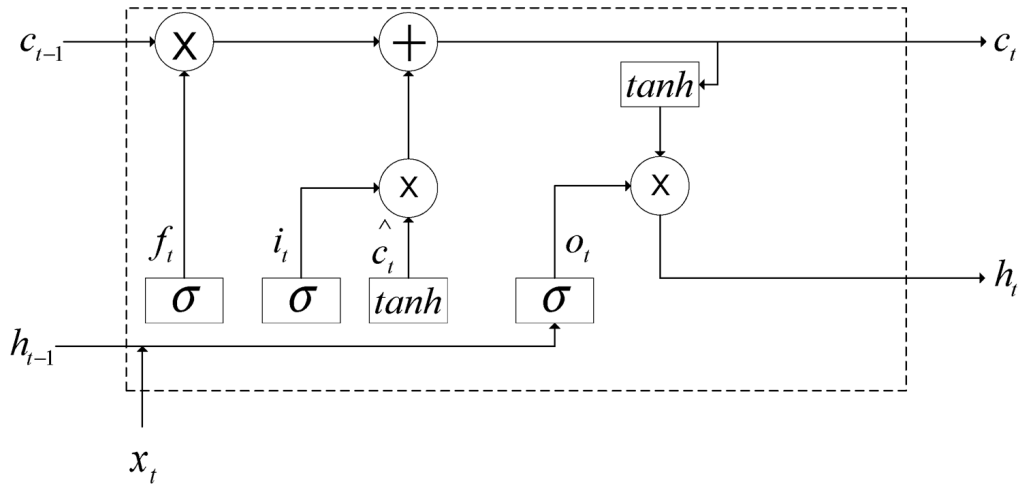

Figure 2. The LSTM model.

3.3. XGBoost (Extreme Gradient Boosting)

XGBoost is an ensemble learning algorithm designed to build a sequence of weak learners, typically decision trees, and combine their predictions to form a robust predictive model. The algorithm iteratively minimizes a specified loss function by sequentially adding weak learners, thereby improving overall model performance. This approach makes XGBoost highly effective for both regression and classification tasks. In supervised learning, the model refers to a mathematical structure that predicts y_i based on input features x_i. For instance, in a linear model, the prediction is expressed as

$$\widehat{y_i} = \sum_j \theta_j x_{ij} \tag{8}$$

The prediction value can have different interpretations, depending on the task, i.e., regression or classification. The parameters are the undetermined part that we need to learn from the data. In linear regression problems, the parameters are the coefficients θ. The task of training the model amounts to finding the best parameters θ that best fit the training data x_i and labels y_i. In order to train the model, we need to define the objective function to measure how well the model fit the training data. A salient characteristic of objective functions is that they consist of two parts, training loss and regularization term:

$$obj(\theta) = L(\theta) + \Omega(\theta) \tag{9}$$

where L is the training loss function, and Ω is the regularization term. The training loss measures how predictive our model is with respect to the training data. A common choice of L for regression is the Mean Squared Error (MSE), which is given by

$$L(\theta) = \sum_i (y_i - \widehat{y_i})^2 \tag{10}$$

Another commonly used loss function, in this case for binary classification, is logistic loss:

$$L(\theta) = \sum_i [y_i ln(1 + e^{-\widehat{y_i}}) + (1 - y_i) ln(1 + e^{\widehat{y_i}})] \tag{11}$$

The prediction scores of each individual tree are summed up to obtain the final score. Mathematically, we can write our model in the following form:

$$Obj = \sum_i^n L(y_i, \widehat{y_i}) + \sum_k^K \Omega(f_k) \tag{12}$$

where $L(y_i, \widehat{y_i})$ is the training loss, measuring the difference between predicted and actual values.

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_j^T w_j^2 \tag{13}$$

where $\Omega(f_k)$ is the regularization term, penalizing model complexity by considering the number of leaves T and leaf weights ω_j, while γ controls the complexity penalty and λ is the regularization coefficient for leaf weights.

During training, XGBoost employs additive learning, where new trees are added to minimize the loss function iteratively. For a given step t, the objective is approximated using a second-order Taylor expansion:

$$Obj_t = \sum_i^n [g_i f_t(x_i) + \frac{1}{2} h_i f_t(X_i)^2] + \Omega(f_t) \tag{14}$$

Here, g_i and h_i are the first- and second-order gradients of the loss function with respect to the predictions.

As enumerating all possible trees is intractable, we will try to optimize one level of the tree at a time. Specifically, we try to split a leaf into two leaves, and the score gain is

$$Gain = \frac{1}{2}[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}] - \gamma \quad (15)$$

Figure 3 illustrates the gradient boosting process, where decision trees are trained sequentially. Each tree corrects the residual errors of the previous ones, and their predictions are combined to build a strong final model [27].

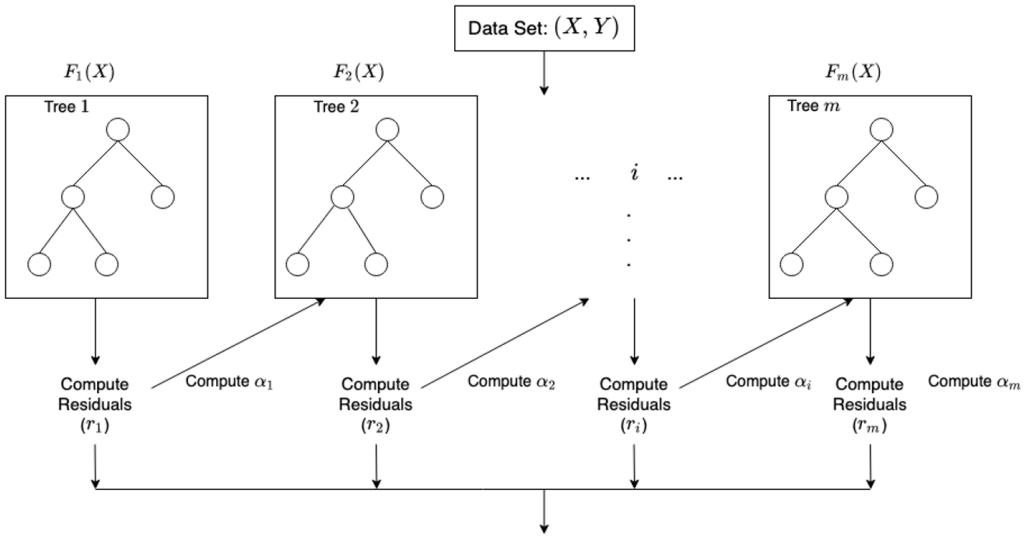

Figure 3. The structure of XGBoost.

4. Result Analysis and Findings

4.1. Data Collection

To find the most accurate model for predicting financial distress, we use the Kaggle dataset (https://www.kaggle.com/code/josh1337/bankchurners/input, (accessed on 19 January 2024)). This dataset consists of information on 10,127 customers, encompassing details such as age, income, marital status, credit card limit, and credit card category, among others. With 23 features in total, approximately 16.1% of customers have either discontinued their services or are inactive, while the remaining 83.9% are actively engaged. Consequently, training our model to accurately predict customer loyalty presents a significant challenge. To address this, we undertake the selection of pertinent features crucial for the prediction task. Subsequently, we normalize or standardize numerical attributes and convert categorical variables into a numerical format to facilitate model training.

To provide a clear understanding of the dataset, Table 1 presents the summary statistics of key numerical features, including their mean, standard deviation, minimum, and maximum values, while Table 2 presents the frequency distribution of categorical features.

Table 1. Descriptive statistics for key numerical features.

Feature	Mean	Std. Dev.	Min.	Max.
Customer_Age	46.3	8.0	26	73
Months_on_book	36.9	7.9	13	56
Credit_Limit	8634.9	9087.2	1438	34,516
Total_Revolving_Balances	1162.8	815.2	0	2517
Total_Trans_Ct	64.9	23.4	10	139
Total_Trans_Amt	3994.4	2276.3	510	18,484
Total_Ct_Chng_Q4_Q1	0.76	0.22	0.0	3.71

Table 2. Frequency distribution of categorical features.

Feature	Categories	Frequency (%)
Gender	Male/Female	52.5/47.5
Marital_Status	Married/Single/Others	57.4/35.4/7.2
Education_Level	Graduate/High School/Others	53.7/27.3/19.0
Income_Category	$< \$40K/ \$40K–\$80K/ > \$80K$	23.3/33.4/43.3

4.2. Feature Selection and Engineering

The selection of relevant features that significantly contribute to predicting creditworthiness or customer churn can be achieved through statistical methods, domain expertise, or machine learning algorithms. In this study, a correlation matrix was employed to examine the relationships between variables in the dataset, providing insights into the strength and direction of these relationships.

Correlation coefficients were interpreted as follows: values approaching +1 signify a strong positive correlation, values approaching -1 indicate a strong negative correlation, and values close to 0 suggest minimal or no linear relationship between the variables. As depicted in Figure 4, certain variables exhibited notable correlations. For instance, Average Open to Buy and Credit Limit showed the highest positive correlation. Additionally, Total Transaction Amount and Total Transaction Count demonstrated a correlation of 0.81, reflecting the natural tendency for transaction amounts to increase with the frequency of transactions. Similarly, Customer Age and Months on Book displayed a correlation of 0.79, likely attributable to younger customers recently gaining eligibility for credit card ownership. In contrast, an inverse correlation (-0.54) was observed between Average Utilization Ratio and Average Open to Buy, highlighting a complementary relationship between these variables.

Based on this analysis, the following features were identified as key predictors due to their strong correlations and potential relevance in a predictive model:

- Total Transaction Amount or Total Transaction Count.
- Average Utilization Ratio.
- Credit Limit.
- Months on Book or Customer Age (as these variables reflect similar underlying trends).
- Total Revolving Balance.
- Average Open to Buy.

These selected features capture a comprehensive profile of customer behavior, credit utilization patterns, and the duration of their relationship with the financial institution, making them valuable inputs for predictive modeling.

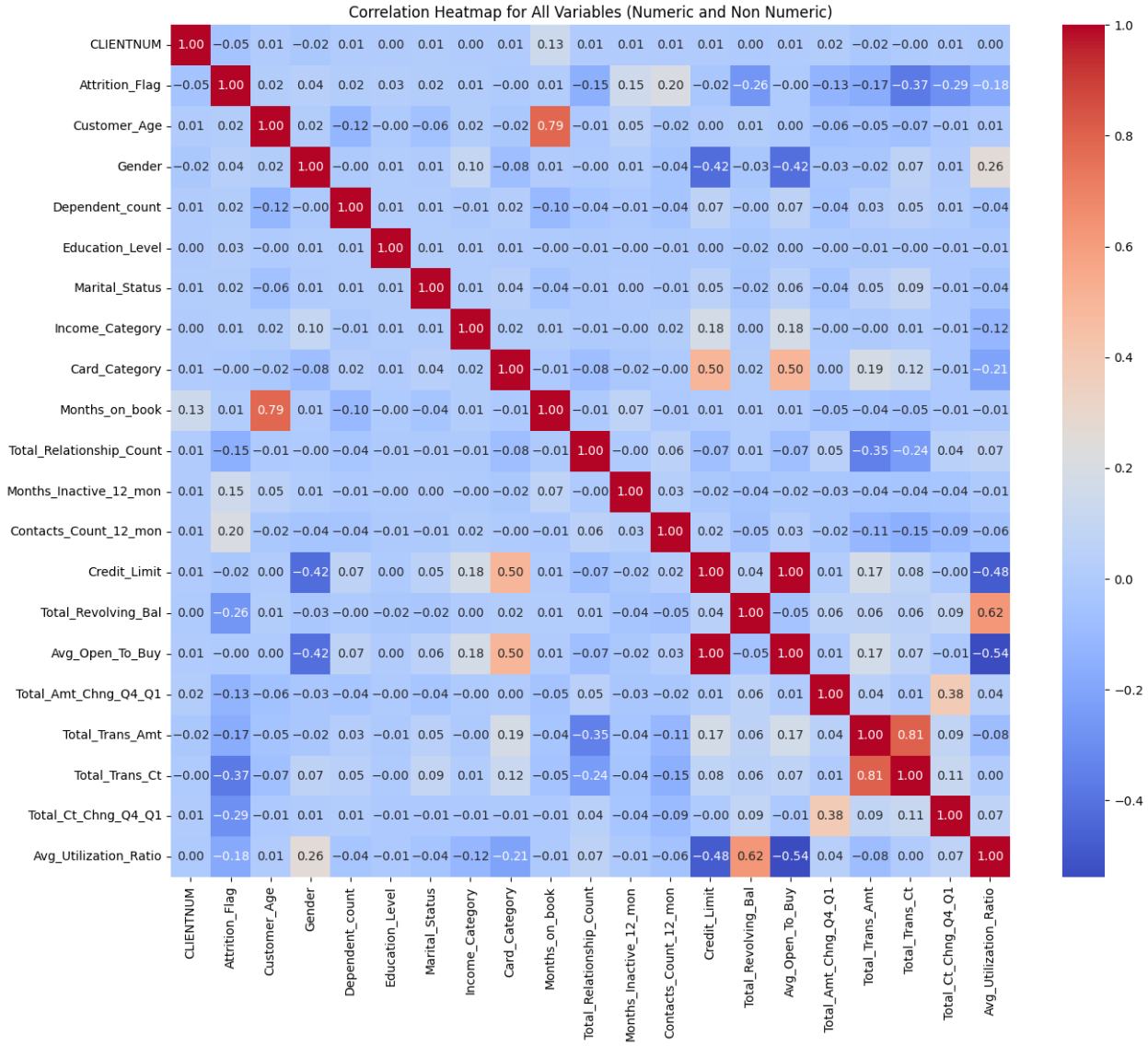

Figure 4. Feature correlation heatmap.

4.3. Data Preprocessing and Balancing

When trying to label a dataset into a category based on an input dataset, or ensuring accurate and reliable results, it is imperative to properly cleanse and pre-process the data. This involves rectifying any inconsistencies or flaws present within the dataset before proceeding further. In addition, standardizing or normalizing features may be necessary based on individual requirements and circumstances. Then, the dataset must be split into training and testing sets. For time series data, ensure a chronological split to avoid data leakage. In our case, we used 75% of dataset for training and 25% of dataset for testing.

SMOTE operates by producing fabricated instances for the underrepresented category, namely the Attrited Customer, in order to achieve equilibrium of class distribution. X refers to the matrix of features representing independent variables, while Y represents the target variable denoting class labels or dependent variables. The X collection encompasses independent variables, with the exception of Clinet_Num and Attrition_Flag. On the other hand, Y comprises the dependent variable, specifically representing Attrition_Flag. The dataset includes the details of 10,127 customers, of which 16.1% are attrited or inactive and 83.9% are excited.

Figure 5 comprises the main visualizations that help us understand the relationships between key numerical variables in our dataset and provide a comprehensive view of the interrelationships between important variables. blue indicates "Existing Customer" and orange indicates "Attrited Customer".

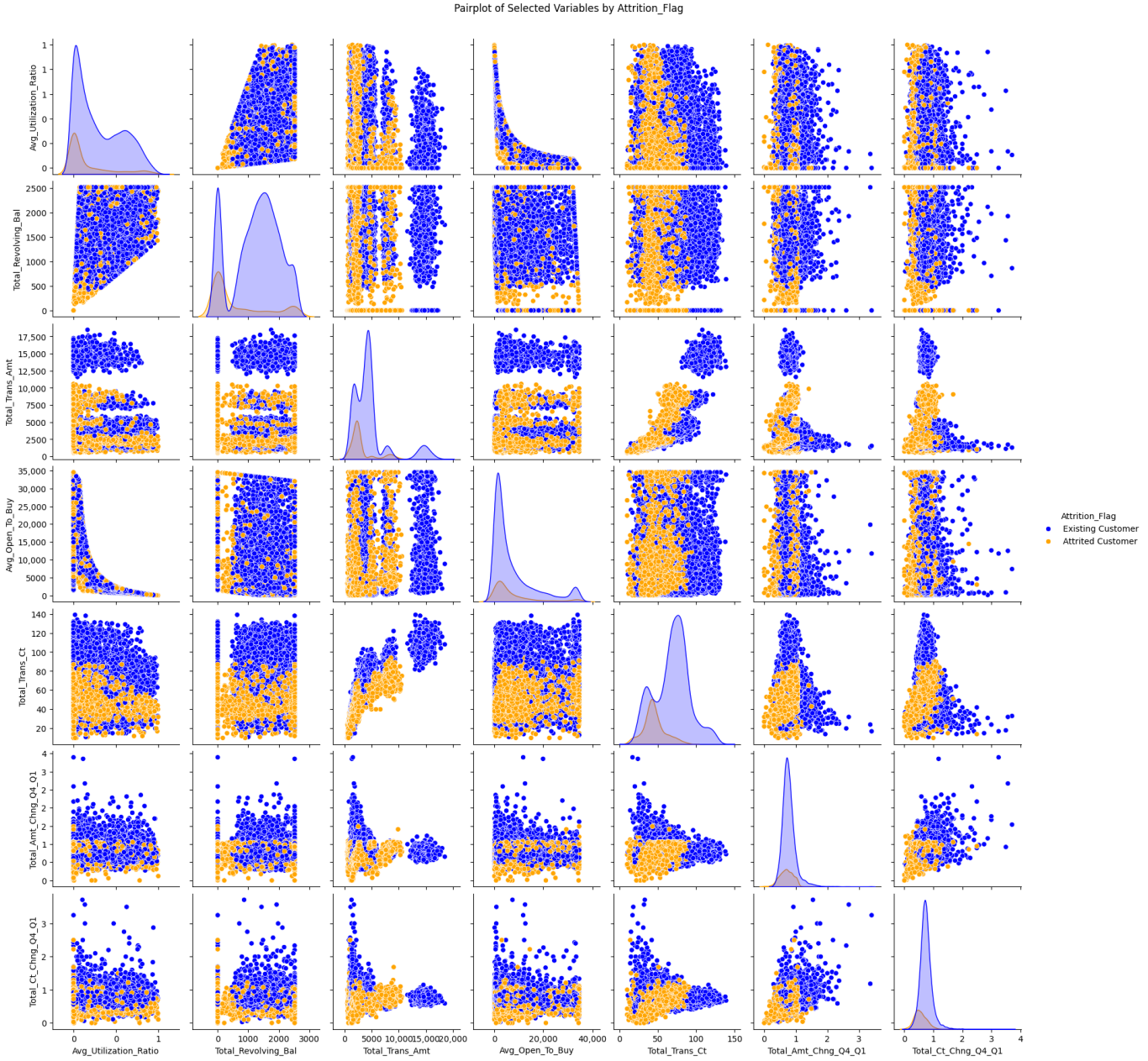

Figure 5. The distribution of the dataset.

4.4. Model Selection

Traditional statistical models, such as Probit and Logit, have long been preferred for binary outcome prediction due to their interpretability and strong theoretical foundation. However, these models face limitations, such as their inability to capture complex, non-linear relationships, sensitivity to data characteristics, and restrictions when handling intricate data structures. Below is a concise summary of the methodology for the Logit, Probit, and MLP models:

- Logit (Logistic Regression): A linear model that predicts binary outcomes using the logistic function. The formula is

$$P(y = 1|X) = \frac{1}{1 + exp(-(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p))} \tag{16}$$

  where X represents the input features, and β is the model parameters.
- Probit: Similar to Logit but assumes a normal distribution for the underlying error terms. The formula is

$$P(y = 1|X) = \Phi(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p) \tag{17}$$

  where Φ is the cumulative distribution function (CDF) of the standard normal distribution.
- MLP (Multilayer Perceptron): A neural network model with hidden layers that uses neurons with sigmoid activation functions to compute the binary outcome. The formula is

$$a^{[1]} = \sigma(W^{[1]} X + b^{[1]}) \tag{18}$$

  where W represents weights, and $\sigma(x)$ is the sigmoid activation.

While Logit and Probit are effective for linear relationships, they are limited in handling non-linear patterns, which is where MLP can offer an advantage. However, MLP requires more careful tuning and regularization to avoid overfitting and achieve optimal performance.

4.4.1. Long Short-Term Memory (LSTM)

A Keras sequential model is utilized to construct the LSTM neural network for binary classification. The architecture begins with an input layer, shaped according to the number of features in the training dataset. An LSTM layer with 32 units and ReLU activation is then added to capture the temporal dependencies within the sequential data. To further increase the complexity of the model, a Dense layer with 16 units and ReLU activation is incorporated. Finally, a Dense layer with a single unit and a sigmoid activation function is used, which is a typical choice for binary classification tasks.

For model compilation, binary cross-entropy is selected as the loss function due to its suitability for binary classification problems. The Adam optimizer is employed with a learning rate of 0.001. During the training process, the model's performance is monitored using the accuracy metric to assess its effectiveness at each step.

Training is carried out using the fit method, with a batch size of 64 and a total of 25 epochs. Additionally, validation data are used to evaluate model performance at each epoch. This setup aims to capture the underlying patterns in sequential data for accurate binary classification.

4.4.2. XGBoost (Extreme Gradient Boosting)

XGBoost, a powerful machine learning algorithm, is employed with the XGBClassifier. The data are preprocessed using StandardScaler for feature scaling, and the model is trained with a set of hyperparameters specific to XGBoost, including parameters such as C, degree, gamma, and kernel. Stratified K-fold cross-validation with five folds is used to ensure that each fold contains a proportional representation of the target class. A GridSearchCV approach is used to find the optimal hyperparameters for the XGBoost model, which allows for systematic tuning of parameters to enhance model performance.

4.5. Evaluation Models

Monitoring validation accuracy helps detect issues like overfitting or underfitting, ensuring the model's overall performance is accurate and reliable. Monitoring metrics such as accuracy, loss, validation accuracy, and validation loss throughout the training process ensure the model's reliability and robustness. These metrics, alongside precision, recall, and F1 score, provide a holistic view of the model's performance, guiding improvements and preventing issues like overfitting. Linear models like Logit and Probit are insufficient for the dataset due to their inability to capture complex, non-linear relationships and feature interactions. Additionally, these models do not effectively address class imbalance, leading to suboptimal performance on the minority class. While the MLP model improves accuracy, it remains limited by challenges such as handling imbalanced data, hyperparameter optimization, and the risk of overfitting without proper regularization. Logit achieved an accuracy of 72.71%, and Probit achieved 72.91%, demonstrating similar performance but limited in handling complex patterns due to their linear assumptions. MLP achieved a much higher accuracy of 91.32%, leveraging its ability to model non-linear relationships, making it more effective for complex datasets. These factors underscore the need for more advanced models to achieve better predictive performance.

The LSTM model exhibits superior performance in handling imbalanced datasets, as evidenced by its higher F1 score, precision, and recall, as can be seen in Table 3. These metrics suggest that LSTM is particularly effective for tasks such as credit card fraud detection, where detecting minority classes (fraudulent transactions) is crucial. The model's ability to achieve perfect classification with a validation accuracy of 100% further underscores its capability to capture complex patterns within sequential data. In contrast, the XGBoost model, while achieving a lower validation loss of 0.08, shows slightly reduced overall performance compared with LSTM, particularly in terms of precision and recall. Although XGBoost delivers robust performance with 97% test accuracy and 100% training accuracy, its ability to handle the class imbalance is not as strong as LSTM's, which results in lower F1 scores and recall.

Table 3. Performance Evaluation of LSTM and XGBoost Models.

Metric	LSTM	XGBoost
Test Accuracy	1	0.97
Train Accuracy	0.99	1
Validation Accuracy	1	1
Validation Loss	1.28	0.08
F1 score	1	0.91
Precision	1	0.92
Recall	1	0.90

4.6. Findings

4.6.1. Test Accuracy

Test accuracy represents the proportion of correctly classified instances from the test dataset, which consists of data that the model has not encountered during training. LSTM achieved a perfect test accuracy of 100%, demonstrating flawless classification of the test data. XGBoost, on the other hand, reached 97% test accuracy, which is impressive but slightly lower than LSTM's, indicating the model's ability to generalize to unseen data.

4.6.2. Train Accuracy

Train accuracy measures how well the model performs on the data it has already been trained on. LSTM achieved 99% accuracy on the training dataset, reflecting its strong

ability to learn the patterns in the training data. XGBoost achieved perfect training accuracy (100%), meaning it perfectly fits the training data, which is typical for powerful models like XGBoost but may also signal a risk of overfitting.

4.6.3. Validation Accuracy

Validation accuracy assesses the model's performance on a separate validation dataset, which was not used during training. This metric helps determine how well the model generalizes to data it has not seen before. LSTM achieved 100% validation accuracy, suggesting excellent generalization and no signs of overfitting. XGBoost also achieved 100% validation accuracy, showing similar performance but potentially more sensitivity to dataset complexity.

4.6.4. Validation Loss

Validation loss measures the discrepancy between the model's predicted outputs and the actual values from the validation dataset. A lower validation loss indicates better generalization. LSTM's validation loss was 1.28, which is relatively high and suggests potential overfitting, where the model may have learned the noise in the training data instead of just the underlying patterns. In contrast, XGBoost achieved a much lower validation loss of 0.08, indicating better generalization and less risk of overfitting compared with LSTM.

4.6.5. F1 Score

The F1 score is the harmonic mean of precision and recall, offering a balanced measure of model performance, particularly important in imbalanced datasets. It is critical in tasks such as fraud detection, where identifying rare events is vital. LSTM achieved a perfect F1 score of 1.00, indicating an excellent balance between precision and recall. XGBoost scored 0.91, which is still strong but slightly lower than LSTM, suggesting it might not be as effective at balancing both precision and recall.

4.6.6. Precision

Precision evaluates the proportion of true positive predictions (correct fraud predictions) out of all instances predicted as positive (fraudulent). LSTM achieved perfect precision (1.00), meaning every prediction of fraud was accurate. XGBoost achieved a precision of 0.92%, indicating that while it is quite precise, about 8% of fraud predictions could be false positives.

4.6.7. Recall

Recall measures the proportion of true positive predictions (correct fraud predictions) out of all the actual fraud cases. LSTM achieved perfect recall (1.00), meaning it detected all fraudulent transactions in the dataset. XGBoost achieved a recall of 0.90, meaning it missed 10% of the fraud cases, which is a minor disadvantage compared with LSTM's perfect recall.

Figure 6 displays the training and validation accuracy and loss of the LSTM model over 25 epochs, emphasizing its learning trajectory and high generalization capacity.

Figure 7 likely visualizes the performance metrics across the five folds of cross-validation for the XGBoost model. The left plot shows the validation loss for each fold, which reflects how well the model fits the validation data during each fold. The right plot displays the validation accuracy, indicating how accurately the model predicts the target variable in each fold.

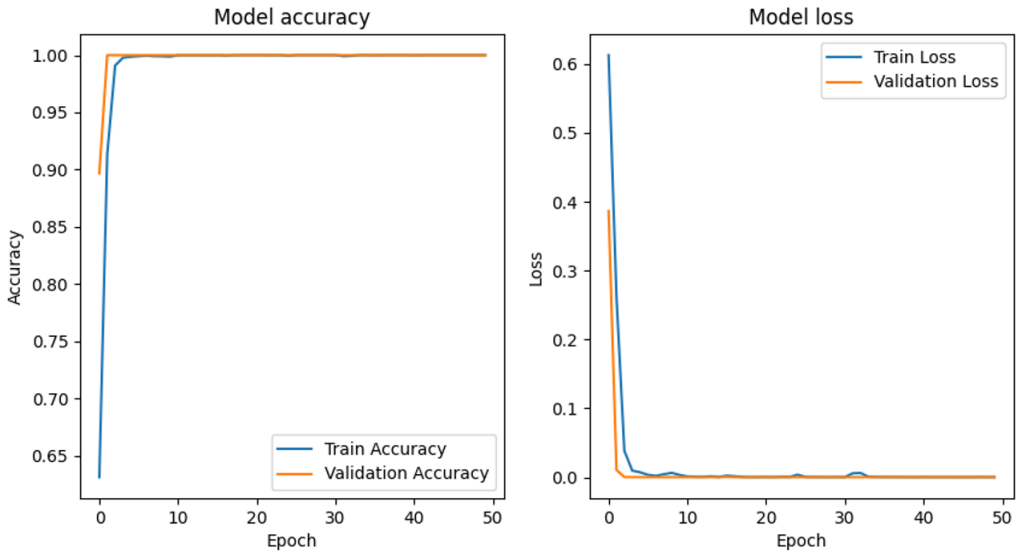

Figure 6. The LSTM plot.

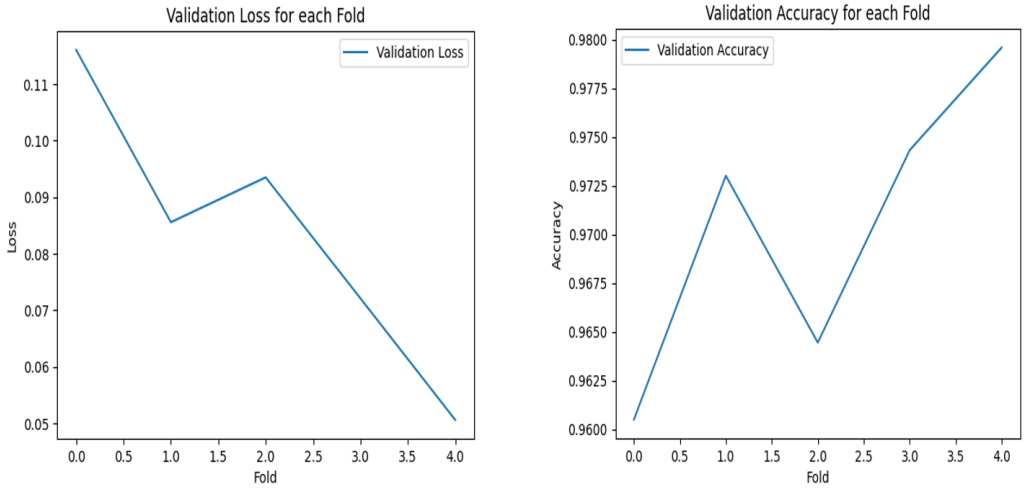

Figure 7. The XGBoost model plot.

5. Conclusions and Future Work

This study concludes that while both LSTM and XGBoost exhibit strong performance in risk-predicting credit cards in contrast with the traditonal method of binary classification, the LSTM model demonstrates a clear advantage when dealing with imbalanced datasets. Specifically, the LSTM model achieved a higher accuracy rate, showcasing its suitability for tasks involving sequential data and temporal patterns.

Despite the competitive performance of XGBoost, particularly in regression and classification tasks with smaller datasets, its precision and F1 score were slightly lower compared with LSTM. This suggests that LSTM is better equipped to capture the complexities inherent in imbalanced financial datasets.

This research highlights the importance of addressing data imbalance and optimizing computational setups to achieve high model accuracy. Additionally, it introduces mathematical formulations for computing LSTM outputs, providing a foundation for further advancements in this domain.

This study highlights the strong predictive capabilities of LSTM models but acknowledges their computational intensity and resource demands, which may limit their deployment in environments with constrained resources. Additionally, the analysis primarily

focuses on static performance metrics, overlooking the potential advantages of adaptive or hybrid models that integrate the strengths of LSTM and XGBoost. Another notable limitation is the interpretability of the models. While XGBoost offers insights into feature importance, LSTM's lack of transparency could pose challenges in finance, where explainability is crucial

Future research should prioritize optimizing LSTM models for real-time fraud detection by enhancing computational efficiency, ensuring seamless integration into financial systems, and testing their robustness under high data inflow conditions. Addressing class imbalance in financial datasets is another critical area, which can be tackled through innovative approaches such as synthetic data generation using GANs, cost-sensitive learning, and adaptive resampling techniques to improve model accuracy and reliability. Additionally, comprehensive benchmarking studies are necessary to compare the performance of LSTM models with emerging architectures like spiking neural networks (SNNs) and graph neural networks (GNNs). Using standardized datasets and evaluation metrics, these studies can identify the most effective architectures for various financial applications, driving advancements in the field.

Author Contributions: K.K.: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing—Original Draft, Visualization. A.G.-D.: Conceptualization, Writing—Review & Editing, Supervision, Project administration. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors, or any external funding.

Data Availability Statement: Data available on the Kaggle webpage (https://www.kaggle.com/code/josh1337/bankchurners/input).

Conflicts of Interest: The authors declare no conflicts of interest, they have no affiliations with or involvement in any organization or entity with any financial or non-financial interest in the subject or materials discussed in this manuscript.

References

1. Said, I.; Qu, Y. Improving the Performance of Loan Risk Prediction based on Machine Learning via Applying Deep Neural Networks. *Eur. J. Electr. Eng. Comput. Sci.* **2023**, *7*, 31–37. [CrossRef]
2. Abdulghani, A.Q.; UCAN, O.N.; Alheeti, K.M.A. Credit Card Fraud Detection Using XGBoost Algorithm. In Proceedings of the 2021 14th International Conference on Developments in eSystems Engineering (DeSE), Sharjah, United Arab Emirates, 7–10 December 2021; pp. 487–492. [CrossRef]
3. Ahmed, A.N.; Saini, R. A Survey on Detection of Fraudulent Credit Card Transactions Using Machine Learning Algorithms. In Proceedings of the 2023 3rd International Conference on Intelligent Communication and Computational Techniques (ICCT), Jaipur, India, 19–20 January 2023; pp. 1–5. [CrossRef]
4. Benchaji, I.; Douzi, S.; El Ouahidi, B. Credit Card Fraud Detection Model Based on LSTM Recurrent Neural Networks. *J. Adv. Inf. Technol.* **2021**, *12*, 113–118. [CrossRef]
5. Deng, A.; Zhang, H.; Wang, W.; Zhang, J.; Fan, D.; Chen, P.; Wang, B. Developing Computational Model to Predict Protein-Protein Interaction Sites Based on the XGBoost Algorithm. *Int. J. Mol. Sci.* **2020**, *21*, 2274. [CrossRef] [PubMed]
6. Gao, J.; Sun, W.; Sui, X. Research on Default Prediction for Credit Card Users Based on XGBoost-LSTM Model. *Discret. Dyn. Nat. Soc.* **2021**, *2021*, 5080472. [CrossRef]
7. Yasasvia, P.; Kumarb, S.M. Improve Accuracy in Prediction of Credit Card Approval Using Novel XGboost Compared with Random Forest. In *Advances in Parallel Computing Algorithms, Tools and Paradigms*; Hemanth, D., Ed.; IOS Press: Amsterdam, The Netherlands, 2022; pp. 582–588. [CrossRef]
8. Mohmad, Y.A. Credit Card Fraud Detection Using LSTM Algorithm. *Wasit J. Comput. Math. Sci.* **2022**, *1*, 26–35. [CrossRef]
9. Hossain, M.N.; Hassan, M.M.; Monir, R.J. Analyzing the Classification Accuracy of Deep Learning and Machine Learning for Credit Card Fraud Detection. *Asian J. Converg. Technol.* **2022**, *8*, 31–36. [CrossRef]

10. Gicic, A.; Donko, D. Proposal of a model for credit risk prediction based on deep learning methods and SMOTE techniques for imbalanced dataset. In Proceedings of the 2023 XXIX International Conference on Information, Communication and Automation Technologies (ICAT), Sarajevo, Bosnia and Herzegovina, 11–14 June 2023; pp. 1–6. [CrossRef]
11. Mekruksavanich, S.; Jantawong, P.; Jitpattanakul, A. LSTM-XGB: A New Deep Learning Model for Human Activity Recognition based on LSTM and XGBoost. In Proceedings of the 2022 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON), Chiang Rai, Thailand, 26–28 January 2022; pp. 342–345. [CrossRef]
12. John, M.P.; Murali, G. Performance Enhancement and Comparative analysis for Credit Approval Using XGBoost, SVM and Multi-Layer Perceptron. In Proceedings of the 2022 IEEE Global Conference on Computing, Power and Communication Technologies (GlobConPT), New Delhi, India, 23–25 September 2022; pp. 1–4. [CrossRef]
13. Raval, J.; Bhattacharya, P.; Jadav, N.K.; Tanwar, S.; Sharma, G.; Bokoro, P.N.; Elmorsy, M.; Tolba, A.; Raboaca, M.S. RaKShA: A Trusted Explainable LSTM Model to Classify Fraud Patterns on Credit Card Transactions. *Mathematics* **2023**, *11*, 1901. [CrossRef]
14. Raj, A.T.; Shobana, J.; Nassa, V.K.; Painuly, S.; Savaram, M.; Sridevi, M. Enhancing Security for Online Transactions through Supervised Machine Learning and Block Chain Technology in Credit Card Fraud Detection. In Proceedings of the 2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Kirtipur, Nepal, 11–13 October 2023; pp. 241–248. [CrossRef]
15. Priscilla, C.V.; Prabha, D.P. Influence of Optimizing XGBoost to handle Class Imbalance in Credit Card Fraud Detection. In Proceedings of the 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 20–22 August 2020; pp. 1309–1315. [CrossRef]
16. Priscilla, C.V.; Prabha, D.P. A two-phase feature selection technique using mutual information and XGB-RFE for credit card fraud detection. *Int. J. Adv. Technol. Eng. Explor.* **2021**, *8*, 1656–1668. [CrossRef]
17. Alonso, A.; Carbó, J.M. *Understanding the Performance of Machine Learning Models to Predict Credit Default: A Novel Approach for Supervisory Evaluation*; Technical Report 2105; Banco de Espana Working Paper No. 2105; Banco de Espña: Madrid, Spain, 2021. [CrossRef]
18. Sathiyapriya, K.; Vankadara, S.; Babu, K.S.; Muralidharan, M. Performance Comparison of LSTM and XGBOOST for Ether Price Prediction from Spam Filtered Tweets. In Proceedings of the 2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS), Coimbatore, India, 9–11 February 2023; pp. 650–655. [CrossRef]
19. Poornima, R.; Elangovan, M.; Nagarajan, G. Network attack classification using LSTM with XGBoost feature selection. *J. Intell. Fuzzy Syst.* **2022**, *43*, 971–984. [CrossRef]
20. N, S.S.; Purnomo, M.H.; Purwitasari, D.; Yuniarno, E.M. Synthesis Ensemble Oversampling and Ensemble Tree-Based Machine Learning for Class Imbalance Problem in Breast Cancer iagnosis. In Proceedings of the 2022 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), Surabaya, Indonesia, 22–23 November 2022; pp. 1–7. [CrossRef]
21. Liu, X.; Wang, S. Credit Card Default Prediction Based on XGBoost. In Proceedings of the 2024 International Conference on Machine Intelligence and Digital Applications, Ningbo, China, 30–31 May 2024; MIDA '24; pp. 374–382. [CrossRef]
22. Alsharaiaha, M.A.; Abu-Sharehaa, A.A.; Abualhaj, M.; Baniata, L.H.; Al-saaidah, A.; Kharma, Q.M.; Al-Zyoud, M. An innovative network intrusion detection system (NIDS): Hierarchical deep learning model based on Unsw-Nb15 dataset. *Int. J. Data Netw. Sci.* **2024**, *8*, 709–722. [CrossRef]
23. Kottilingal, S. Comparative Analysis of Feature Selection Techniques for LSTM Based Network Intrusion Detection Models. *Int. J. Netw. Secur. Its Appl.* **2024**, *16*, 1–11. [CrossRef]
24. Ma, J.; Li, C. A comparison of Logit and Probit models using Monte Carlo simulation. In Proceedings of the 2021 40th Chinese Control Conference (CCC), Shanghai, China, 26–28 July 2021; pp. 8963–8967. [CrossRef]
25. Jhansi Ida, S.; Balasubadra, K.; R R, S.; T, L.N. Enhancing Credit Card Fraud Detection through LSTM-Based Sequential Analysis with Early Stopping. In Proceedings of the 2024 2nd International Conference on Networking and Communications (ICNWC), Chennai, India, 2–4 April 2024; pp. 1–6. [CrossRef]
26. Pisa, I.; Morell, A.; Vicario, J.L.; Vilanova, R. Denoising Autoencoders and LSTM-Based Artificial Neural Networks Data Processing for Its Application to Internal Model Control in Industrial Environments—The Wastewater Treatment Plant Control Case. *Sensors* **2020**, *20*, 3743. [CrossRef] [PubMed]
27. Upadhyay, J.; Gonsalves, T. Robust and Lightweight System for Gait-Based Gender Classification toward Viewing Angle Variations. *AI* **2022**, *3*, 538–553. [CrossRef]