EXPLORING DEEP LEARNING AND MACHIINE LEARNING TECHNIQUES PREVENT PAYMENT FRAUDS WITH THE HELP OF ENSEMBLE LEARNING

By

Yurii Zmytrakov

BTech, Zaporizhzhia National Technical University, 2014

A Major Research Project

presented to Toronto Metropolitan University

In partial fulfilment of the

requirements for the degree of

Master of Science

in the Program of

Data Science and Analytics

Toronto, Ontario, Canada, 2023

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A MAJOR RESEARCH PROJECT(MRP)

I hereby declare that I am the sole author of this Major Research Paper. This is a true copy of the MRP, including any required final revisions.

I authorize Toronto Metropolitan University to lend this MRP to other institutions or individuals for the purpose of scholarly research.

I further authorize Toronto Metropolitan University to reproduce this MRP by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my MRP may be made electronically available to the public.

Yurii Zmytrakov

Exploring deep learning and machine learning techniques

to prevent payment frauds with the ensemble learning

Yurii Zmytrakov

Master of Science 2023

Data Science and Analytics

Toronto Metropolitan University

ABSTRACT

The purpose of this research project is to develop a generalized model that will be preventing fraudulent online transactions, this binary classification model can be implemented into online payment platforms that are used by online retailers, and ultimately help online retailers reduce losses caused by fraudulent activities. Furthermore, this paper will identify what factors contribute to the accuracy and effectiveness of fraud transactions classifier algorithms, and how those algorithms can be optimized to reduce the false positives and false negatives. To detect fraudulent activities, features will be constructed based on domain knowledges of industry experts and based on the best practices in Machine and Deep Learning.

Keywords: Credit card fraud detection, Ensemble voting classifier, Binary classification model, Synthetic minority oversampling technique (SMOTE), Machine Learning for cyber-security.

ACKNOWLEDGEMENTS

I am very thankful to Dr. Fahid Shirazi for his support and assistance in making this project happen. Dr. Fahid Shirazi was my supervisor for this Major Research Project, and he has been supportive throughout the term to guide and direct my research and provide valuable suggestions. I would also like to thank him for introducing me to this challenge as well, along with acquiring the dataset for me, without which this project would not have materialized. I appreciate all the support and guidance that has gone into making this project.

Thank you, Dr. Fahid Shirazi.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

This report presents an overview of the domain of research chosen for my major research project. It includes detailed descriptions of the dataset I intend to utilize, as well as the background information relevant to this study. Additionally, the report delves into the research problem itself and outlines the methodology adopted to address it. To gain a comprehensive understanding of prior work in this field, the report incorporates a literature review focusing on segmentation as an application of neural networks, ranging from specific segmentation perspectives to more general analyses. Furthermore, the findings from the experimentation are discussed, along with potential future avenues for further contributions.

A. Background

The world has become increasingly digitalized, cybersecurity is a crucial aspect of protecting sensitive information and preventing fraudulent activities. With the rise of online retailing giants, such as Amazon, Apple, Walmart, online transactions are becoming more and more popular, as customers prefer ordering their favorite products online. In 2020, the fraudulent transactions have exceeded $32 billion (Borgne & Bontempi, 2021), and this amount is projected to grow even more in the following years. It is important that the online retailers can detect and prevent fraudulent transactions, so that the customers are not charged for fraudulent purchases. Furthermore, fraudulent transactions can cause online retailers significant revenue losses. The identification of fraudulent transactions has become a must for electronic payments providers. This paper proposes an ensemble method based on voting and deep learning models to detect fraudulent online transactions. The performance of proposed approaches will be measured with various evaluation metrics, such as accuracy rate, precision, recall and confusing matrix.

B. Dataset

The proposed dataset contains legitimate online transactions that occurred in two days. It is highly unbalanced with only 0.2 % of all transactions are labelled as fraudulent transactions. Due to confidentiality, the actual features and values are not provided, and were transformed with PCA. The only featured that have not been transformed are the time and transaction amounts. The other features are named V1...V28 and contain numerical values. The column “Class” contains the label value, meaning if the value “1” then it is a fraudulent transaction, otherwise it is “0”.

Columns	Data Type	Description
Time	Numeric	'Time' denotes the duration in seconds between each transaction and the initial transaction in the dataset.
V1-V28	Numeric	V1 to V28 have undergone transformation using Principal Component Analysis (PCA), and for security reasons, the attribute names are not specified.
Amount	Numeric	The monetary value of the transaction.
Class	Numeric	In binary classification, a value of '1' indicates the detection of fraud, while a value of '0' represents no fraud detected.

Table 1 – Dataset description.

C. Research Question

The purpose of this research project is to develop a generalized model that will be preventing fraudulent online transactions, this binary classification model can be implemented into online payment platforms that are used by online retailers, and ultimately help online retailers reduce losses caused by fraudulent activities. Furthermore, this paper will identify what factors contribute to the accuracy and effectiveness of fraud transactions classifier algorithms, and how those algorithms can be optimized to reduce the false positives and false negatives. To detect fraudulent activities, features will be constructed based on domain knowledges of industry experts and based on the best practices in Machine and Deep Learning, and Ensemble Hard Vote will be used for final classification.

LITERATURE REVIEW

Credit card fraud prevention has been a prominent research topic, with numerous studies focusing on developing effective strategies and techniques to mitigate this issue. The following literature review provides an overview of key findings and approaches in credit card fraud prevention.

Kim et al. compared the performance of two approaches using a massive real-world transaction dataset [1]. The authors utilized practical evaluation metrics and introduced the champion-challenger framework to develop and compare the models. Kim et al. adopted a deep learning-based approach to credit card fraud detection and conducted an in-depth comparison with the industry-standard hybrid ensemble method. The study described the processes of developing both the hybrid ensemble model and the deep learning model, using a highly imbalanced dataset with a massive volume of real-world transactions. Authors evaluated the post-launch performance after deployment. The deep learning-based model emerged as the winner and was implemented in the FDS of their partner organization. The research contributed by adopting a deep learning approach, addressing practical challenges, and conducting experiments under realistic settings.

With the increasing prevalence of e-commerce transactions, the need for effective fraud detection and prevention methods has become crucial. Abdul et al. examined the proposed Credit Card Fraud Detection and Prevention (CCFDP) method, which integrates various modern techniques to enhance accuracy and prevent fraudulent activities [2]. The CCFDP method incorporates several advanced techniques, including Random Under sampling (RU), t-distributed Stochastic Neighbor Embedding (t-SNE), Principal Component Analysis (PCA), Logistic Regression Learning (LR), and Singular Value Decomposition (SVD). These techniques contribute to quicker data training processes and increased accuracy in fraud detection. One major challenge in fraud detection is dealing with unbalanced datasets. The CCFDP approach utilizes Random under sampling to create a new balanced dataset. By randomly selecting legitimate transactions that match the size of fraudulent transactions, this method helps address the issue of imbalanced data, allowing for more authentic data to be used. The comparison is conducted based on fraud detection accuracy and fraud prevention accuracy. The results demonstrate that the CCFDP outperforms other methods in terms of accuracy and prevention capability.

Ajeet et al. proposes an efficient approach called CSWLB (Cost-Sensitive Weighted Bagging) for detecting and reducing fraudulent credit card transactions in imbalanced datasets [3]. The CSWLB approach combines cost-sensitive learning, cost functions, and two weak learners to improve the accuracy and cost-effectiveness of fraud detection. The proposed CSWLB approach consists of two phases. In the first phase, the Brazilian dataset is preprocessed using min-max data normalization to enhance the accuracy of CSWLB. Numeric features in the dataset are normalized on a scale of 0-1. In the second phase, high costs are assigned to fraudulent transactions through a cost-sensitive classifier. The Random Forest classifier is then applied to these weighted bags for classification, ultimately reducing prediction costs. The results showed significant improvements in various performance metrics compared to state-of-the-art techniques. The CSWLB approach achieves a high detection rate, precision, and classification accuracy, while reducing misclassification of transactions.

Nicholas et al. examines the use of Artificial Immune Systems (AIS) as an approach to online credit card fraud detection [4]. The reviewed paper emphasizes the relevance of AIS in addressing credit card fraud management. It suggests that the application of AIS, particularly through mechanisms like new transaction representation and variable width r-contiguous bit matching algorithms, vaccination processes, and memory cell evolution processes, has the potential to significantly improve detection performance. The example of distributing the AIS for credit card fraud detection across multiple servers, including database components and transaction processing subsystems, is presented. Additionally, the possibility of distributing detector generation based on specific criteria, such as credit card numbers or merchant IDs, is proposed.

Monika et al. discusses a framework called Deep Ensemble Algorithm (DEAL) for predicting credit card fraud (CCF) in real-time data streams [5]. The paper highlights the use of the Extra-Tree Ensemble technique and deep learning (DL) to improve categorical prediction accuracy and prevent model overfitting. The proposed framework addresses challenges such as imbalanced class distribution and overlain spending patterns in credit card transaction data. It also mentions the evaluation of the framework using performance metrics such as categorical accuracy, false classification rate (FCR), false positives (FP), and loss. The paper suggests the utilization of the CC paradigm, which combines data analysis and intelligence tasks to address computational and storage challenges.

Detecting credit card fraud is a critical task in ensuring secure financial transactions. Anurag et al. discusses the proposed Credit Card Fraud Detection and Prevention (CCFDP) method, which incorporates various modern techniques to enhance detection accuracy [6]. The method addresses the challenge of imbalanced datasets through the application of Random Under sampling (RU) to create a balanced dataset. The CCFDP combines several techniques to improve accuracy and fraud detection capabilities. The integration includes RU, t-SNE, PCA, Logistic Regression (LRL), and Singular Value Decomposition (SVD). RU helps address the issue of data imbalance by randomly selecting authentic data points to create a balanced dataset. PCA reduces dimensionality while retaining the data's important variation, and t-SNE further reduces dimensionality by separating similar and dissimilar instances.

Credit card fraud is a significant concern for banks due to the substantial financial losses it incurs each year. To combat this issue, rule-based systems have traditionally been employed, and more recently, researchers have proposed artificial intelligence (AI)-based approaches. The utilization of ensemble models and three voting mechanisms (OPT, PES, and WGT) for credit card fraud detection. Yigit et al. proposed model, named OPWEM (Optimistic, Pessimistic, and Weighted Voting in an Ensemble of Models), combines multiple popular supervised AI models, including Decision Trees (DT), Random Forests (RF), Bayesian Networks (BN), Naive Bayes (NB), Support Vector Machines (SVM), and K* models [7].

The performance of the ensemble model is assessed using the OPT, PES, and WGT voting mechanisms. The evaluation utilizes the real transaction dataset acquired from the Turkish bank. The results indicated promising outcomes for the proposed ensemble model. The OPT strategy detects 31.59% of fraudulent transactions with a false alarm rate of only 0.10%. On the other hand, the PES strategy achieves a fraud detection rate of 93.92% with a false alarm rate of 13.72%. The WGT strategy detects 64.02% of fraudulent transactions with a false alarm rate of 0.75%. Banks can choose among these voting mechanisms based on their preferred fraud detection strategies and desired false alarm rates. The practical application of the OPWEM model addresses the challenge of credit card fraud detection in collaboration with existing rule-based systems. The model makes decisions regarding the legitimacy or fraudulence of new transactions by employing the ensemble of models and the selected voting approach.

Logistic Regression, Random Forest, KNN, and Neural networks. The pioneering work by Ghosh et al. in 1994 introduced the application of neural networks for fraud detection [8]. They trained multiple neural networks using a substantial dataset of labeled credit card transactions,

which were then validated using account activities spanning a two-month period. The neural network models were trained using different types of fraud, such as lost or stolen cards, application fraud, counterfeit fraud, mail-order fraud, and NRI (non-received issue) fraud (Misra et al., 2020). In a similar vein, Brause et al. (1999) employed association rules mining and neural networks to minimize the false positive rate in their research. Over the past years, a variety of supervised and unsupervised machine learning and optimization algorithms have been employed to detect credit card fraud.

The detection of fraud primarily involves a binary classification problem that can be effectively addressed through supervised learning techniques. This method requires a dataset with labeled instances of "fraud" and "non-fraud," which is used to train a classifier. One of the advantages of supervised learning is that the algorithm's output, which pertains to class labels, is directly interpretable by humans. Also, supervised learning facilitates the identification of discriminative patterns (Abdallah et al., 2016). In contrast, unsupervised learning techniques are employed when dealing with unlabeled datasets. This approach examines fresh transactions and seeks out anomalous patterns within them. Both supervised and unsupervised learning techniques employ various methods, including classification algorithms such as artificial neural networks, K-nearest neighbors, decision trees, logistic regression, Naive-Bayes, and support vector machines (SVM) (Bolton et al., 2002).

Esenogho et al. introduced an effective approach for credit card fraud detection, employing a neural network ensemble classifier along with a hybrid data resampling technique [9]. Their method utilized a Long Short-Term Memory (LSTM) network as a base learner within the adaptive boosting (AdaBoost) framework. Additionally, the authors incorporated a hybrid resampling approach that combined the synthetic minority oversampling technique (SMOTE) with the edited nearest neighbor (ENN) method, known as SMOTE-ENN. The proposed approach holds significant value for two main reasons. Firstly, LSTM networks are known for their robustness in modeling sequential data, making them well-suited for capturing the temporal nature of credit card transactions. Secondly, the AdaBoost technique constructs strong classifiers that are less prone to overfitting, thereby reducing false-positive predictions. The combination of these components proved to be a highly effective approach for detecting credit card fraud, enhancing both accuracy and generalization capabilities.

Ileberi et al. investigated the utilization of the Synthetic Minority Over-sampling Technique (SMOTE) for oversampling in the context of credit card fraud detection [10]. They evaluated

several machine learning algorithms, including Support Vector Machine (SVM), Random Forest (RF), Extra Tree (ET), Extreme Gradient Boosting (XGBoost), Logistic Regression (LR), and Decision Tree (DT) individually. Furthermore, they paired the Adaptive Boosting (AdaBoost) algorithm with each of these methods to enhance their robustness. The findings of the study indicated that incorporating the AdaBoost algorithm had a positive impact on the performance of the evaluated machine learning methods. Moreover, the proposed framework was validated using a highly imbalanced synthetic credit card fraud dataset, and the results were highly promising. The achieved performance, as measured by various classification metrics, demonstrated an impressive accuracy rate of 97% across all evaluations. Makki et al. presented their solution in two phases and discussed their findings. In the first phase, they considered eight classification algorithms, namely C5.0, SVM, and ANN, to determine the most suitable ones for their task [11]. In the second phase, the selected algorithms were utilized to compare various imbalance classification approaches, including Random Oversampling, One-Class Classification, and Cost Sensitive techniques. The results of their study indicated that all the algorithms achieved accuracies higher than 90%, albeit with varying sensitivity and AUPRC (Area Under Precision-Recall Curve) values. However, based on the evaluation of three performance measures, namely Accuracy, Sensitivity, and AUPRC, the researchers concluded that LR (Logistic Regression), C5.0 decision tree algorithm, SVM, and ANN outperformed the other methods, making them the best approaches for their intended purpose.

Tingfei et al. conducted a study focused on credit card fraud detection using three distinct oversampling models: SMOTE, GAN, and VAE. Additionally, their research employed a deep learning method as the baseline, rather than a traditional machine learning approach. The study revealed that the application of different oversampling methods to increase the number of positive cases had varying degrees of impact on the classifier's performance, specifically, the recall rate, which measures the ability to correctly identify positive cases, increased by 0.02 for the SMOTE method and 0.03 for the GAN method [12]. This resulted in recall rates of 0.85 and 0.86, respectively, representing a significant improvement compared to the baseline recall rate of 0.83. These findings suggest that both SMOTE and GAN oversampling techniques offer valuable enhancements in detecting credit card fraud, demonstrating their effectiveness in improving the classifier's ability to identify fraudulent transactions.

Carcillo et al. employed a hybrid technique in their study to enhance the feature set of the fraud detection classifier by incorporating unsupervised outlier scores [13]. A notable aspect of their

research was the implementation and evaluation of various levels of granularity in defining these outlier scores. The researchers' significant contribution lay in demonstrating the effectiveness of their recommended strategy, which led to improved detection accuracy in fraud detection. By leveraging unsupervised outlier scores and exploring different levels of granularity, Carcillo et al. showcased the potential for enhancing the classifier's performance in identifying fraudulent activities. Their findings thus highlight the value of incorporating unsupervised techniques for improved fraud detection outcomes.

In their work, Carcillo et al. introduced the SCAlable Real-time Fraud Finder (SCARFF), a machine learning method specifically designed to tackle challenges such as nonstationary, imbalance, and feedback latency in the context of fraud detection [14]. To address these issues, the researchers incorporated big-data tools like Cassandra, Kafka, and Spark into their machine learning approach. Through experiments conducted on a large-scale dataset of real credit card transactions, Carcillo et al. demonstrated that the SCARFF system is not only effective and accurate but also scalable. By leveraging the power of big-data tools, their method successfully handles the complexities of real-time fraud detection, ensuring efficient processing, accurate predictions, and the ability to adapt to changing patterns in a dynamic environment. The findings of their study highlight the practical value of integrating big-data tools into the machine learning pipeline for robust and scalable fraud detection systems.

EXPLORATORY DATA ANALYSIS

During the initial analysis stages, great attention was given to reviewing the properties and characteristics of the attributes. The primary focus at this stage was to explore the distribution of variables, identify correlation dependencies, and construct data-driven narratives. To facilitate this process, the data analysis was divided into three key criteria, which will be described in the following subsections.

Data Summary

This dataset encompasses credit card transactions conducted by European cardholders during September 2013. It comprises a total of 274,807 rows and 32 columns. The predominant portion of these columns consists of numerical features, categorized as 'Number' while the remaining columns are of 'Text' type. All features, except for 'DateTime,' exhibit instances of missing data. Notably, this dataset exclusively comprises numerical input variables that result from a PCA transformation. Regrettably, due to confidentiality constraints, the original features are withheld, along with additional contextual details about the data. The primary components derived from PCA are denoted as Features V1 through V28, whereas 'Time' and 'Amount' are the sole attributes untouched by PCA transformation. 'Time' denotes the elapsed seconds between each transaction and the initial transaction in the dataset, while 'Amount' corresponds to the transaction value. This 'Amount' feature holds potential for applications like example-dependent cost-sensitive learning. Lastly, the 'Class' feature serves as the response variable, assuming a value of 1 in cases of fraud and 0 otherwise.

Data Cleaning

In the process of preparing and refining the data for my project, several crucial data cleaning steps were undertaken. Initially, I addressed data integrity by meticulously removing duplicate entries, ensuring the dataset's accuracy. To enhance consistency and comparability, I standardized the 'Time' and 'Amount' columns, aligning their values for more effective analysis. Given the underrepresentation of Class 1 instances, I employed the Synthetic Minority Over-sampling Technique (SMOTE) to rebalance the dataset, thereby mitigating potential biases. This strategic random sampling bolstered the dataset's overall robustness. To assess and validate the performance of the developed model, I further partitioned the dataset into an 80% training subset

and a 20% test subset, facilitating a comprehensive evaluation of the model's predictive capabilities and ensuring its generalizability to real-world scenarios.

Univariate Analysis

Having performed checks for null values and data types, it was found that 1081 duplicate observations existed in the dataset. These duplicates were subsequently removed. Upon examining the count plot of the target variable, 'Class', a significant data imbalance was identified. The instances of credit card fraud were so minimal that they were barely visible. Specifically, there were 284,315 instances of non-credit card fraud and only 492 instances of credit card fraud (see Figure 1). This represents less than 1% of the entire dataset. To address this issue, oversampling techniques will be employed to balance the data (SMOTE).

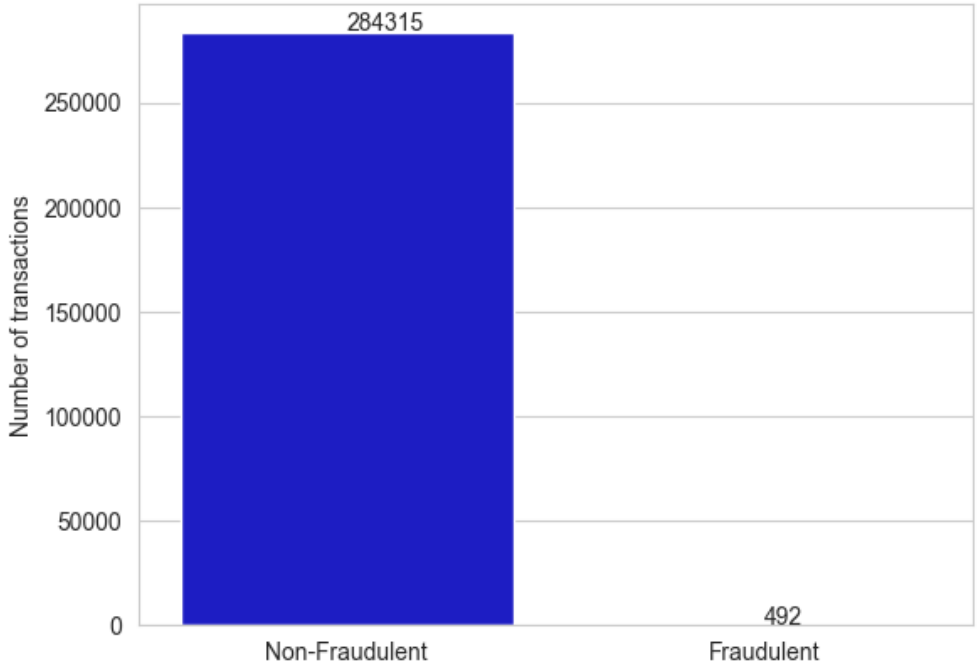


Figure 1 – Distribution of classes.

I placed significant emphasis on outlier detection. An outlier refers to a value within a random sample from a population that significantly deviates from the other values. To identify such outliers in our dataset, I employed the Box Plot technique specifically for the 'Amount' feature (see Figure 2 and 3). In this method, any data point located above or below the whiskers is considered an outlier. Given that 'Amount' is the sole numeric feature with meaningful interpretation.

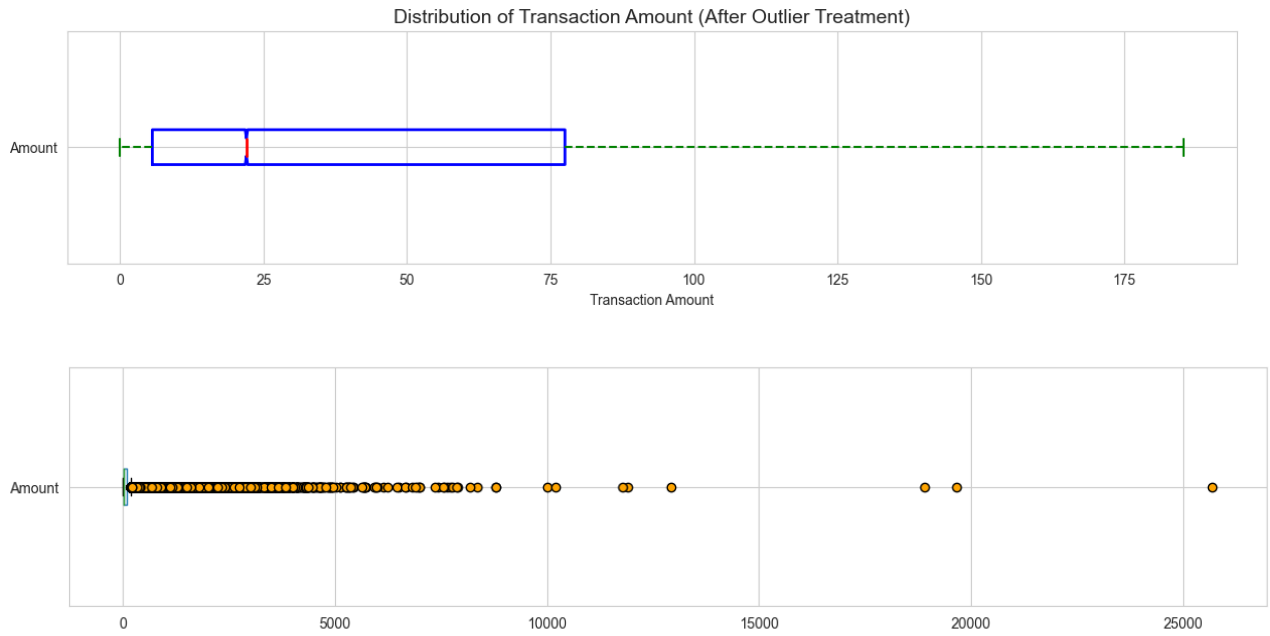


Figure 2- Anomaly detection via boxplots.

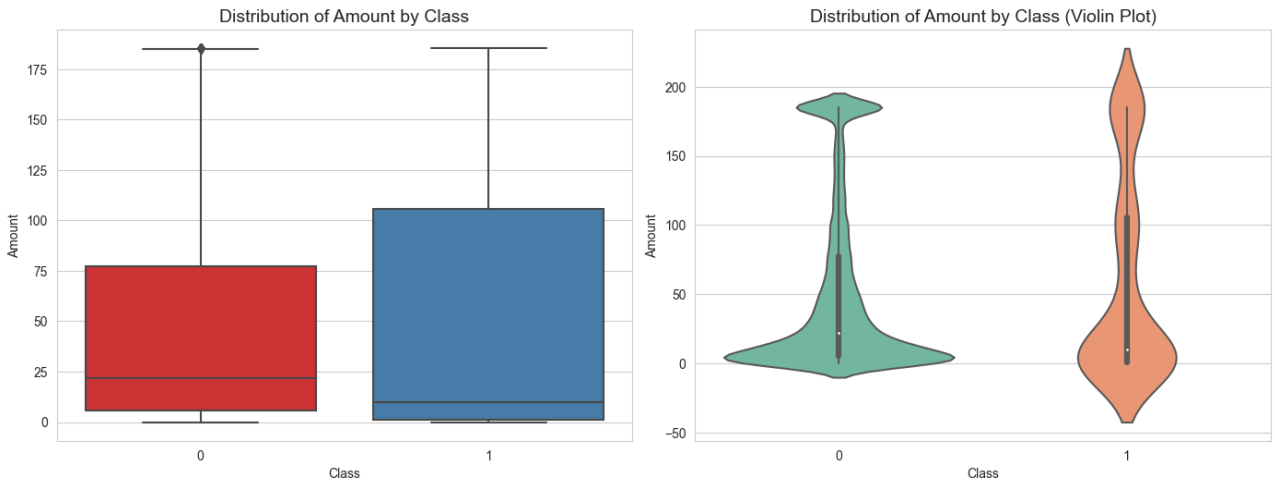


Figure 3 – Distribution of amounts per class.

The visualization of the 'Amount' values revealed a clear concentration towards smaller amounts, with only a few instances of larger values. Failing to address this issue could significantly bias the detection model during prediction. To mitigate the impact of outliers, I have applied Median Imputation after identifying them. This approach involves replacing extreme values with the median value. The interquartile range (IQR), calculated as the difference between the third quartile (Q3) and the first quartile (Q1), serves as a useful metric for outlier detection.

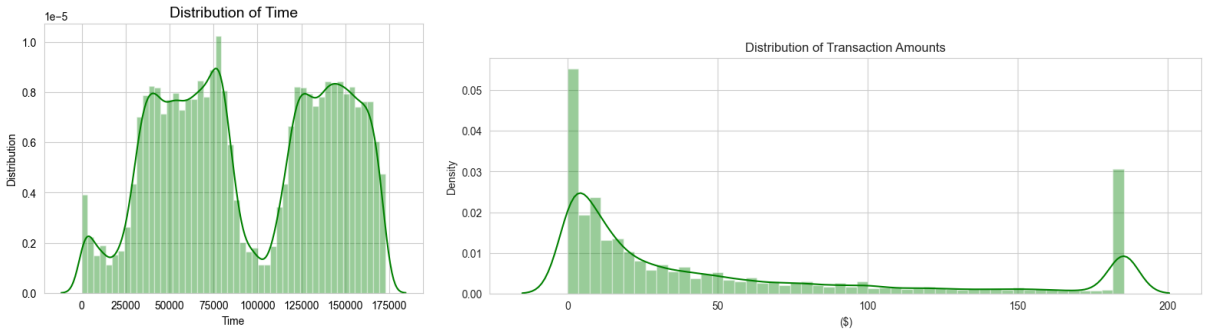


Figure 4 – Amount and time values distribution plot

Upon analyzing the distribution of 'Time' and 'Amount', (see Figure 4) no significant observations or patterns were identified. However, it is evident that there is a high concentration of transactions with very small amounts close to zero. Additionally, there is a higher density of transactions during daytime.

Bivariate Analysis

To assess the correlation between two discrete dimensions encompassing all 32 attributes, I have generated a correlation heatmap (see Figure 5). The heatmap visualizes the 2D correlation matrix, ranging from a positive correlation of '+1' to a negative correlation around '-0.5'. Positive correlations are depicted in green, while negative correlations are represented in golden color. The intensity of the color reflects the magnitude of the correlation, with stronger colors indicating larger magnitudes. The corresponding correlation values are also displayed. Upon analyzing the heatmap, it was observed that columns V1 to V28 exhibit no significant correlation with each other. However, since these attributes lack meaningful names for security purposes, their descriptive statistics do not hold valuable interpretations. Correlations include Time/V3 = -0.42, Amount/V2 = -0.53, Amount/V5 = -0.39, Amount/V7 = 0.4, and Amount/V20 = 0.34. Other minor correlations ranging from -0.3 to 0.3 exist but are not considered significant.

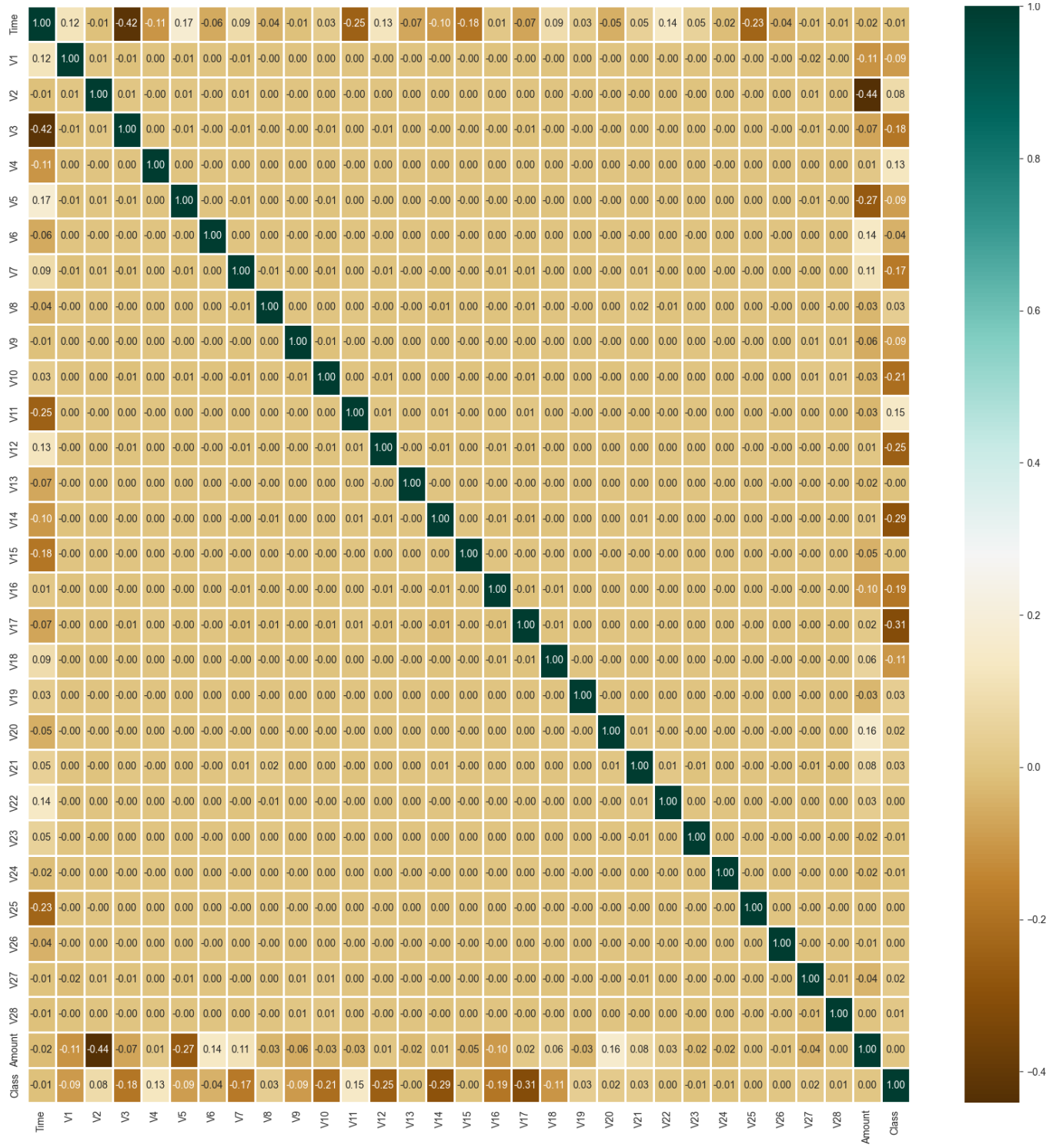


Figure 5 - Heatmap of dataset features.

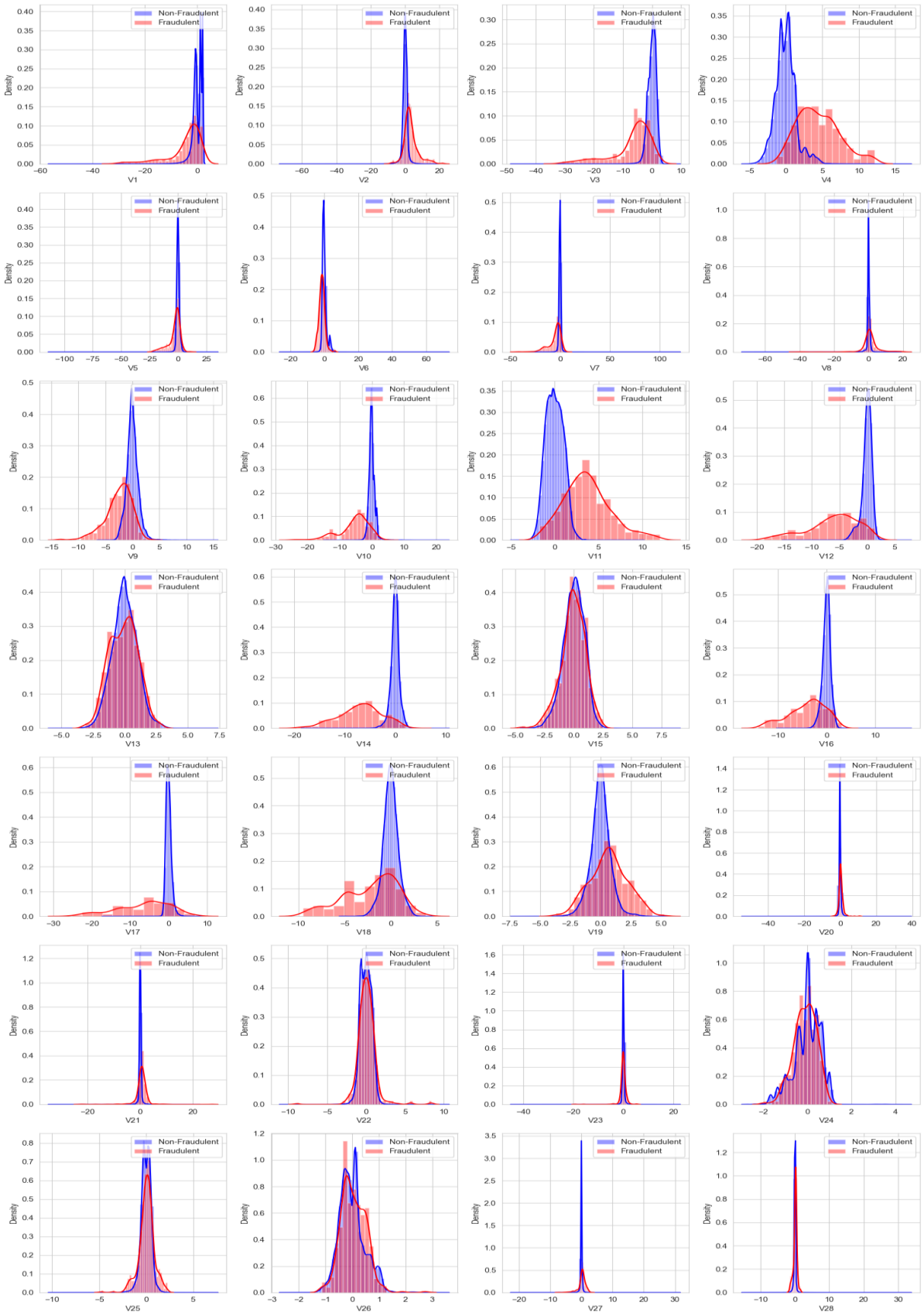


Figure 6 – Correlations between features V1 to V28.

'Amount/V5 = -0.39' indicates a negative correlation, while 'Amount/V7 = 0.4' and 'Amount/V20 = 0.34' show notable positive correlations. Additionally, there are other minor correlations ranging from -0.3 to 0.3; however, they are considered less significant. Based on the analysis, we can conclude that the variables 'Time' and 'Amount' hold the most importance within the dataset. The distributions for both classes exhibit a Gaussian bell curve shape. Notably, for the 'Fraudulent' class, features V3, V9, V10, V12, V14, V16, V17, and V18 tend to have a higher probability of negative or lower values compared to the other class. Conversely, features V4 and V11 display an opposite pattern. It is worth mentioning that the findings would be more valuable if the features had proper titles or descriptions.

Furthermore, I plotted the distribution of 'Amount' over the two classes to examine patterns in genuine and fraudulent transaction amounts (see Figure 7). The graph clearly demonstrates that fraud transactions predominantly occur with very small amounts. This may be because fraudsters aim to go unnoticed by both the account holder and financial institutions. Generally, people tend to disregard very small transactions in their accounts, assuming they are bank charges, account maintenance fees, or interest deductions. This behavior is exploited by fraudsters as part of their forgery tactics. Upon further analysis, it was observed that transactions for both the genuine and fraud classes exhibit similar patterns. This implies that predicting the class based solely on the amount value is challenging. Additionally, the amount values have been previously modified through anomaly reduction techniques.

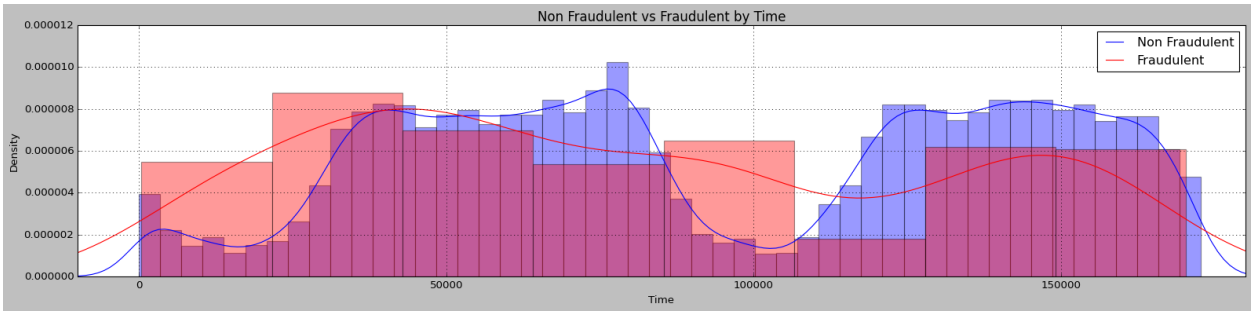


Figure 7 – Amount distributions based on label values.

Multivariate Analysis

To gain a better understanding of the current class imbalance, a scatter graph has been plotted to visualize the class instances alongside the genuine instances. In the graph, the minority class instances are plotted with five times more weight to appear larger.

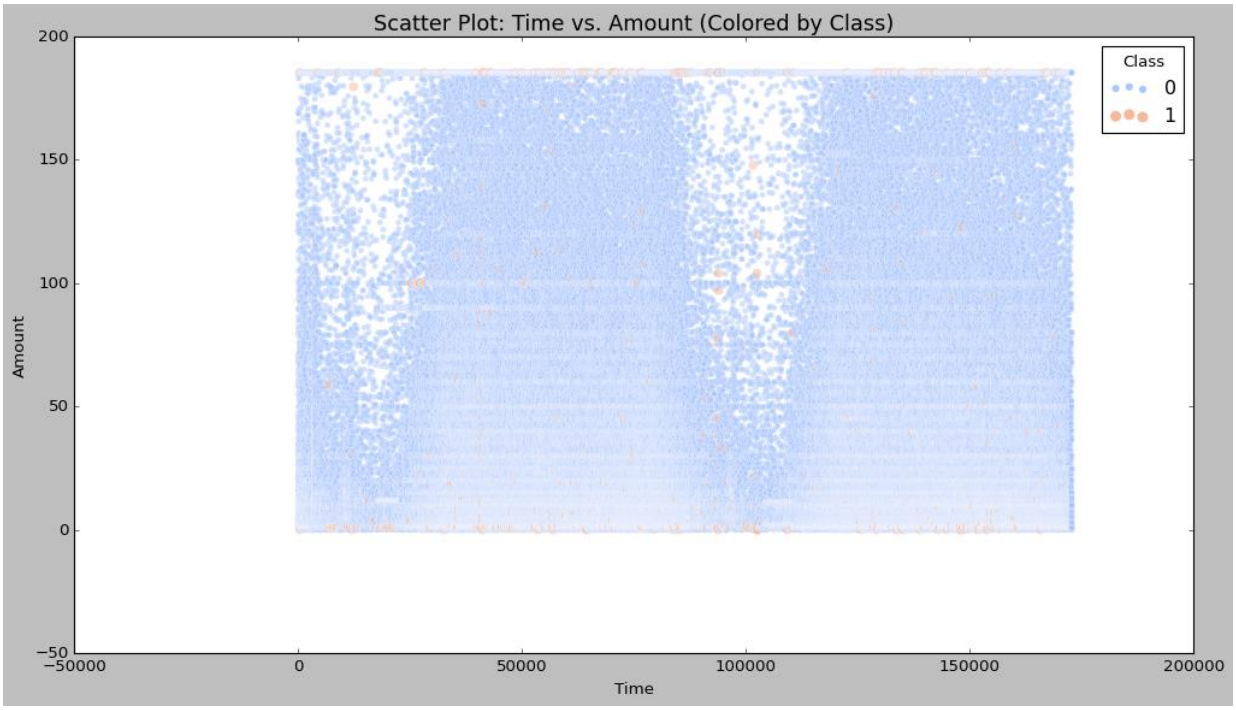


Figure 8 – Amount and time scatter plot per class.

The scatter graph reveals that many fraud transactions occur with small amounts (see Figure 8). There is also a density of fraud transactions around the amount value of $0-$20. Additionally, it can be observed that the density of fraud transactions is higher during specific time slots. When the number of normal transactions is higher, the number of fraud transactions also tends to be higher. These characteristics provide insights into the patterns associated with fraud transactions, emphasizing the importance of amount and time variables in detecting fraudulent activities.

METHODOLOGY AND EXPERIMENTS

Aim of study

The aim of this study is to train a model for credit card fraud detection using machine learning and deep learning and design and develop an intelligent system that leverages advanced algorithms to accurately identify and prevent fraudulent credit card transactions.

Experimental Design

1. Processing Data

To address the issue of the highly imbalanced dataset, I will employ an oversampling technique to balance the classes Synthetic Minority Oversampling Technique (SMOTE). SMOTE generates synthetic instances of the minority class by interpolating between existing instances, thereby increasing the representation of the minority class in the dataset. SMOTE consists of these steps:

1. Determine the feature vector's closest neighbour from the minority class instances.

2. Calculate the distance between the two sample points.

3. Multiply the calculated distance by a random number between 0 and 1.

4. At the computed distance along the line connecting the two points, find a new synthetic point.

5. Repeat the process for other identified feature vectors, creating additional synthetic samples.

By following these steps, SMOTE effectively creates synthetic data points that are similar to the minority class instances, expanding the representation of the minority class in the dataset.

2. Cross-validation

I will employ cross-validation for training and validation purposes. Specifically, I will utilize K-Fold cross-validation with 5 folds. The data will be divided into 5 folds, and iteratively, I will train the model using 4 folds (n-1) while validating it on the remaining 5th fold. The test set will be obtained by averaging the model predictions.

3. Splitting data for validation

I will reserve 10000 samples specifically for the test data. The remaining data will be split into a 4:1 ratio, with 80% allocated for training and the remaining 20% allocated for validation.

4. Parameters tuning using sklearn

Optimizing model parameters is crucial for maximizing model performance. For instance, by finetuning parameters such as max_depth, criterion, max_features, min_sample_leaf, and n_estimators in a classification model like Random Forest, significant improvements in precision can be achieved. To accomplish this, I will utilize sklearn package to conduct grid search and randomized search for parameter tuning. This approach will enable me to systematically explore different parameter combinations and select the optimal configuration for enhanced model performance.

Methodology

I will be using 5 ML models for classification on the test dataset, such as Logistic Regression, SVC, Decision Tree, Random Forest and KNN.
In addition to that, I will create five Neural Networks which will be used individually for prediction. I will construct a four-layer Artificial Neural Network (ANN) architecture. The hidden layers of the network consist of 6, 20, and 10 units respectively (see Figure 9).

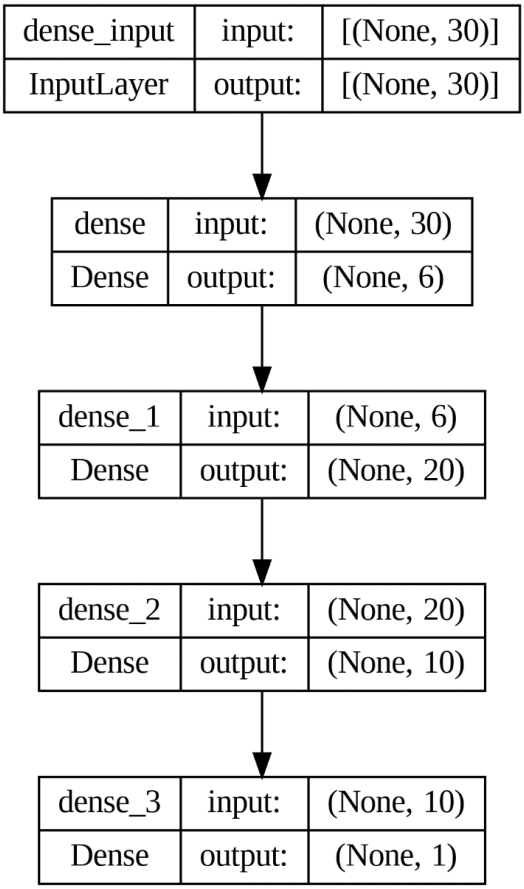


Figure 9 – Artificial Neural Network architecture.

Since we are dealing with a binary classification problem, the output layer contains a single node. For the activation function, I will employ the rectified linear unit (ReLU) for all the hidden layers. At the output layer, I will be using the sigmoid activation function. The sigmoid function is commonly used for binary classification tasks as it maps the output to a probability value between 0 and 1. For training the network, I have chosen the binary cross-entropy loss function. Binary cross-entropy is suitable for binary classification problems, as it measures the dissimilarity between the predicted probabilities and the true labels. For the Recurrent Neural Network (RNN) architecture, I will implement a three-layer structure consisting of two hidden layers and one output layer. The hidden layers contain 32 and 8 units respectively (see Figure 10), and the rectified linear unit (ReLU) activation function is applied in both. ReLU introduces non-linearity to the network and helps in learning complex temporal patterns.

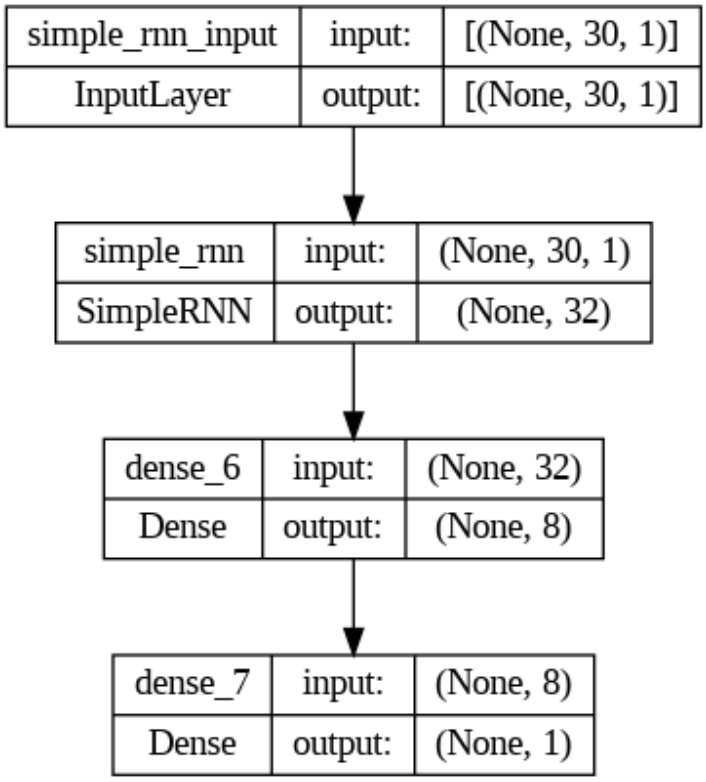


Figure 10 – Recurrent Neural Network architecture.

Regarding the loss function, I will be using the same binary cross-entropy as in the artificial neural network (ANN). The loss function penalizes the deviations of the predicted probabilities from the actual values, indicating how close or far the predicted values are from the true values. For optimizing the parameters of the RNN, I will employ the RMSprop optimizer. RMSprop is an optimization algorithm that adapts the learning rate based on the gradients of the parameters,

helping in efficient training of the network. I aim to leverage the sequential nature of the data and capture temporal dependencies, enabling effective modelling and prediction in tasks where the order and timing of events matter.

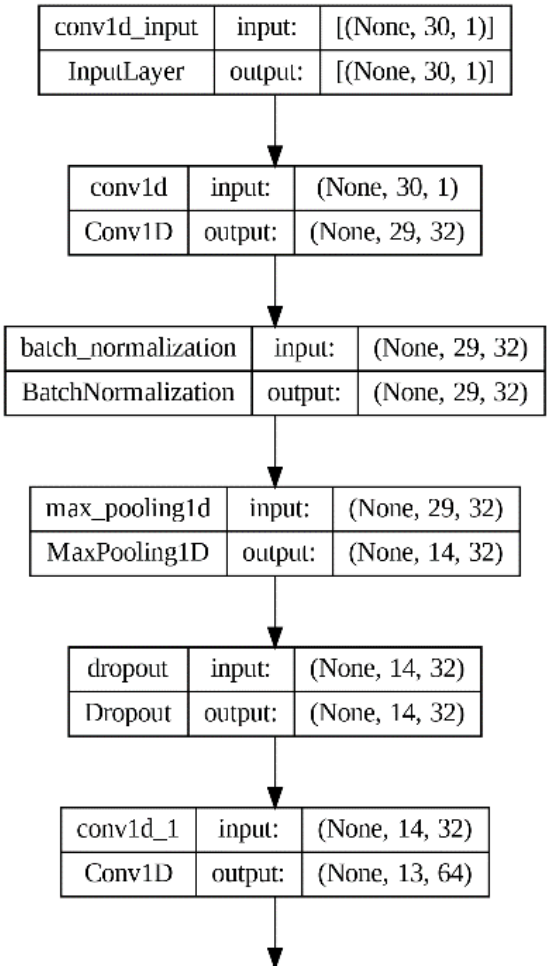
Binary cross entropy:

$$-\frac{1}{N}\sum_{i=1}^{N}(\log(p_i))$$

The Log loss formula:

$$\text{Log loss} = \frac{1}{N}\sum_{i=1}^{N} -(y_i * \log(p_i) + (1-y_i) * \log(1-p_i))$$

During the training of a Convolutional Neural Network (CNN), I will manipulate the training and testing data in a three-dimensional format. In the initial convolution layer, I will utilize 32 filters with a kernel size of 2 and apply the 'RELu' activation function to the first rows of the input shape.



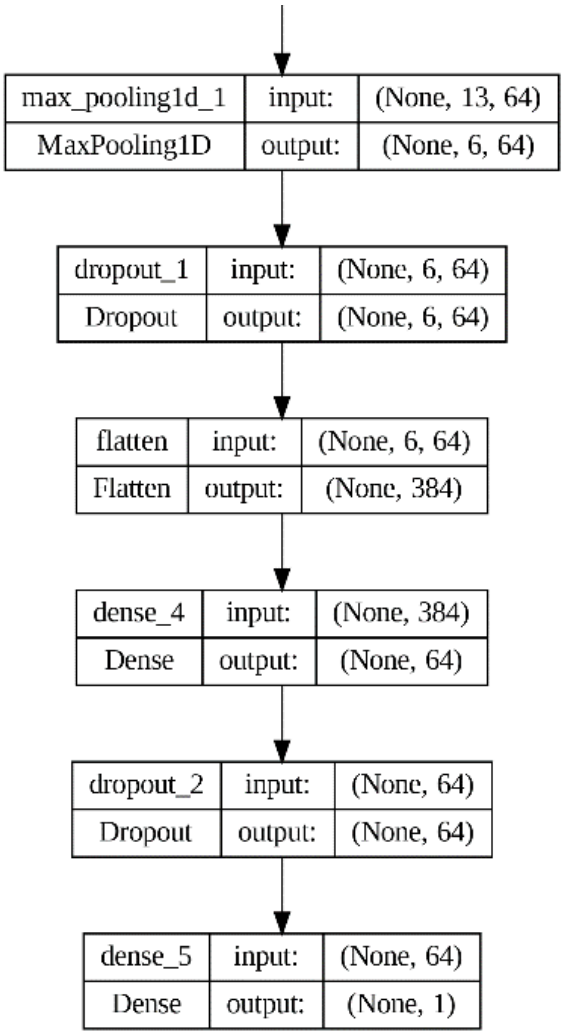


Figure 11 – Convolutional Neural Network architecture.

Additionally, I will incorporate batch normalization, max pooling with a window size of 1, and a dropout ratio of 0.2. Batch normalization is a technique that helps maintain the mean output close to 0 and the standard deviation of the output close to 1. Max pooling is employed to down sample the input representation by selecting the maximum value within a defined spatial window. The Dropout layer is used to mitigate overfitting by randomly setting input units to 0 at a specified rate during training. Inputs that are not set to 0 are scaled up by a factor of 1/(1 - rate) to ensure the total sum remains the same. In this specific case, I randomly drop 20% of the neurons.

I have constructed a deep learning model consisting of four layers using LSTM (Long Short-Term Memory) architecture. The first two layers are LSTM layers, while the remaining two

layers are dense layers. In the LSTM layers, we have utilized 100 and 50 units, respectively, and applied the 'RELu' activation function to both layers.

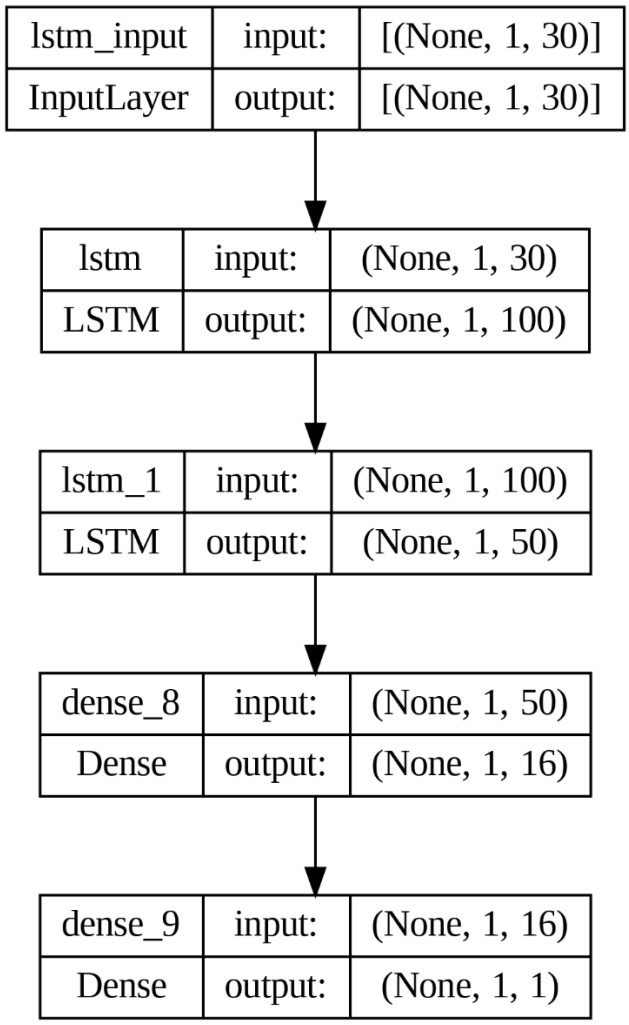


Figure 12 – Long Short Term Memory architecture.

The GRU (Gated Recurrent Unit) model shares the same set of hyperparameters as the LSTM model (see Figure 12, 13). The final hidden dense layer comprises 16 units and employs the 'RELu' activation function. Binary cross-entropy will be used as the loss function during compilation. This loss function is commonly used for binary classification tasks, where the model is trained to predict one of two classes. Rmsprop optimizer is used as the optimizer during compilation. RMSprop is an optimization algorithm that adapts the learning rate based on the gradients of the parameters. It is often used for recurrent neural networks (RNNs) such as LSTM and GRU.

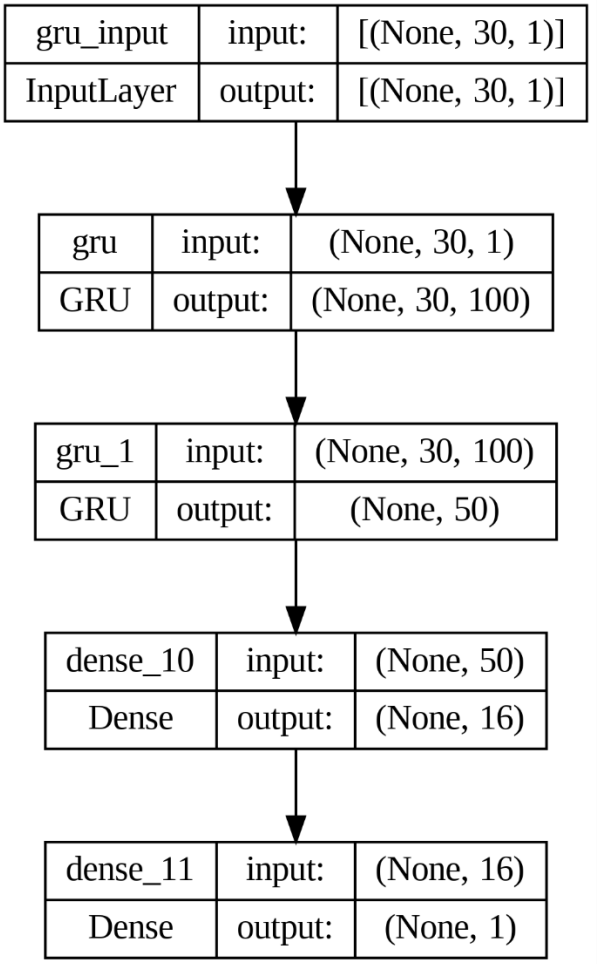


Figure 13 – Gated Recurrent Unit.

Credit card fraud detection is a binary classification problem, and one commonly used method to evaluate model performance is the confusion matrix. This specific table layout allows us to visualize the algorithm's performance. During prediction, four cases can occur:

1. True Positive (TP): The model correctly predicts the true label. In the context of fraud detection, this refers to instances where non-fraudulent transactions are predicted correctly.
2. True Negative (TN): The model correctly predicts the false label. In our case, this corresponds to correctly predicting fraudulent transactions.
3. False Positive (FP): The model incorrectly predicts the true label. In fraud detection, this occurs when the model predicts a genuine transaction, but it is actually a fraudulent transaction. This is an area that requires special attention.
4. False Negative (FN): The model incorrectly predicts the false label. In our scenario, this happens when the model predicts a fraudulent transaction as genuine. This is another critical aspect that needs to be addressed.

Since, I will be deploying five machine learning algorithms, namely logistic regression, support vector machine, decision tree, random forest, and k-nearest neighbors, as well as five deep learning algorithms, including ANN, CNN, RNN, LSTM, and GRU, for each of these algorithms, I will obtain corresponding confusion matrices to assess their performance in credit card fraud detection.

Furthermore, I will utilize the following performance metrics to evaluate the classification of credit card fraud detection:

- **Accuracy**: Measures the overall correctness of the model's predictions.
- **Precision**: Assesses the proportion of correctly predicted positive instances out of all instances predicted as positive. It focuses on the accuracy of fraud detection specifically.
- **Recall** (also known as sensitivity or true positive rate): Evaluates the proportion of correctly predicted positive instances out of all actual positive instances. It highlights the model's ability to identify fraud cases.
- **F1 Score**: Represents the harmonic mean of precision and recall. It provides a balanced measure of the model's performance, taking into account both precision and recall.
- **Area Under the ROC Curve (AUC-ROC)**: Quantifies the model's ability to distinguish between positive and negative instances by plotting the true positive rate against the false positive rate. It provides an overall performance measure for binary classification tasks.

The proposed ensemble hard vote classifier model framework can be visually represented as the diagram below, providing an overview of the system's structure and components (see Figure 14).

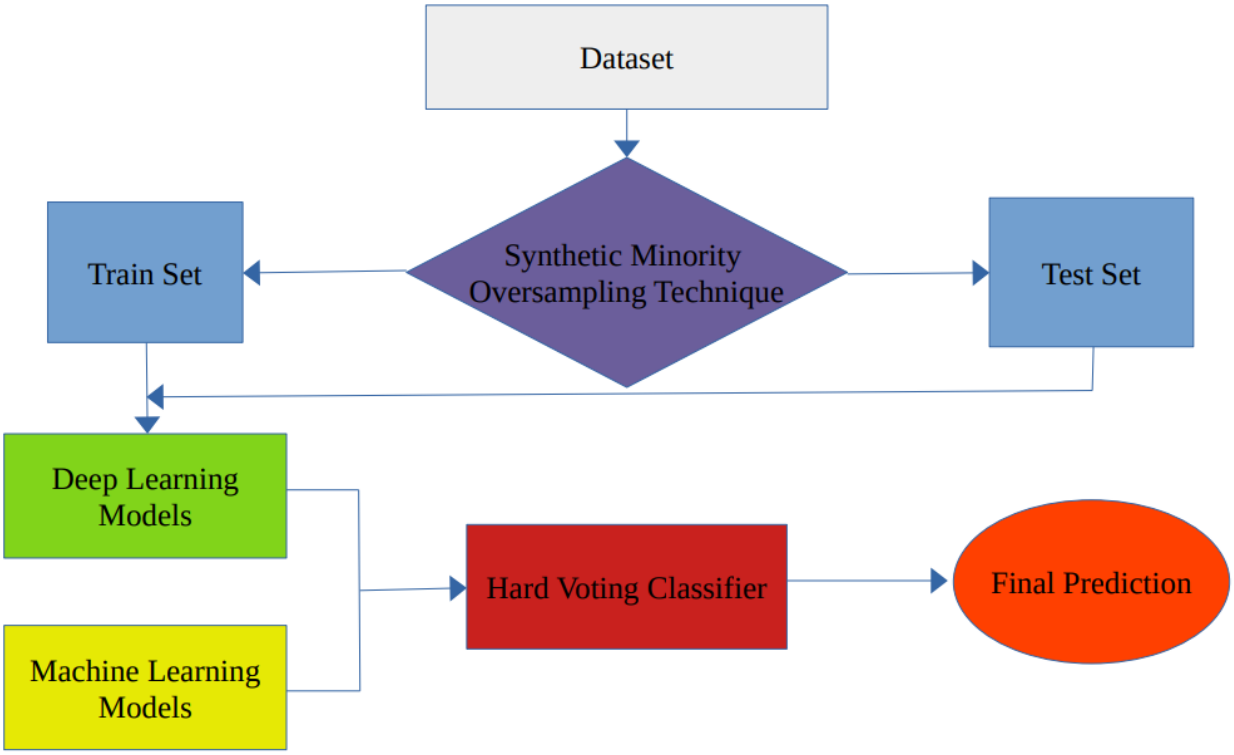


Figure 14 - The architecture of the proposed ensemble hard vote classifier.

RESULTS AND DISCUSSION

Credit card fraud detection involves binary classification. The most popular method for evaluating model performance is by using a confusion matrix, which presents a specific table layout to visualize algorithm performance. During prediction, four possible cases can occur. After applying logistic regression, support vector machine, decision tree, random forest, and k-nearest neighbor initially, I obtained the following performance reports (see Figure 15 – 19):

```
Model Name: LogisticRegression
MODEL ACCURACY SCORE: 0.96
Training Score: 0.96
Confusion Matrix:
[[1993   53]
 [ 113 1841]]
Accuracy Decision: 95.85
Error Rate Decision: 4.15
Specificity Decision: 97.20
Sensitivity Decision: 94.63
CLASSIFICATION REPORT:
              precision    recall  f1-score   support

           0       0.95      0.97      0.96      2046
           1       0.97      0.94      0.96      1954

    accuracy                           0.96      4000
   macro avg       0.96      0.96      0.96      4000
weighted avg       0.96      0.96      0.96      4000


------------------------------------------------------------
```

Figure 15 – Logistic Regression performance report.

```
Model Name: SVC
MODEL ACCURACY SCORE: 0.97
Training Score: 0.97
Confusion Matrix:
[[2012   34]
 [ 100 1854]]
Accuracy Decision: 96.65
Error Rate Decision: 3.35
Specificity Decision: 98.20
Sensitivity Decision: 95.27
CLASSIFICATION REPORT:
              precision    recall  f1-score   support

           0       0.95      0.98      0.97      2046
           1       0.98      0.95      0.97      1954

    accuracy                           0.97      4000
   macro avg       0.97      0.97      0.97      4000
weighted avg       0.97      0.97      0.97      4000
```

Figure 16 – Support Vector Classification performance report.

```
Model Name: DecisionTreeClassifier
MODEL ACCURACY SCORE: 0.96
Training Score: 0.96
Confusion Matrix:
[[1989   57]
 [ 102 1852]]
Accuracy Decision: 96.03
Error Rate Decision: 3.98
Specificity Decision: 97.01
Sensitivity Decision: 95.12
CLASSIFICATION REPORT:
              precision    recall  f1-score   support

           0       0.95      0.97      0.96      2046
           1       0.97      0.95      0.96      1954

    accuracy                           0.96      4000
   macro avg       0.96      0.96      0.96      4000
weighted avg       0.96      0.96      0.96      4000
```

Figure 17 – Decision Tree performance report.

```
Model Name: RandomForestClassifier
MODEL ACCURACY SCORE: 1.00
Training Score: 1.00
Confusion Matrix:
[[2043    3]
 [   5 1949]]
Accuracy Decision: 99.80
Error Rate Decision: 0.20
Specificity Decision: 99.85
Sensitivity Decision: 99.76
CLASSIFICATION REPORT:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2046
           1       1.00      1.00      1.00      1954

    accuracy                           1.00      4000
   macro avg       1.00      1.00      1.00      4000
weighted avg       1.00      1.00      1.00      4000
```

Figure 18 – Random Forest performance report.

```
Model Name: KNeighborsClassifier
MODEL ACCURACY SCORE: 0.99
Training Score: 1.00
Confusion Matrix:
[[2025   21]
 [   1 1953]]
Accuracy Decision: 99.45
Error Rate Decision: 0.55
Specificity Decision: 98.94
Sensitivity Decision: 99.95
CLASSIFICATION REPORT:
              precision    recall  f1-score   support

           0       1.00      0.99      0.99      2046
           1       0.99      1.00      0.99      1954

    accuracy                           0.99      4000
   macro avg       0.99      0.99      0.99      4000
weighted avg       0.99      0.99      0.99      4000
```

Figure 19 – K Neighbour performance report.

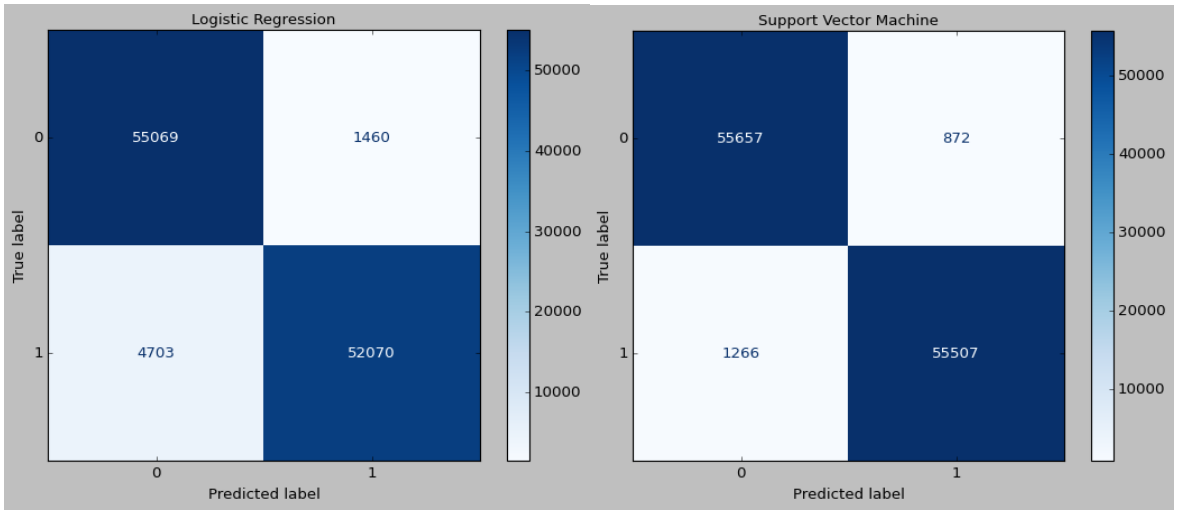


Figure 20 – The confusion matrices for Logistic Regression and SVC models.

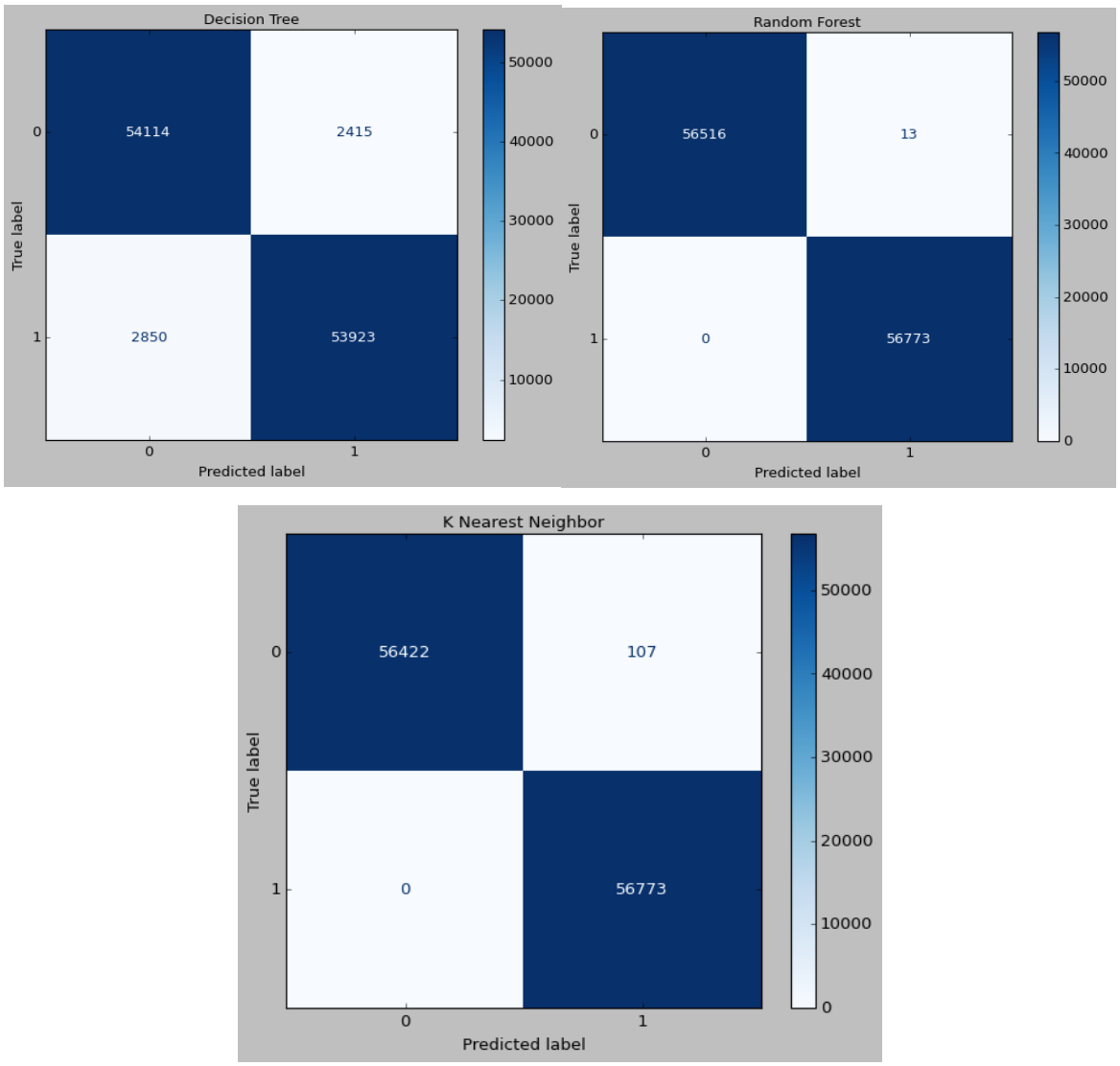


Figure 21 - The confusion matrices for the Decision Tree, Random Forest and KNN models.

Random Forest achieved the highest accuracy of 0.999, indicating excellent performance in credit card fraud detection. Additionally, K-nearest neighbors also produced positive results, although the accuracy value is not explicitly mentioned (see Table 1).

Model	Precision	Recall/Sensitivity	F1 Score	Accuracy
Logistic Regression	0.945	0.945	0.945	0.98
Support Vector Classification	0.98	0.98	0.98	0.98
Decision Tree	0.95	0.95	0.95	0.95
Random Forest	1	1	1	1

KNN	1	1	1	0.998

Table 2 – Summary of Machine Learning performance.

Having deployed various deep learning algorithms for classification: Artificial Neural Network, Convolution Neural Network, Recurrent Neural Network, LSTM, and GRU. Additionally, I implemented a hard vote classifier, which determined the majority outcome for classification. To ensure fairness, all neural networks were predicted simultaneously using the following numbers of epochs 50, 60, 70, 80, 90, 100. I have conducted multiple runs to evaluate and observe any discrepancies (see Table 2). I generated a heatmap of the confusion matrix and accuracy and loss curves to visualize the results comprehensively (see Figure 22-27).

Models	**ANN**			**CNN**			**RNN**			**LSTM**			**GRU**		
Epochs	FN	FP	TM	FN	FP	TM	FN	FP	TM	FN	FP	TM	FN	FP	TM
50	208	1926	2135	0	628	628	3760	2021	5781	87	211	298	87	211	298
60	87	1524	1611	0	653	653	1353	898	2251	129	242	371	129	242	371
70	589	1181	1770	7	619	626	20348	1608	21956	0	233	223	0	223	223
80	304	1323	1627	0	500	500	1012	3276	4288	89	174	263	89	174	263

Table 3 - Summary Table of Misclassification Rates for Various Neural Networks

Among the networks, CNN demonstrated the best overall effectiveness in detecting false negatives. Interestingly, LSTM and GRU exhibited similar performances. However, the RNN model's performance was the most inconsistent. Surprisingly, it was found that the performance of each model was not significantly dependent on the difference in epochs over this period.
While the Random Forest classifier achieved the highest accuracy, it is essential to recognize that performance evaluation should not solely rely on accuracy or other computation figures. Time complexity also plays a crucial role in assessing a model's suitability. In this context, time complexity becomes a significant factor. It's important to strike a balance between accuracy and time complexity when selecting the most appropriate model for a particular task, as both factors can significantly impact the practicality and usability of the solution.
I observed accuracy and loss curves for all neural networks (see Figure 22 - 27). The horizontal axis represents the epoch range, while the vertical axis displays accuracy for all networks in the first column, and loss values are represented in the second column. The random forest classifier with a maximum depth of '7' achieved the highest accuracy, performance assessment should not solely rely on accuracy or other computation metrics.

Accuracy Curve: A value close to '1' indicates that the model is performing well in terms of classification accuracy.

Loss Curve: Values close to '0' suggest that the model is functioning well, as lower loss signifies a better fit to the data.

Here's a summary of the findings for each network:

ANN: The accuracy curve shows an organic growth over epochs, and the loss curve consistently decreases with slight oscillations for both the training and validation sets (see Figure 22).

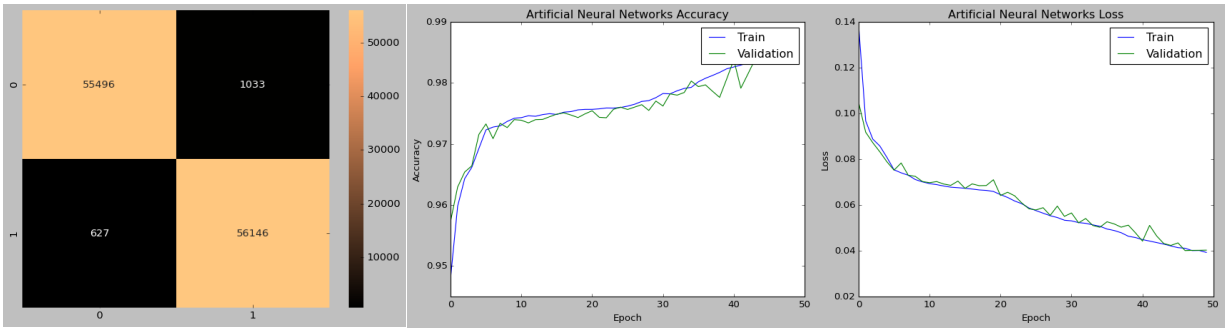


Figure 22 – Confusion matrix, accuracy, and loss of Artificial Neural Network.

CNN: The accuracy curve exhibits a similar pattern to ANN, but with a notable advantage – the model achieves higher accuracy on the validation set compared to the training set. Conversely, the loss curve shows better performance on the training set compared to the validation set. There are some fluctuations in the validation accuracy and loss (see Figure 23).

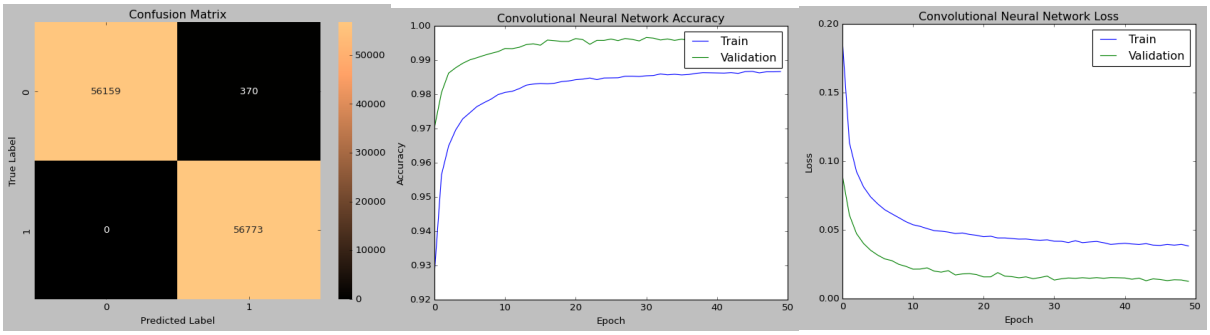


Figure 23 – Confusion matrix, accuracy, and loss curve of Convolutional Neural Network.

RNN: The accuracy and loss curves show similar swings in nature. However, there are inconsistencies in the oscillations of validation accuracy and loss compared to the training set (see Figure 24). These inconsistencies indicate that RNN is not operating well in the current implementation, as previously acknowledged.

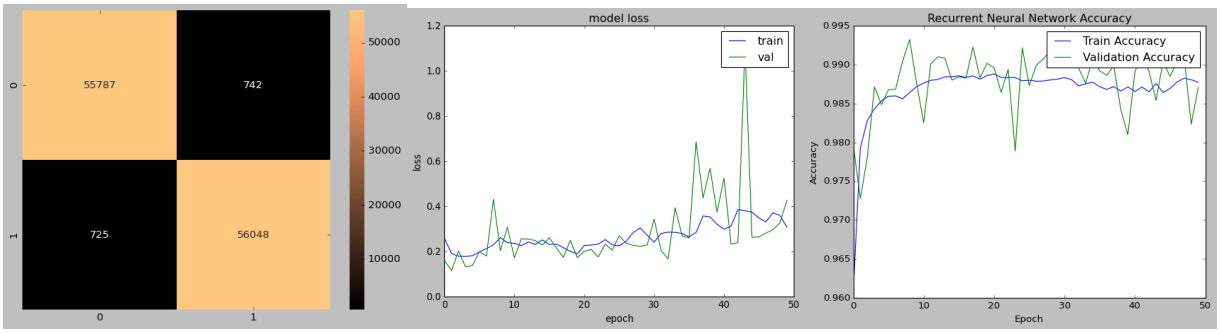


Figure 24 – Confusion matrix, accuracy, and loss curve of Recurrent Neural Network

LSTM and **GRU**: The accuracy and loss curves for both LSTM and GRU demonstrate similar patterns. There are swings in the curves, indicating some fluctuations during training.

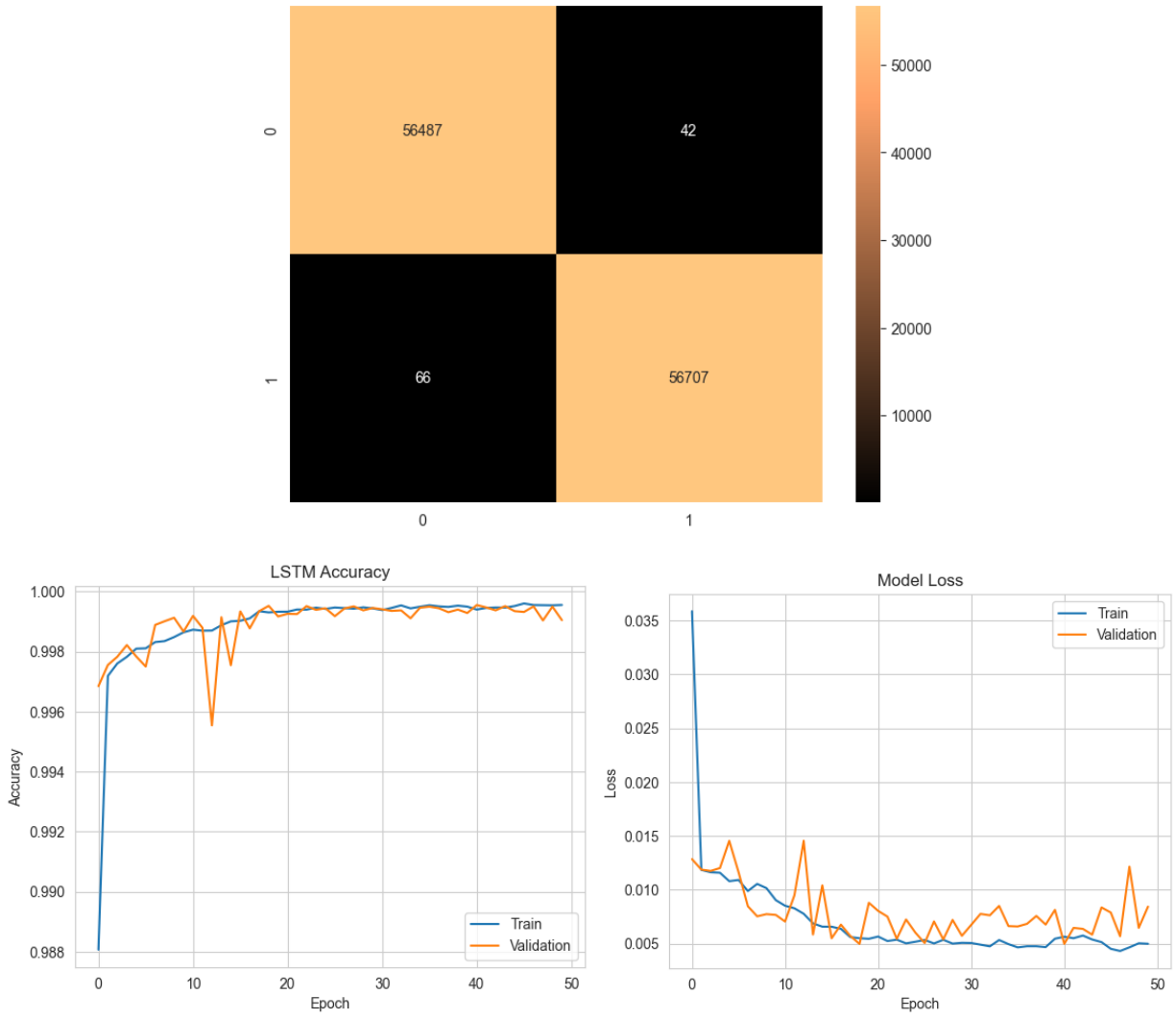


Figure 25 - Accuracy, and loss curve of Long Short Term Memory.

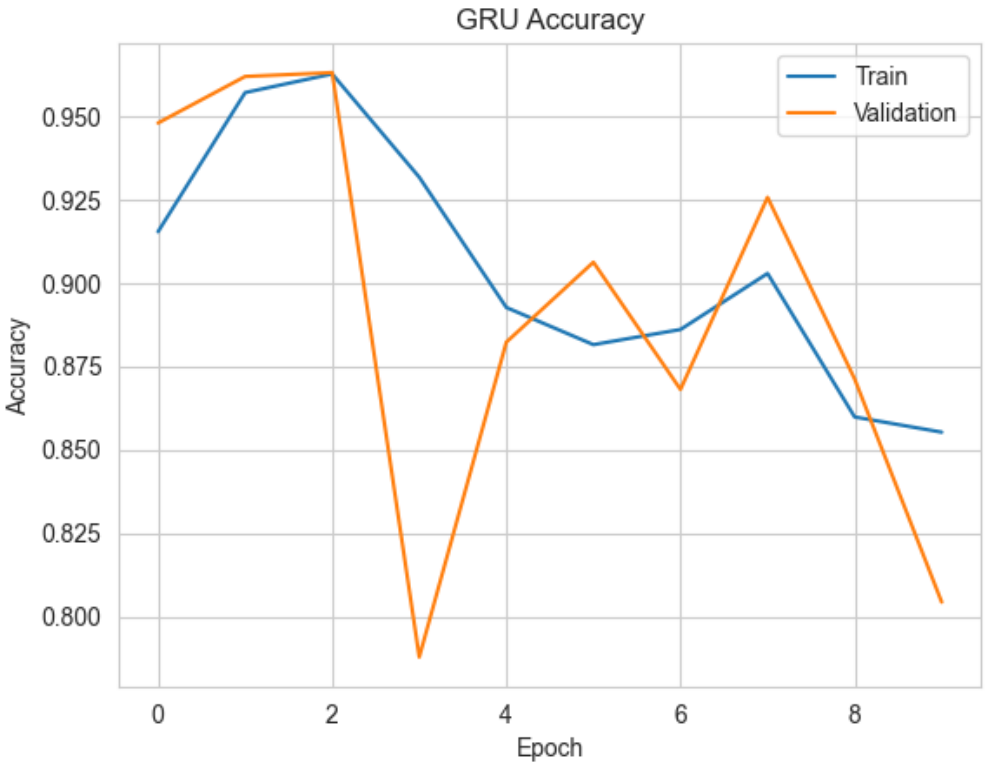


Figure 26 - Accuracy curve of Gated Recurrent Unit.

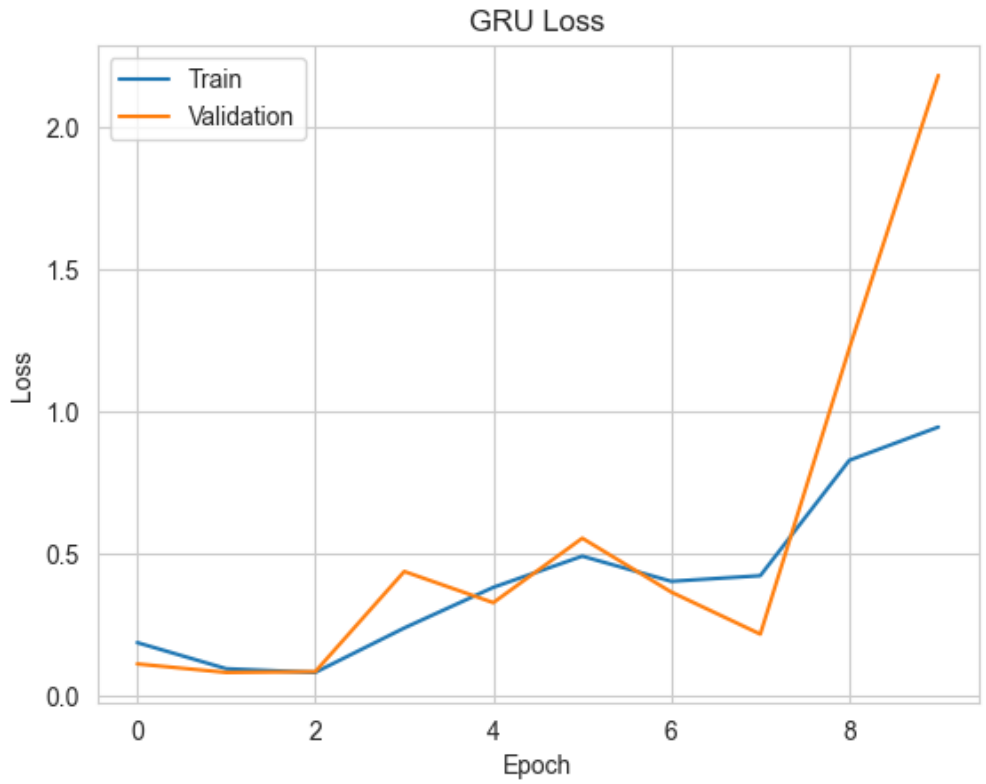


Figure 27 – Loss curve of Gated Recurrent Unit.

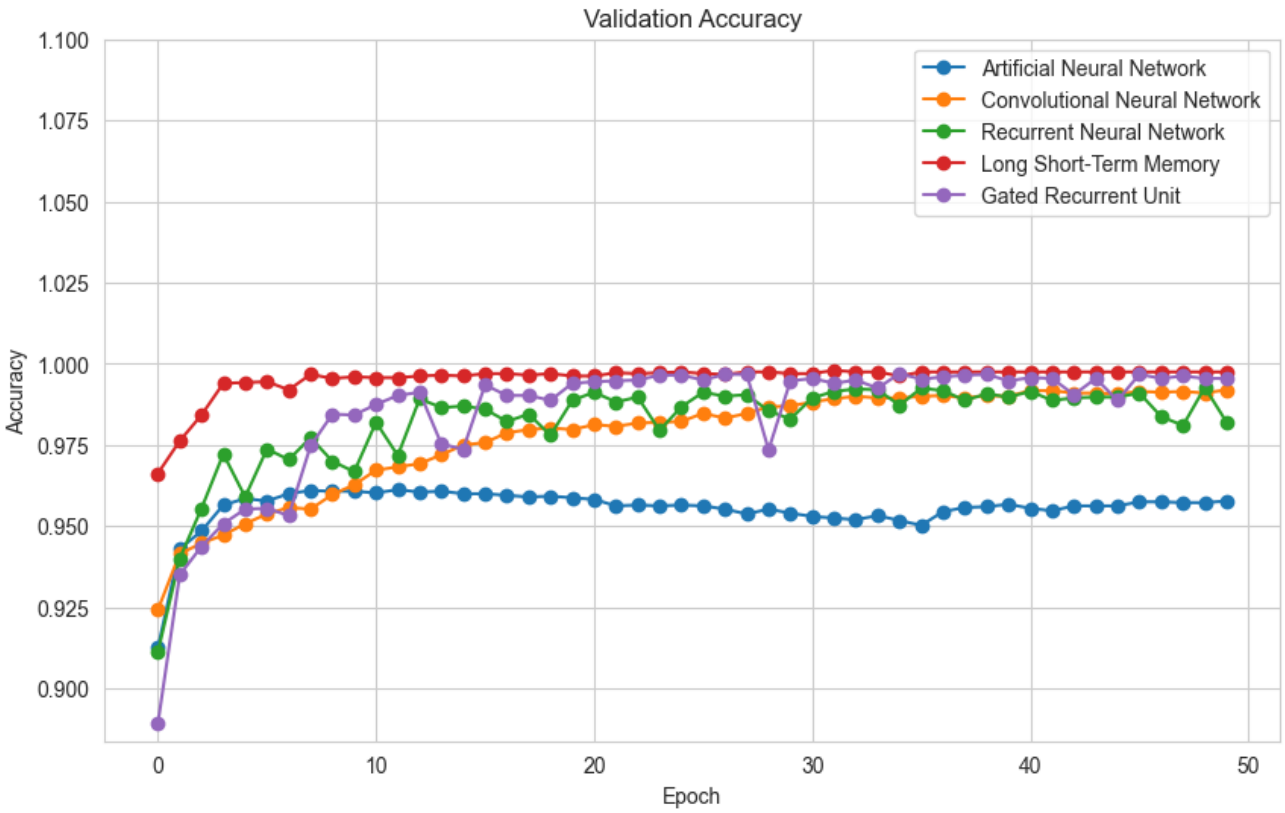


Figure 28 –Accuracy curves of deep learning models.

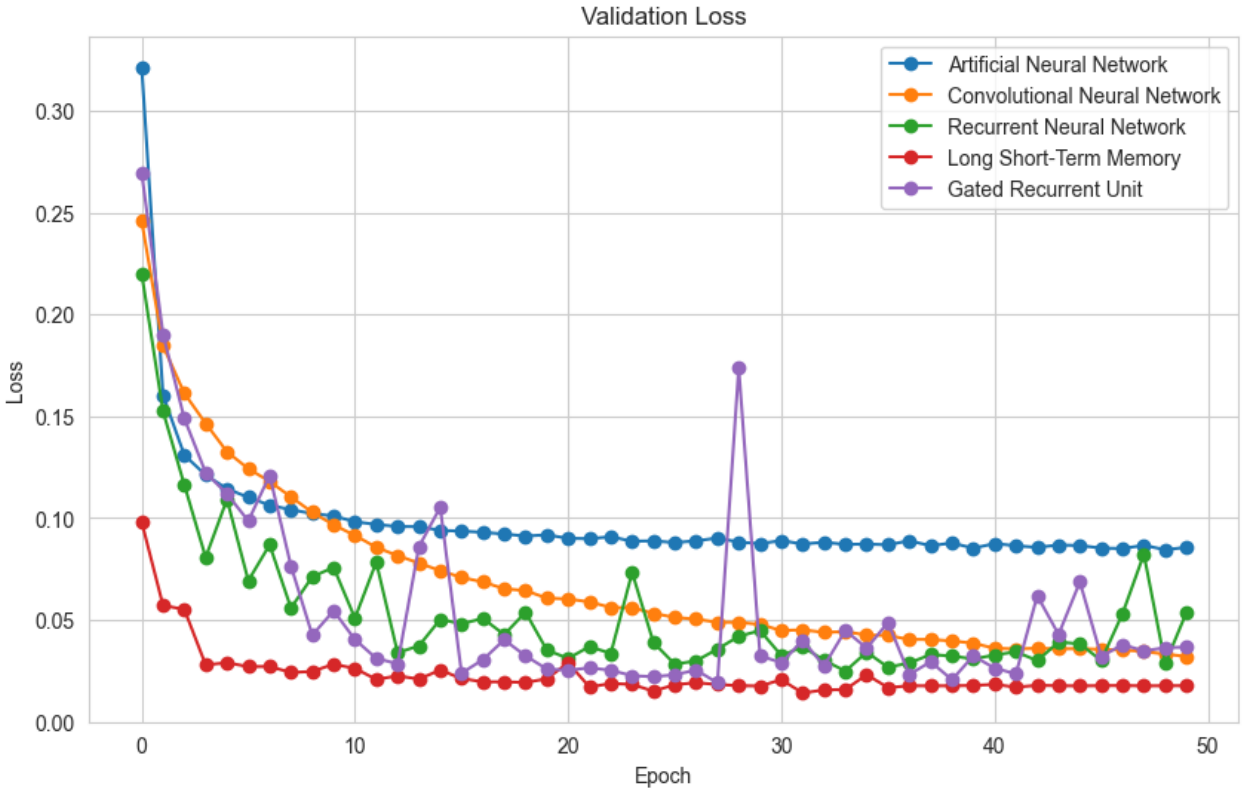


Figure 29 –Loss curves of deep learning models.

Lastly, I conducted an additional experiment consisting of 100 epochs for all neural networks. For this experiment, I excluded ANN and RNN from the model. Consequently, the voting classifier was tested on three different networks: CNN, LSTM, and GRU. The results showed that the GRU algorithm achieved the highest accuracy, outperforming both LSTM and CNN. However, our classifier's accuracy was slightly lower compared to the individual GRU model. From the results, we can infer that the GRU model was effective in detecting some fraudulent transactions that the LSTM and CNN models failed to identify.

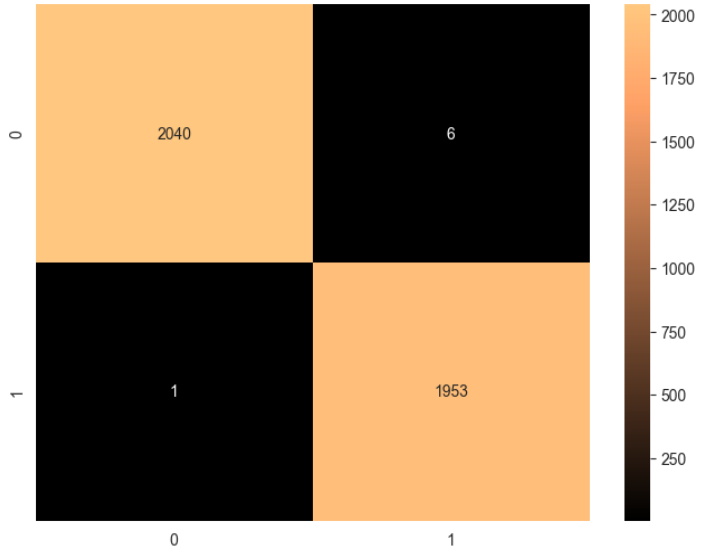


Figure 30 – Confusion matrix of Ensemble Voting Classifier (Hard)

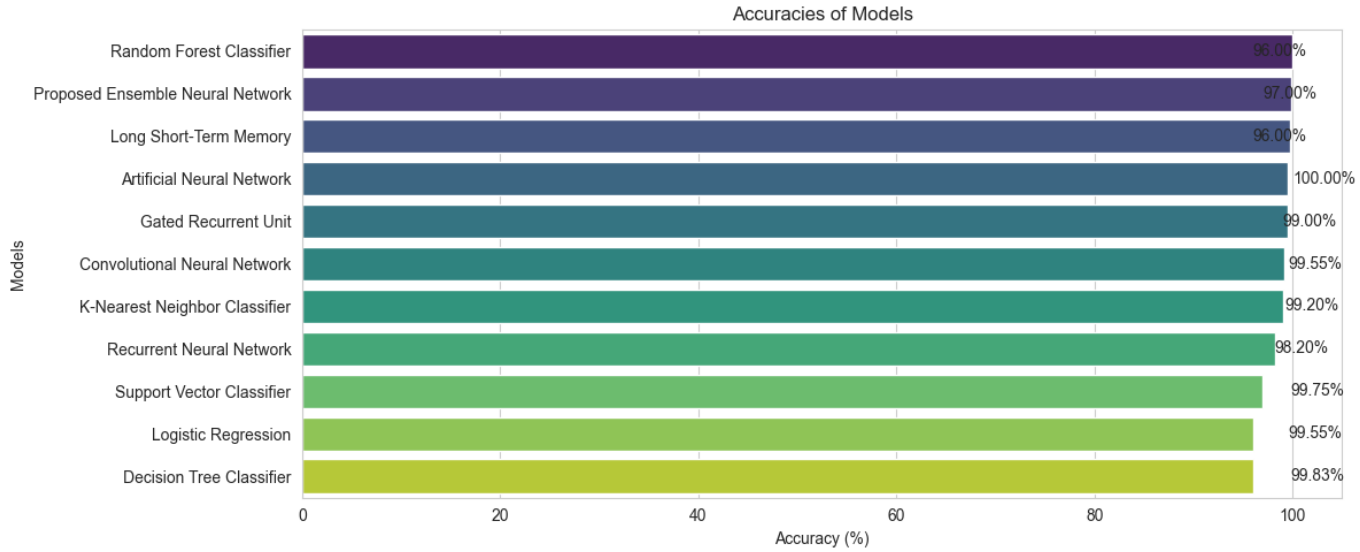


Figure 31 – Accuracies of all models.

CONCLUSION AND FUTURE WORK

Credit card fraud detection is a challenging problem that requires robust detection methods. The dataset lacked sufficient descriptions, which hindered the process of perfect feature selection. Additionally, the inclusion of an unnecessary feature had a significant impact on the model's performance. This highlights the importance of careful feature engineering and selection in such tasks.

The proposed ensemble voting approach, using hard and soft voting with neural network classifiers, occasionally exhibited lower accuracy compared to individual neural networks. The reason behind this outcome was the presence of observations that were difficult to classify correctly based on their true labels. Many of the neural networks struggled to predict these challenging cases, leading to failures in the hard voting classifier's ability to identify fraudulent transactions with false labels.

Improving the performance of the ensemble classifier may require addressing these challenging observations, such as refining the dataset or using more sophisticated techniques for feature extraction. Additionally, exploring other ensemble methods or tweaking the individual neural network models could potentially enhance the overall accuracy of the fraud detection system.

In the realm of credit card fraud detection, several promising directions for future work can be pursued to enhance the effectiveness of fraud detection systems. Firstly, focusing on improving data quality by incorporating more informative features and conducting rigorous data preprocessing techniques to mitigate the impact of irrelevant or noisy data. Secondly, developing anomaly detection algorithms that can adapt and evolve with the changing patterns of fraud is vital. Incorporating real-time data streams and investigating unsupervised learning techniques may aid in detecting emerging fraud patterns swiftly. Lastly, I would address in the future work the issue of interpretability in complex models would enhance the trustworthiness of the detection system and facilitate regulatory compliance. And ultimately providing enhanced security and protection to consumers and financial institutions.

APPENDIX

Dataset source

https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

Github link

https://github.com/YuriZmytrakov/credit_card_fraud_detection

REFERENCE

1. Eunji Kim a , Jehyuk Lee a , Hunsik Shin a , Hoseong Yang a , Sungzoon Cho a , ∗, Seung-kwan Nam b , Youngmi Song b , Jeong-a Yoon b , Jong-il Kim (2019): Champion-challenger analysis for credit card fraud detection: Hybrid ensemble and deep learning.

2. Hendi Yogi Prabowo Department of Accounting, Islamic University of Indonesia, Yogyakarta, Indonesia: A better credit card fraud prevention strategy for Indonesia.

3. Ajeet Singh, Anurag Jain (2022): An efficient credit card fraud detection approach using cost-sensitive weak learner with imbalanced dataset.

4. Nicholas Wong, Pradeep Ray, Greg Stephens, Lundy Lewis: Artificial immune systems for the detection of credit card fraud: an architecture, prototype, and preliminary results.

5. Abdul Razaque, Mohamed Ben Haj Frej, Gulnara Bektemyssova, Fathi Amsaad, Muder Almiani, Aziz Alotaibi, N. Z. Jhanjhi, Saule Amanzholova and Majid Alshammari: Credit Card-Not-Present Fraud Detection and Prevention Using Big Data Analytics Algorithms

6. Monika Arya & Hanumat Sastry G (2020): Deep Ensemble ALgorithm Framework for Credit Card Fraud Detection in Real-Time Data Stream with Google TensorFlow.

7. Yiğit Kültür, Mehmet Ufuk Çağlayan: Hybrid approaches for detecting credit card fraud

8. Abdallah, A., Maarof, A., & Zainal, A. (2016). Fraud detection system.

9. Brause, R., Langsdorf, T., Hepp, M. (1999). Neural data mining for credit card fraud detection. In Proceedings of the 11th International Conference on Tools with Artificial Intelligence.

10. Phua, C., Gayler, R., Lee, V., & Smith-Miles, K. (Year not provided). On the communal analysis suspicion scoring for identity crime in streaming credit applications. Eur. J. Oper. Res.

11. Esenogho, E., Mienye, I. D., Swart, T. G., Aruleba, K., & Obaido, G. (Year not provided). A Neural Network Ensemble With Feature Engineering for Improved Credit Card Fraud Detection. IEEE Access.

12. Ileberi, E., Sun, Y., & Wang, Z. (Year not provided). Performance Evaluation of Machine Learning Methods for Credit Card Fraud Detection Using SMOTE and AdaBoost. IEEE Access.

13. Carcillo, F., Pozzolo, A. D., Le Borgne, Y.-A., Caelen, O., Mazzer, Y., & Bontempi, G. (May 2018). SCARFF: A scalable framework for streaming credit card fraud detection with Spark.

14. Carcillo, F., Le Borgne, Y.-A., Caelen, O., Kessaci, Y., Oblé, F., & Bontempi, G. (Year not provided). Combining unsupervised and supervised learning in credit card fraud detection. Inf. Sci.

15. Tingfei, H., Guangquan, C., & Kuihua, H. (Year not provided). Using Variational Auto Encoding in Credit Card Fraud Detection. IEEE Access.

16. The Ascent (2020). Identity theft and credit card fraud statistics.

17. Machine Learning Group-ULB. (2013, September). Credit Card Fraud Detection, Version 3.

18. Misra, S., Thakur, S., Ghosh, M., & Saha, S. K. (2020). An autoencoder based model for detecting fraudulent credit card transactions.

19. Venkatanareshbabu Kuppili, Diwakar Tripathi, Damodar Reddy Edla. (2019) Credit cars fraud classification using spiking extreme learning machine.

20. Marcos Roberto Machado, Salma Karray. (2022). Assessing credit risk of commercial customers using hybrid machine learning algorithms.

21. Palak GuptaAnmol VarshneyMohammad Rafeek KhanRafeeq AhmedMohammed ShuaibShadab Alam (2023) Unbalanced Credit Card Fraud Detection Data: A Machine Learning-Oriented Comparative Study of Balancing Techniques.

22. Fawaz Khaled Alarfaj, Iqra Malik, Hikmat Ullah Khan, Naif Almusallam, Muhammad Ramzan, Muzamil Ahmed (2022). Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms.

23. Naoufal Rtayli, Nourddine Enneya (2023). Credit card fraud detection using predictive features and machine learning algorithms.

24. T. John Berkmans and S. Karthick (2022). A widespread survey on machine learning techniques and user substantiation methods for credit card fraud detection.

25. Mienye, Ibomoiye Domor; Sun, Yanxia. Applied Sciences (2023). A Machine Learning Method with Hybrid Feature Selection for Improved Credit Card Fraud Detection.

26. John O. Awoyemi; Adebayo O. Adetunmbi; Samuel A. Oluwadare (2017) Credit card fraud detection using machine learning techniques: A comparative analysis.

27. Faleh Alshameri and Ran Xia (2023). Credit card fraud detection: an evaluation of SMOTE resampling and machine learning model performance.

28. Thakur Santosh; Dharavath Ramesh (2020). Machine Learning Approach on Apache Spark for Credit Card Fraud Detection.

29. Gayan K. Kulatilleke (2022). Challenges and Complexities in Machine Learning based Credit Card Fraud Detection.

30. Tyagi, Rishabh (2021). Credit Card Fraud Detection Using Machine Learning Algorithms.