POZNAŃ UNIVERSITY OF ECONOMICS AND BUSINESS
Research Papers in Economics and Finance *Vol. 9, No. 1, 2025: 55–80*
https://ref.ue.poznan.pl *https://doi.org/10.18559/ref.2025.1.2108*

Credit card fraud detection and risk management strategies: A deep learning-based approach for EU banks

Habib Zouaoui[1] **Meryem-Nadjat Naas**[2]

Abstract

This study explores supervised ML-DL based approaches for enhancing credit card fraud detection and improving financial risk management systems for EU banks. This research proposes an ensemble method based on majority voting (Hard Voting Classifier) of deep learning models to detect fraud transaction. Artificial Neural Network (ANN), Convolution Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) have been used as deep learning models. First, the most significant features that affect the type of transaction (fraud or not fraud) have been selected. After that, the ML-DL models were applied. The performance of the proposed approach is tested using a confusion matrix, recall, precision, F-measure and accuracy. The proposed method is tested using accurate data that consists of 540,099 transactions recorded in Kaggle repository dataset of two days based on European card holder for September, 2023. The result shows that the Random Forest (RF) model detected anomalies with 99.99% accuracy, F1-score with 1.00, and excellent recall with 99.99%. As a result, the machine learning model based on RF approach shows promise as a real-time anomaly detection method with high performance and low computational cost.

Keywords

- credit card
- fraud detection
- deep learning
- risk management
- EU banks

Article received 21 February 2025, accepted 11 June 2025.

Suggested citation: Zouaoui, H., & Naas, M. N. (2025). Credit card fraud detection and risk management strategies: A deep learning-based approach for EU banks. *Research Papers in Economics and Finance*, *9*(1), 55–80. https://doi.org/10.18559/ref.2025.1.2108

[1] University of Relizane, Cité Bourmadia, W. Relizane BP 48000, Algeria, corresponding author: habib.zouaoui@univ-relizane.dz

[2] University of Relizane, Cité Bourmadia, W. Relizane BP 48000, Algeria, meryemnadjat.naas@univ-relizane.dz

Introduction

Artificial Intelligence and Machine Learning are revolutionising fraud detection in banking and financial institutions. Leveraging sophisticated algorithms, these technologies analyse vast datasets to identify patterns indicative of fraudulent activities. Machine learning models continuously learn and adapt, enhancing their accuracy over time. By automating the detection process, AI minimises false positives and accelerates the identification of suspicious transactions. This proactive approach not only safeguards financial assets but also ensures a more efficient and secure environment for both institutions and customers, bolstering the resilience of the financial sector (Kolli et al., 2023). Whereas, the identification of fraudulent transactions has become a major factor affecting the greater utilisation of electronic payment. As a result, efficient and effective methods for detecting fraud in credit card transactions are demanded. Realistically, Credit card fraud has become a significant challenge for banks in the European Union. The projected credit card fraud losses between 2020 and 2025 reflect the ongoing challenges that the banking sector faces in dealing with financial fraud risks. However, in the period between 2020–2021, the EU witnessed a notable surge in card-not-present (CNP) fraud, exacerbated by the COVID-19 pandemic, which drove more consumers to online shopping. In 2020, card fraud losses amounted to approximately €1.5 billion across the EU, and it has increased to around €1.7 billion due to a continued rise in online fraud and e-commerce activities (*Nilson Report*, 2020). In 2022–2023, the growth of digital wallets, mobile payments and cross-border fraud continued to fuel fraud risks. It was estimated at €1.8 billion in 2022 (ECB, 2023), with CNP fraud accounting for 80% of total losses (Buzzard, 2022). In 2023, it was projected to exceed €2 billion, driven by the increasing adoption of instant payments, mobile wallet vulnerabilities and AI-powered fraud techniques (Detura et al., 2022). Furthermore, between 2024 and 2025, despite increasing security measures, fraud risks are expected to rise even further as new technologies like cryptocurrencies and real-time payments gain traction. Moreover, they were expected to reach €2.2 billion in 2024 due to higher transaction volumes in instant payments, AI-driven fraud and cross-border fraud. In 2025, these losses are forecast to exceed €2.5 billion, fuelled by the rise in synthetic identity fraud, cryptocurrency fraud and SIM swapping (ECB, 2025).

In order to minimise fraud losses, the development of advanced techniques and technologies has become essential for detecting and preventing fraudulent activity while effectively managing associated risks. Moreover, financial fraud detection and risk management are evolving rapidly with advancements in AI and ML. By leveraging deep learning and advanced techniques, organisations can significantly reduce fraud losses and improve operational efficiency. Financial fraud

detection and risk management are critical areas in the financial industry, aimed at identifying fraudulent activities and mitigating associated risks. Over the years, advancements in technology, particularly in Artificial Intelligence (AI), Machine Learning (ML) and Deep Learning (DL), have revolutionised these fields (Chaudhari & Kaur, 2025).

Therefore, the key challenges in this domain include handling imbalanced datasets, ensuring real-time detection and maintaining data privacy. To address these challenges, we discuss emerging trends such as federated learning, reinforcement learning and self-supervised learning, which enable secure, low-latency and scalable fraud detection systems. Furthermore, this literature review explores the evolution of fraud detection techniques, the challenges faced and the advancements made in recent years. The review is structured around such key themes as traditional methods, machine learning approaches, deep learning techniques and emerging trends.

In this study, we aim to test the following hypothesis (H1): Deep Learning (DL) models provide more accurate fraud risk management for credit card transactions than traditional Machine Learning (ML) models. The central research question is: which of the two approaches, i.e. Machine Learning models (Logistic Regression, Support Vector Classifier (SVC), Decision Tree Classifier, Random Forest Classifier, K-neighbor Classifier (KNN)) or Deep Learning models (LSTM, GRU, ANN, RNN, CNN), offers superior performance in managing credit card fraud risk?

1. Credit card fraud and risk management

Credit card fraud and risk management in EU banks involves a combination of advanced technologies, regulatory compliance and customer-centric strategies to detect, prevent and mitigate fraudulent activities. The European Union (EU) has a robust regulatory framework that governs data protection, payment security and consumer rights, which significantly influences the ways in which banks manage fraud risks. Below is a detailed overview of credit card fraud risk management in EU banks (BIS, 2024):

1. Regulatory Compliance: EU banks must adhere to strict regulations designed to protect consumers and ensure secure payment systems such as:

- Payment Services Directive 2 (PSD2): requires strong customer authentication (SCA) for online transactions, reducing the risk of fraud,
- General Data Protection Regulation (GDPR): ensures the secure handling of personal data and imposes heavy penalties for data breaches,

- Anti-Money Laundering (AML) Directives: mandate monitoring and reporting of suspicious transactions to prevent money laundering and terrorist financing.

2. Strong Customer Authentication (SCA): PSD2 mandates SCA for most electronic payments, which involves:

- Two-Factor Authentication (2FA): requires at least two of the following: something the customer knows (e.g. password or PIN), something the customer has (e.g. mobile device or card reader), something the customer is (e.g. fingerprint or facial recognition),
- Dynamic Linking: ensures that the authentication code is uniquely linked to the transaction amount and recipient.

3. Advanced Fraud Detection Technologies; EU banks leverage cutting-edge technologies to detect and prevent fraud:

- Real-Time Transaction Monitoring: uses AI and machine learning to analyse transactions in real time and flag suspicious activities,
- Behavioural Analytics: monitors customer behaviour to identify deviations from normal spending patterns,
- Geolocation Tracking: verifies the location of the cardholder and compares it to the transaction location,
- Risk Scoring Models: assigns risk scores to transactions based on factors such as transaction amount, location and merchant category.

4. Data Security and Encryption; EU banks prioritise the protection of sensitive cardholder data:

- End-to-End Encryption (E2EE): encrypts data during transmission and storage to prevent unauthorised access,
- Tokenisation: replaces sensitive card data with unique tokens to reduce the risk of data breaches,
- PCI DSS Compliance: adheres to the Payment Card Industry Data Security Standard (PCI DSS) to ensure secure handling of cardholder information.

5. Collaboration and Information Sharing; EU banks collaborate with other financial institutions, card networks and law enforcement agencies to combat fraud:

- Fraud Data Sharing: shares fraud-related data and trends to identify emerging threats,
- European Banking Federation (EBF): participates in industry initiatives to develop best practices for fraud prevention.

6. Customer Education and Awareness; EU banks educate customers on how to protect themselves from fraud:

- Fraud Prevention Tips: provide guidance on safeguarding card information and recognising phishing attempts,
- Real-Time Alerts: send notifications for suspicious transactions to keep customers informed.

7. Incident Response and Recovery; EU banks have robust incident response plans to address fraud incidents:

- Fraud Investigation Teams: dedicated teams investigate flagged transactions and confirm fraud cases,
- Customer Notification: notifies affected customers and provides guidance on securing their accounts,
- Chargeback Management: handles chargebacks efficiently to minimise financial losses.

8. Key Metrics for Fraud Risk Management; EU banks monitor key metrics to assess the effectiveness of their fraud risk management strategies:

- Fraud Detection Rate: percentage of fraudulent transactions detected,
- False Positive Rate: percentage of legitimate transactions flagged as fraudulent,
- Chargeback Rate: number of chargebacks as a percentage of total transactions,
- Average Time to Detect Fraud: time taken to identify and respond to fraud incidents.

Finally, the BIS Innovation Hub explores the use of artificial intelligence (AI) to support central banks and supervisors in their missions. So far, eight projects have employed AI methods, including: Ellipse, Aurora, Gaia, Symbiosis, Raven, Neo, Spectrum and Insight. They cover a wide range of use cases from information collection and statistical compilation, payments oversight and supervision, as well as macroeconomic and financial analysis to monetary policy analysis These projects draw on both in-house expertise and that of external providers.

We will analyse the effectiveness of traditional rule-based systems versus modern AI-driven approaches, highlighting the growing adoption of hybrid models, federated learning and behavioural biometrics to enhance detection accuracy while ensuring compliance. Key findings reveal that Card-Not-Present (CNP) fraud accounts for the majority of cases (73%), while synthetic identity fraud and authorised push payment (APP) scams are emerging as significant and growing threats. The study also identifies critical gaps, such as fragmented fraud data across EU member states and the trade-off between model explainability and performance. However, in order to address these challenges, we will propose a regulatory-aligned AI framework combining lightweight ML models, real-time anomaly detection and cross-border data collaboration. Our recommendations emphasise the need for standardised fraud datasets, privacy-preserving techniques and scalable risk man-

agement strategies tailored to EU banking ecosystems. This research contributes to the ongoing discourse on securing digital payments while fostering innovation in financial risk mitigation (Vadisena et al., 2024).

2. Literature review

Table 1 presents a structured analysis of recent studies (2020–2025) using Deep Learning (DL) models for credit card fraud detection, including a quantitative breakdown of model popularity and performance metrics.

Table 1. Reviewed previous studies

Author(s) and year	Study title	Methodology	Reported accuracy	Key findings
Moturi et al. (2024)	Optimizing credit card fraud detection using deep learning by smote-enn technique	Robust deep-learning approach using LSTM-GRU models with the SMOTE-ENN method versus ML classifiers.	99.7% (recall)	The experimental results showed that combining the proposed deep learning ensemble with the SMOTE-ENN method is superior to other widely used ML classifiers.
Chidananda (2025)	Deep learning for fraud detection in financial transactions using CNN-LSTM hybrid and GRU Model	Hybrid deep learning model combining CNN-LSTM and CNN-GRU.	88.85% (accuracy), 85.39% (recall), 88.30% (F1-Score)	Hybrid deep learning model with CNN-LSTM better than alternative CNN-GRU models.
Chaudhari et al. (2025)	Enhancing global banking security: A novel approach integrating federated learning and CNN-GRU for effective anti-money laundering measures	Anti-Money Laundering (AML), Federated Learning, Privacy-Preserving, Intrusion Detection, Hybrid CNN-GRU Model.	98.7% (accuracy)	The proposed method offers a scalable and effective solution for the global banking sector. It also surpasses traditional techniques in terms of security and efficiency in the battle against money laundering in the emerging financial scenario.

cont. Table 1

Author(s) and year	Study title	Methodology	Reported accuracy	Key findings
Tayebi & El Kafhali (2025)	A novel approach based on XGBoost classifier and Bayesian optimisation for credit card fraud detection	This study proposes an enhanced XGBoost algorithm for detecting fraudulent transactions using an intelligent technique that tunes the hyperparameters of the algorithm through Bayesian optimisation.	99.96% (accuracy), 87.40% (recall) 98.79% (F1-AUC)	For Data 1, the best performance was obtained using SMOTE. For Data 2, the random under-sampling technique yielded the highest performance.
Mienye & Swart (2024)	A hybrid deep learning approach with generative adversarial network for credit card fraud detection	Hybrid deep learning framework that integrates Generative Adversarial Networks (GANs) with Recurrent Neural Networks (RNNs) versus LSTM-GRU.	99.2% (recall)	This work highlights the potential of GANs combined with deep learning architectures to provide a more effective and adaptable solution for credit card fraud detection.
Sulaiman et al. (2024)	Credit card fraud detection using improved deep learning models	Three deep learning models, i.e. AutoEncoder (AE), Convolution Neural Network (CNN), and Long Short-Term Memory (LSTM), are proposed to investigate how hyperparameter adjustment impacts the efficacy of deep learning models used to identify credit card fraud.	Accuracy (99.2%), detection rate (93.3%), and area under the curve (96.3%)	The results demonstrate that LSTM significantly outperformed AE and CNN.
Wahab et al. (2024)	Credit card default prediction using ML and DL techniques	Evaluates the efficacy of a DL model (ANN) and compares it to other ML models, such as Decision Tree (DT) and AdaBoost.	Accuracy (82%)	The evaluation indicates that the AdaBoost and DT exhibit the highest accuracy rate of 82% in predicting credit card default, surpassing the accuracy of the ANN model, which is 78%.

Source: authors' analysis based on literature review (2025).

Several research works have proposed different methods in tackling credit card fraud. From Regression, Random Forest and KNN to Neural networks. Ghosh and Reilly (1994) were the first to apply neural networks for fraud detection. They used a large sample of labelled credit card transactions to train half a dozen neural networks, which was then validated on validation data consisting of account activities over a two-month period. Lost or stolen cards, application fraud, counterfeit fraud, mail-order fraud and NRI (non-received issue) fraud were all utilised to train the neural network (Misra et al., 2020). Brause et al. (1999) used association rule mining and neural networks to reduce the false positive rate in their study. Several supervised and unsupervised machine learning and optimisation algorithms have been used to detect credit card fraud in recent years. Therefore, the above table is a summarised comparison of the results from recent studies (2022–2025) on credit card fraud detection using deep learning models versus machine learning algorithms. These results highlight the key findings and performance metrics from the studies. Effectively, the choice between ML and DL depends on the dataset size, data complexity and real-time requirements of the fraud detection system. ML works well for traditional fraud detection in smaller datasets, while DL is suited for large, complex datasets, where deep pattern recognition and higher accuracy are needed.

3. Materials and methods

3.1. Definitions

Table 2 below presents a summary of the definitions, **mathematical formulation**, key components and descriptions of deep learning algorithms used in our research, including LSTM, CNN, GRU, RNN, ANN and the KNN model.

Table 2. Definitions, mathematical formulation, key components and descriptions of deep learning algorithms

Model	Definition	Mathematical formulation	Key components & descriptions
LSTM (Long Short-Term Memory)	LSTM is a type of RNN that addresses the vanishing gradient problem, enabling the learning of long-term dependencies. It uses gates (input, forget,	$r_t = \sigma(W_f.[h_{t-1}, x_t]) + b_f$ $d_t = \tanh(W_d.[h_{t-1}, x_t]) + b_d$ $f_t = \sigma(W_i.[h_{t-1}, x_t]) + b_i$ $C_t = f_t.C_{t-1} + r_t.d_t$ $o_t = \sigma(W_0.[h_{t-1}, x_t]) + b_0$ $h_t = o_t \tanh C_t$	Input Gate (x_t): controls the amount of incoming information written to the cell state. Forget Gate (f_t): decides how much of the previous memory is kept.

cont. Table 2

Model	Definition	Mathematical formulation	Key components & descriptions
	and output) to control the flow of information in the network.		Output Gate (o_t): determines the next hidden state. Cell State (C_t): the long-term memory that carries information across time steps.
GRU (Gated Recurrent Unit Model)	GRU is a simplified version of LSTM, combining the input and forget gates into a single update gate. It is more efficient and works well for learning long-term dependencies with fewer parameters.	$z_t = \sigma(W_z.[h_{t-1}, x_t]) + b_z$ $r_t = \sigma(W_r.[h_{t-1}, x_t]) + b_r$ $\hat{h}_t = \tanh(W_h.[r_t.h_{t-1}, x_t] + b_h)$ $h_h = (1 - z_t).h_{t-1} + z_t.\hat{h}_t$	Update Gate (z_t): controls the amount of previous memory retained. Reset Gate (r_t): decides how much of the past hidden state is discarded. Candidate Hidden State (h_t): a potential new hidden state.
RNN (Recurrent Neural Network)	RNNs process sequential data by maintaining a hidden state that is updated with each time step based on the current input and the previous hidden state. They suffer from vanishing gradients with long sequences.	$h_t = \sigma(W_{hh}.h_{t-1} + W_{xh}x_t + b)$	Hidden State (h_t): the memory of the previous time steps. Weight Matrices (W_{hh}, W_{xh}): used to transform the input and hidden state to compute the next hidden state.
CNN (Convolutional Neural Network)	A Convolutional Neural Network (CNN) is a deep learning model specifically designed for processing structured grid data, such as images. It uses convolutional layers to automatically and adaptively learn spatial hierarchies of features from input data.	1. Convolution Operation: $(I \cdot K)(i \cdot j) = \sum_m \sum_n I(i - m, j - n).K(m, n)$ 2. Activation Function: $A = f(Z)$, where $Z = W \cdot X + b$ 3. Pooling Operation: $P(i, j) = \max_{m,n}(X(i.s + m, j.s + n))$ 4. Fully Connected Layer: $y = f(W.X + b)$ 5. Output Layer: $\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$ 6. Loss Function: $\text{Loss} = -\sum_i y_i \log(\hat{y}_i)$ 7. Backpropagation: $W = W - \eta \frac{\partial \text{Loss}}{\partial W}$	Convolutional Layers: apply filters to detect local features (edges, textures). Pooling Layers: reduce dimensionality to retain important features while improving efficiency. Fully Connected Layers: use output from convolutional layers for classification or regression.

cont. Table 2

Model	Definition	Mathematical formulation	Key components & descriptions
ANN (Artificial Neural Network)	ANN is a network of interconnected neurons used for supervised learning tasks. It consists of layers of nodes (neurons), where each node represents a mathematical function.	1. Neuron Computation: $z = \sum_{i=1}^{n} w_i x_i + b$, $a = f(z)$ 2. Forward Propagation: – compute activations layer by layer. 3. Loss Function: – measure prediction error (e.g. MSE, cross-entropy). 4. Backpropagation: – compute gradients and update weights and biases.	Input Layer: takes the input features. Hidden Layers: layers that apply non-linear transformations to the input data. Output Layer: produces the final predictions (classification or regression).

Source: Bolton & Hand (2002); Naas & Zouaoui (2024); Zareapoor & Shamsolmoali (2015); Zouaoui & Naas (2023).

3.2. Data description

The study was conducted using real-world time series data from a credit card fraud dataset, obtained through a research collaboration focused on big data mining and fraud detection. The dataset was downloaded from Credit Card Fraud Detection repository on Kaggle. It contains 540,099 credit card transactions recorded over a two-day period in September 2023, involving European card holders. Among these, 270,049 transactions were labelled as fraudulent, indicating a highly imbalanced dataset. Notably, the fraud class represents approximately 50% of the total transactions, which is atypical compared to real-world distributions. The dataset includes 31 attributes, each representing different features relevant to transaction behaviour and fraud detection (Table 3).

Table 3. Description of the attributes in the credit card fraud detection dataset

N°	Columns	Data type	Description
1	Time	Numeric	Time represents the seconds elapsed between each transaction and the first transaction in the dataset.
2–29	V1–V28	Numeric	V1 to V28 are transformed using Principal Component Analysis. For security reasons, original attribute names are not disclosed.
30	Amount	Numeric	Transaction amount.
31	Class	Numeric	Binary classification: ‘1’ indicates fraud; ‘0’ indicates a legitimate transaction.

Source: own study.

3.3. Initial analysis

The attributes' properties and characteristics were carefully examined during the initial stages of the analysis. Key focuses at this stage included the distribution of variables, correlation dependencies, and uncovering data-driven insights. The data analysis was structured around three main focal criteria, which are detailed in the following subsections.

3.3.1. Univariate analysis

After inspecting the dataset for null values and verifying data types, it was found that 0% duplicate observations were present and subsequently removed. An analysis of the target variable "Class" through a count plot revealed a significant class imbalance. The number of non-fraudulent transactions was 270,050, while 270,049 were fraudulent – each comprising nearly 50% of the dataset (Figure 1). This unusual balance is atypical compared to real-world fraud detection scenarios, where fraudulent cases are usually rare. To address this, oversampling techniques will be applied in subsequent stages of the analysis to ensure the robustness of model training and evaluation (Figure 2).

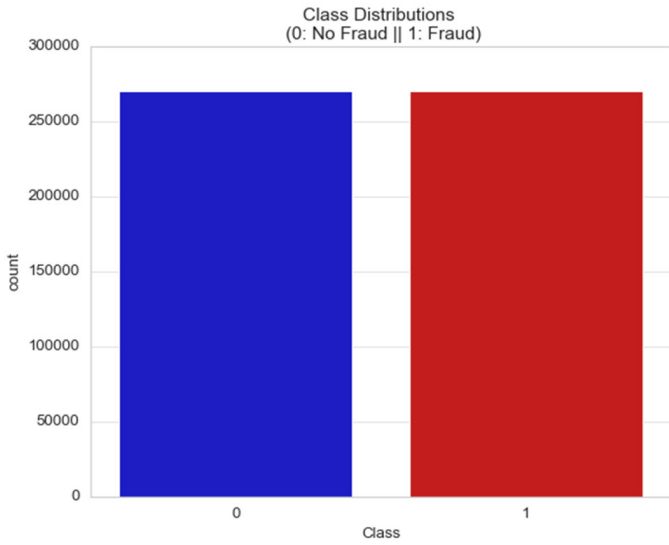

Figure 1. Distribution of Class variable

Source: based on python code GitHub.

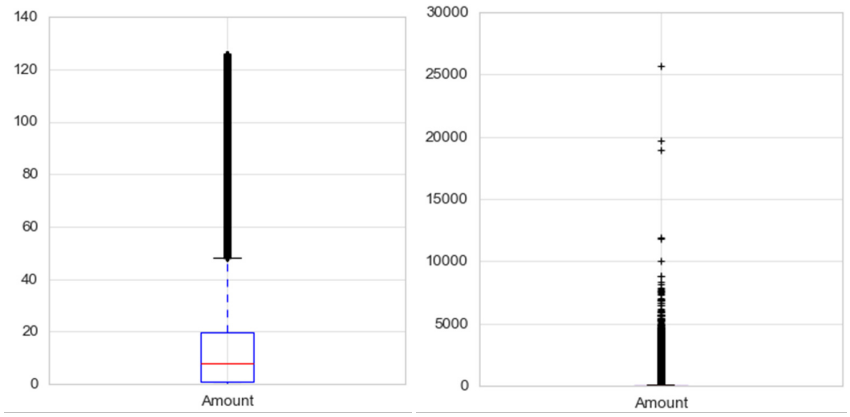

Figure 2. Anomaly detection and transformation using the Interquartile Range (IQR) method

Source: based on python code GitHub.

The Amount variable was plotted vertically, revealing that most transaction values are concentrated in the lower range, with only a few instances involving large amounts. If left unaddressed, these outliers could significantly bias the prediction performance of the fraud detection model. To mitigate this, we applied Median Imputation following the detection of anomalies. In this approach, extreme values are replaced with the median, ensuring the integrity of the distribution without the influence of outliers. The Interquartile Range (IQR), calculated using the formula IQR = Q3 − Q1, was used to identify the outliers. For the Amount feature, the IQR was computed using Python, and extreme values were adjusted accordingly. Upon re-plotting the Amount variable after this transformation, the values predominantly fall within the range of approximately $100 to $200, indicating a successful reduction of anomalies (Figure 3).

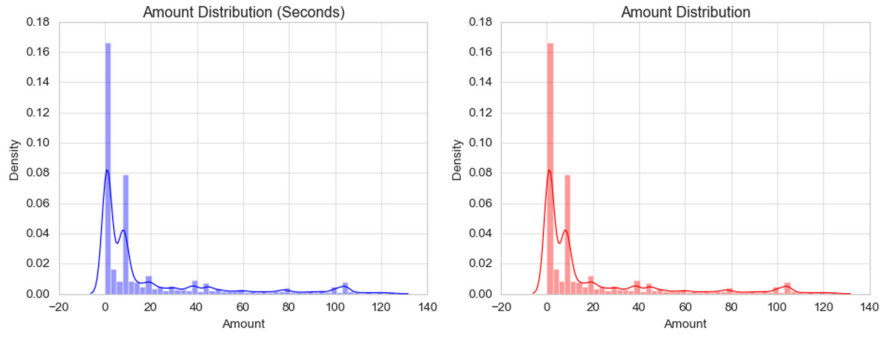

Figure 3. Time and Amount distribution plot

Source: based on python code GitHub.

From the distribution analysis of the Time and Amount features, no significant patterns were immediately evident. However, it was observed that transaction amounts close to zero exhibited the highest concentration, indicating that most transactions involve relatively small sums. In contrast, the Time variable – plotted in seconds over two consecutive days – showed a higher transaction density during daytime hours. To interpret the time feature in a more intuitive format, it can be converted to hourly intervals by dividing the values by 3600, as one hour equals 3600 seconds.

3.3.2. Bivariate analysis

A correlation heatmap was plotted to visualise the two-dimensional correlation matrix (Figure 4), assessing the pairwise relationships between all 31 attributes in the dataset. The heatmap illustrates correlation coefficients ranging from +1 (perfect positive correlation) to approximately –0.5 (moderate negative correlation), represented using a two-colour scale: red for positive correlations and blue for negative correlations. The intensity of the colour indicates the strength of the correlation, with values also displayed within each cell for reference. The analysis revealed that features V1 to V28 show little to no correlation with each other. Due to data anonymisation for security purposes, the original names and meanings of these features are not disclosed. As a result, traditional descriptive statistical analysis for these components offers limited interpretive value.

Although the correlation heatmap appears visually cluttered due to its size (it is readable in the notebook file), several key relationships can be highlighted. Notable correlations include:

- Time / V3 = –0.42
- Amount / V2 = –0.53
- Amount / V5 = –0.39
- Amount / V7 = 0.40
- Amount / V20 = 0.34

These correlations represent moderate linear relationships. Other minor correlations were observed in the range of –0.3 to 0.3, but they are not considered statistically significant for this analysis. We can conclude that Time and Amount are the most important variables.

Next, we plotted the distributions of features V1 to V28, grouped by the two target classes: Genuine and Fraud. The density is represented on the vertical axis, while the feature values are shown on the horizontal axis. The visualisation is organized into a grid layout, with four features per row and a total of seven rows,

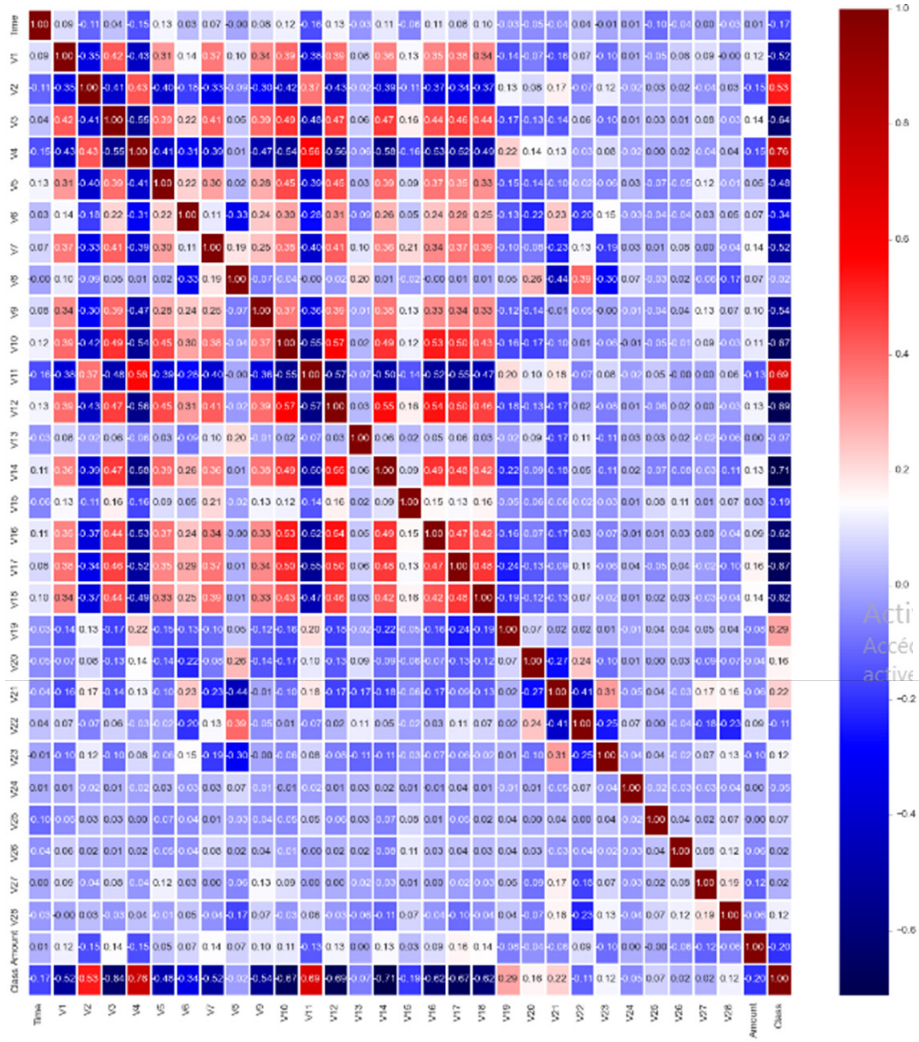

Figure 4. Correlation heatmap of the credit card fraud detection dataset

Source: based on python code GitHub.

covering all 28 principal components. The purpose of this analysis is to explore how the distributions of these features differ between the two classes, and to identify any patterns or feature behaviours that could aid in distinguishing fraudulent transactions from genuine ones.

The distributions for both classes generally resemble Gaussian bell curves, indicating normal-like behaviour across most features. However, certain features exhibit clear differences between the Genuine and Fraud classes. Specifically, features

V3, V9, V10, V12, V14, V16, V17 and V18 show a higher probability of negative or lower values for fraudulent transactions compared to genuine ones. In contrast, features V4 and V11 display the opposite trend, where fraudulent transactions tend to have higher values. The remaining features appear to have similar distributions across both classes, offering limited discriminatory power.

It is worth noting that due to anonymisation for privacy and security reasons, the original names and meanings of these features are unavailable. Had the features been properly labelled, these distributional differences could have provided even more valuable insights for fraud detection.

Next, we analysed the distribution of the Amount feature across the two classes – Genuine and Fraud – to investigate patterns in transaction values (Figure 5). The plot clearly indicates that fraudulent transactions are predominantly associated with very small amounts. One plausible explanation is that fraudsters intentionally use low-value transactions in an attempt to remain undetected by the account holder or financial institution. In many cases, individuals may overlook minor debits, assuming they are routine charges such as bank fees, interest adjustments or service costs. This tactic reflects a subtle and strategic form of deception, often referred to as the "art of forgery" in fraud detection literature (Chidananda, 2025).

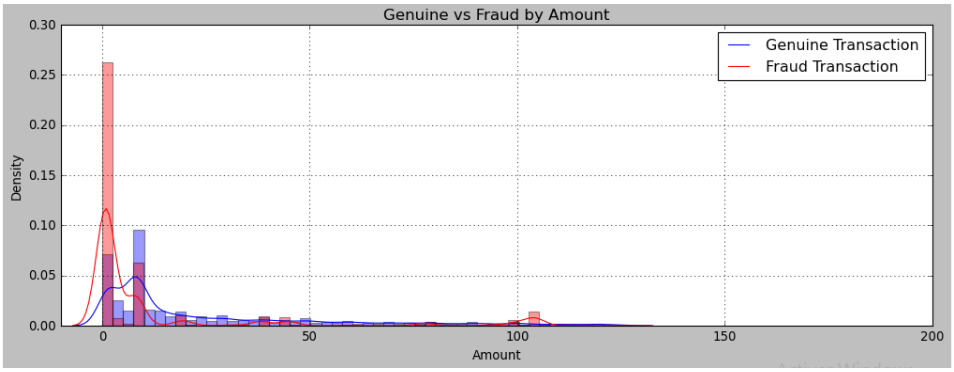

Figure 5. Distribution of Amount by Class label

Source: based on python code GitHub.

Furthermore, we observed that the Amount distributions for genuine and fraudulent transactions are quite similar after anomaly reduction. This suggests that transaction amount alone is not a reliable predictor of fraud, as both classes exhibit overlapping value ranges. In addition, we plotted the two classes against the Time variable to examine temporal patterns. The resulting graph shows that both classes are similarly distributed across the time axis. The Time feature represents the number of seconds elapsed since the first transaction, and the dataset cov-

ers a two-day period. Each day consists of 86,400 seconds, allowing for the conversion of time values into hourly intervals using the formula: (86400/(60 · 60)). For instance, a time value of 50,000 seconds corresponds to approximately 13:00 (1:00 PM). From this analysis, we can infer that transaction frequency tends to peak around midday, particularly near 12:00 PM, for both genuine and fraudulent activities.

3.3.3. Multivariate analysis

To better visualise the class imbalance, a scatter plot was generated to display the distribution of class instances (Figure 6). Genuine transactions were plotted as the majority class, while fraudulent (minority) instances were displayed with five times more visual weight to enhance their visibility on the graph. This scaling was applied solely for visualization purposes and does not affect the data distribution.

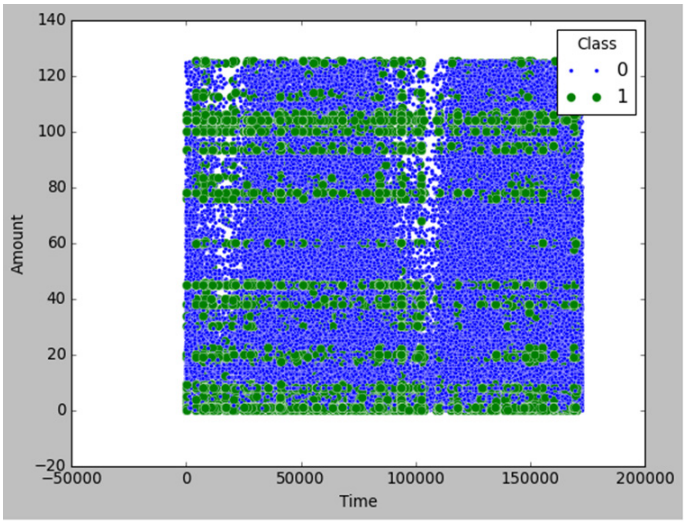

Figure 6. Scatter plot of Class in terms of Amount and Time

Source: based on python code GitHub.

In the scatter plot, the Amount variable is plotted on the vertical axis, while Time is represented on the horizontal axis. Blue dots indicate genuine transactions, whereas green dots represent fraudulent ones. The plot reveals that most fraudulent transactions involve small amounts, often close to $0. Additionally, a noticeable concentration of fraudulent activity appears around the $20 range.

Another key observation is that fraudulent transactions tend to cluster within specific time intervals. Interestingly, the density of fraud appears to increase during periods of high overall transaction activity.

4. Results

In the first step, the data set was randomly divided into a training set (80%) and a test set (20%). Then, in order to balance the data, the following methods were used: SMOTE, random oversampling and random undersampling. Initially, there were 540,099 observations in the source set, of which only 270,049 observations were fraud transactions (Class variable equal to 1), which accounted for 50% of all observations. The training set consisted of 216,040 fraud transactions, and the test set of 54,009. The overall proposed model framework is illustrated in the diagram below, which provides a visual representation of the sequential stages involved in the fraud detection process (Figure 7).

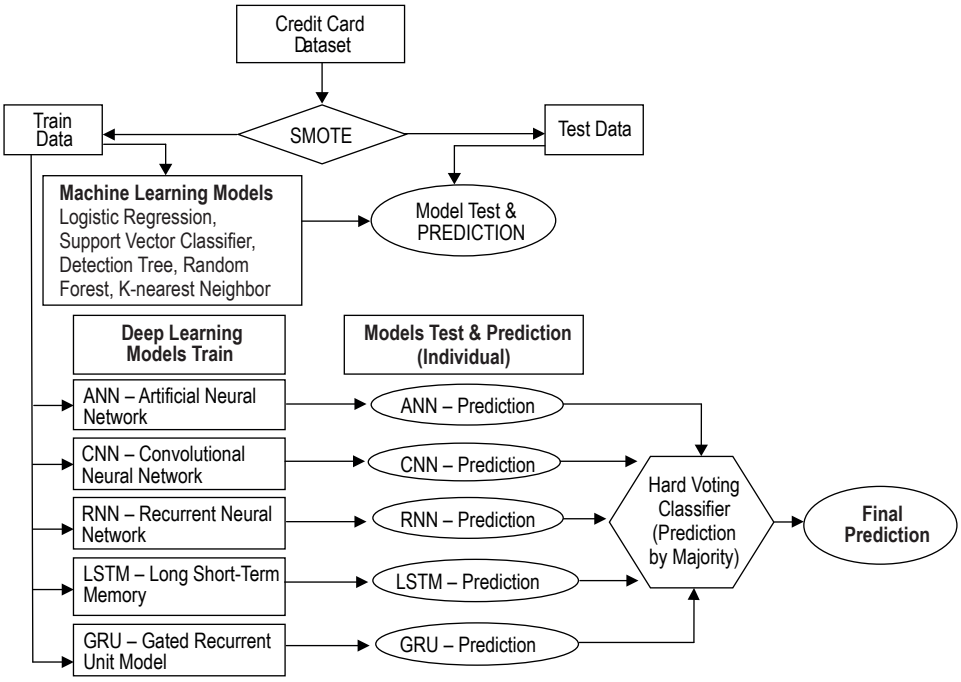

Figure 7. Proposed ensemble hard voting classifier architecture

Source: own study.

Given the high class imbalance in the dataset, balancing techniques are necessary to improve model performance and fairness. This can typically be achieved through either undersampling the majority class or oversampling the minority class. For this project, we opted to perform oversampling of the minority class to retain all valuable information in the dataset.

To address this, we applied the SMOTE (Synthetic Minority Oversampling Technique) method. SMOTE creates synthetic samples of the minority class, rather than duplicating existing ones, thereby enhancing the diversity of the training data. It operates based on the k-nearest neighbors algorithm and constructs new synthetic instances as follows (Chhabra et al., 2024):

- determine the feature vector's closest neighbour,
- calculate the distance between the two sample points,
- multiply the distance by a random number between 0 and 1,
- find a new point on the line segment at the computed distance,
- repeat the process for identified feature vectors.

After scaling the Time and Amount features to ensure uniformity in feature range, we applied SMOTE to oversample the minority class and address the dataset's imbalance. Following this preprocessing step, we evaluated the performance of five different machine learning models for classification purposes during the initial testing phase. These models were selected based on their widespread use and proven effectiveness in fraud detection tasks (Ren, 2023).

The machine learning algorithms used for initial prediction are:

- Logistic Regression,
- Support Vector Classifier (SVC),
- Decision Tree Classifier,
- Random Forest Classifier (maximum depth = 6),
- K-neighbor Classifier (KNN) (k = 5).

We created five neural networks which will be used for prediction individually. Moreover, the hyperparameters and their respective options for this paper are illustrated in Table 4.

In addition, we developed a 4-layer Artificial Neural Network (ANN) for binary classification. The network architecture includes:

- input layer: a 1D array of 30 features (i.e., one observation per transaction),
- three hidden layers: with 6, 20, and 10 units respectively, each using the ReLU activation function,
- output layer: a single neuron with a sigmoid activation function, appropriate for binary output (fraud or genuine).

Table 4. Variations of DL models involved in hyper-parameter tuning

Parameter	Options
Activation function	ReLU, Tanh, Sigmoid
Loss function	RMSE, RMSPE
Neurons	[100, 100, 100, 100, 100, 1]
Learning rate	0.001
Optimiser	Adam
Layers	2, 3, 4
Batch Size	32, 64

Note: Here, [100, 100, 100, 100, 100, 1] represents the number of neurons from the first to the last network layer.

Source: own study.

The model was compiled using binary cross-entropy as the loss function and Adam as the optimizer. For sequential data processing, we also implemented a Recurrent Neural Network (RNN) consisting of three layers: two hidden layers and one output layer. The hidden layers used 32 and 8 units respectively, again with the ReLU activation function. While the loss function remained binary cross-entropy, the RMSprop optimizer was used, which is better suited for handling temporal dependencies in sequential data.

Binary cross entropy distinguishes each of the predicted probabilities to the actual class output, which can be either 0 or 1. The score is then calculated, penalising the probabilities depending on their deviation from the predicted value. This refers to how close or far the value is to the actual value. The negative value of log of corrected predicted probabilities is binary cross entropy. Lastly, we made a hard voting classifier for final prediction. In hard voting (also known as majority voting), every individual classifier votes for a class, and the majority wins. In statistical terms, the predicted target label of the ensemble is the mode of the distribution of individually predicted labels.

Credit card fraud detection is a binary classification. To evaluate the performance of classification models, the confusion matrix is one of the most widely used and effective tools. It provides a structured layout to visualize and assess the predictive outcomes of the model. In binary classification, there are four possible outcomes during prediction:

1. True Positive (TP): the model predicts the correct true label. In our case, this refers to the number of instances where non-fraudulent transactions are correctly identified.
2. True Negative (TN): the model predicts the correct false label. That means, in our problem, the number of fraudulent transactions correctly predicted as fraud.

3. False Positive (FP): the model incorrectly predicts the true label. In the fraud detection context, this occurs when the model predicts a transaction as genuine, but it is actually fraudulent. This is the most critical area where our attention needs to be focused.
4. False Negative (FN): the model predicts a false negative label. In our problem, this means the model predicts a fraudulent transaction, but it is actually genuine. This is our next concern to address.

Mathematically, several terms were computed to summarise the overall model performance based on model prediction, including:

- Sensitivity/Recall/True positive rate, which shows the probability of true positive prediction:

$$\text{Sensitivity/Recall/TPR} = TP/(TP + FN)$$

- Specificity/True negative rate, which shows the probability of true negative prediction:

$$\text{Specificity/TNR} = TN/(TP + FP)$$

- Precision/Positive predicted value, i.e. the probability to predict a positive class among all positive classes:

$$\text{Precision/PPV} = TP/(TP + FP)$$

- Negative predicted value (NPV), which is the opposite of precision; it measures the proportion of correctly predicted negatives among all predicted negatives:

$$TN/(TN + FN)$$

- F1 score: to compute the F1 score we need to take into account both precision and recall. The F1 score can be thought of as a weighted average of the precision and recall values:

$$\text{F1 score} = 2TP/(2TP + FP + FN)$$

- Accuracy: the accuracy of a model is determined by how well it perceives correlations and patterns between variables in a dataset using the input (training) data:

$$\text{Accuracy} = (TP + TN)/(TP + FP + TN + FN)$$

- AUC: a measurement of the complete two-dimensional area beneath the entire ROC curve. ROC curve plotted by TP vertically by TN horizontally.

As we initially applied five machine learning algorithms – Logistic Regression, Support Vector Machine, Decision Tree, Random Forest, and K-Nearest Neighbors – we obtained a confusion matrix for each model. Based on these, we computed the following summary performance metrics (Table 5).

Table 5. Five Machine Learning algorithms

Model	Model Precision	Recall/Sensitivity	F1-Score	Accuracy
LR	1.00	0.9987	1.00	0.9991
SVC	1.00	0.9993	1.00	0.9995
DT	1.00	0.9979	1.00	0.9979
RF	1.00	0.9999	1.00	0.9999
KNN	1.00	0.9997	1.00	0.9998

Source: own study based on python code GitHub.

We observed that the Random Forest algorithm achieved the highest accuracy among the traditional machine learning models, reaching 0.9999. Additionally, the K-Nearest Neighbors (KNN) model also delivered strong performance. Notably, both Random Forest and KNN produced zero false positives, which is a highly desirable outcome in classification tasks – especially in fraud detection. Furthermore, the false negatives for Random Forest were relatively low, with only 15 instances, indicating the model's effectiveness in correctly identifying fraudulent transactions.

We applied five deep learning algorithms – ANN, CNN, RNN, LSTM, and GRU – for classification, followed by the implementation of a hard vote classifier, which outputs the majority prediction among these models. All neural networks were trained and evaluated simultaneously using 50 epochs to ensure consistency in training. To assess the stability and reliability of the models, we ran the classifiers four times and compared the results across runs. To visualise the performance, we generated a heatmap of the confusion matrix based on the computed values. This augmented confusion matrix provides a comprehensive view of the classification performance of each deep learning model, as well as the ensemble classifier we proposed.

The confusion matrix values were machine-scaled and color-coded for visual clarity. Among the individual deep learning models, the Convolutional Neural Network (CNN) achieved the highest accuracy, with a false positive (FP) count of 0 and a false negative (FN) value of 628. Interestingly, our ensemble hard voting classifier demonstrated a significant improvement: it misclassified only 205 transactions as false negatives – meaning the model incorrectly labelled genuine transactions as fraudulent (Table 6).

To more effectively evaluate the performance of the hard voting classifier over multiple runs, we compiled a summary table of misclassifications, including False Negatives (FN), False Positives (FP), and Total Misclassifications (TM), where:

$$TM = FN + FP$$

This approach offers a clearer comparison across models and time, supporting the effectiveness of ensemble learning in minimizing classification errors (Ahmed et al., 2025).

Table 6. Misclassification summary table of all neural networks

Model	ANN			CNN			RNN			LSTM			GRU		
Epochs	FN	FP	TM	FN	FP	TM	FN	FP	TM	FN	FP	TM	FN	FP	TM
50	0	628	3760	2021	5781	87	211	298	87	211	298	298	0	205	205

Source: own study based on python code GitHub.

One intriguing finding was that our proposed classifier consistently outperformed other networks in terms of misclassification. The intended TM value for 50 epochs has been highlighted. CNN has the best overall effectiveness in detecting false negatives. LSTM and GRU performance are very similar. In this model, the RNN performances are the most inconsistent. For example, at epoch 50, there were more than 20,000 incorrect classifications on FN. Due to limited time and resources, we conducted our experiments using a relatively small number of epochs. However, it was discovered that the performance of each model is not dependent on the difference in epochs over a short period of time. As a result, when we combine the model accuracy for machine learning, deep learning (based on 50 epochs) and the proposed classifier, we get Figure 8.

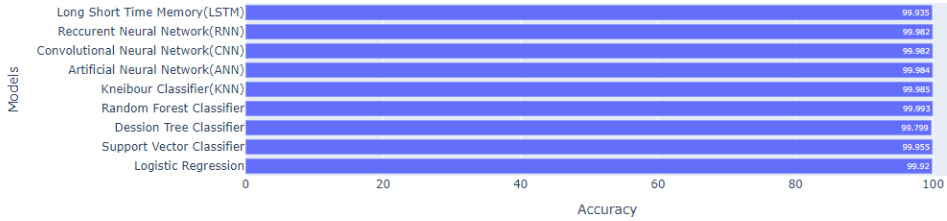

Figure 8. All neural network models and proposed ensemble model accuracy plot

Source: own study based on python code GitHub.

Although the Random Forest classifier with a maximum depth of 6 yielded the highest accuracy among traditional models, performance should not be evaluated solely on accuracy or other numerical metrics. Time complexity is also a critical factor. In our case, training Random Forest and KNN models took over 4 hours, which is comparable to the training time required for the five deep learning models.

When comparing accuracy across epochs, our proposed ensemble classifier achieved the highest overall accuracy, outperforming the individual neural net-

works. However, when isolating performance by epoch-based accuracy, we observed that the GRU model slightly outperformed the ensemble in specific cases.

To better understand these nuances, we proceeded with a deeper evaluation of each model's performance metrics.

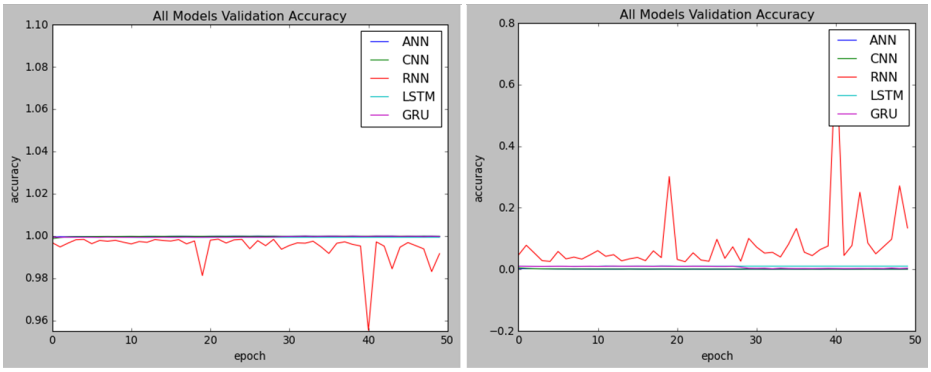

Figure 9. Accuracy and loss curve for all models together

Source: based on python code GitHub.

It was evident from the start of the epochs that RNN performance was substandard, although the other four networks produced a similar curve pattern for accuracy and loss. Furthermore, because the LSTM and GRU architectures are similar by nature, and we selected similar hyper parameters in our model, the accuracy and loss for both networks are extremely similar (Figure 9). When we increase epochs the pattern is still identical for these five networks.

Conclusions

This research highlights the evolving challenges and opportunities in combating credit card fraud within the EU banking sector. The study underscores the critical role of machine learning (ML) and deep learning (DL) in enhancing fraud detection while navigating the constraints imposed by PSD2, GDPR and strong customer authentication (SCA). Key findings reveal that Card-Not-Present (CNP) fraud remains the dominant threat, accounting for 50% of cases, while emerging risks such as synthetic identity fraud and AI-driven scams demand innovative countermeasures.

Credit card fraud detection remains a challenging problem due to the complexity of accurately identifying fraudulent transactions. In this case, the dataset lacked

detailed descriptions, which limited our ability to perform optimal feature selection. Even a single irrelevant feature can significantly impact model performance.

Interestingly, the proposed ensemble hard voting neural network classifier sometimes exhibited lower accuracy than individual neural networks. This outcome likely stems from certain difficult-to-classify observations where the true label was ambiguous or challenging to predict. Since the hard voting classifier relies on the majority decision of all models, it struggled when most individual networks failed to correctly classify these ambiguous cases, resulting in misclassification of some fraudulent transactions.

There is significant potential for further research in applying ensemble techniques to neural networks to address this challenge. Due to limited time and resources, we were unable to perform extensive hyperparameter optimization. Future work could explore different parameter settings and build neural network ensembles based on the highest accuracy scores. For example, assembling more than ten neural networks – such as multiple GRU, LSTM or CNN models – and combining their predictions could improve performance. Additionally, we did not employ k-fold cross-validation during training, as the training accuracy scores were already within an acceptable range. However, incorporating k-fold validation could enhance model robustness by providing a better assessment of generalisation performance.

Hyperparameter tuning remains a critical step to develop a more robust model architecture. On the data balancing front, we applied SMOTE to oversample the minority class, but there are at least six other oversampling techniques available in the research community, alongside various undersampling methods. Exploring these could yield better balance and performance. Moreover, ensemble methods such as AdaBoost show promise as powerful alternatives. In an optimised multi-neural network design, AdaBoost could be combined with GRU as the base learner. Finally, integrating both machine learning algorithms and neural networks within ensemble frameworks might further enhance classification effectiveness.

Based on the results presented above, we cannot confirm our hypothesis (H1) because our proposed solution outperforms other deep learning models, as demonstrated by these experimental results.

In future work, we will evaluate the model's scalability by testing it on larger and more diverse real-world datasets of EU banks. We aim to assess the model's potential deployment within real-world financial infrastructures, analysing its adaptability to live transactional data and integration with existing fraud detection pipelines. Ensuring robustness across various financial environments will be a key focus.

References

BIS. (2024). *Annual economic report*. Bank for International Settlements. https://www.bis.org/publ/arpdf/ar2024e.pdf

Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical Science*, *17*(3), 235–255. https://doi.org/10.1214/ss/1042727940

Brause, R., Langsdorf, T. & Hepp, M. (1999). Neural data mining for credit card fraud detection. In *Proceedings 11th International Conference on Tools with Artificial Intelligence* (pp. 103–106). https://doi.org/10.1109/TAI.1999.809773

Buzzard, J. (2022). *2022 Identity fraud study: The virtual battleground*. https://javelinstrategy.com/2022-Identity-fraud-scams-report

Chaudhari, A., & Kaur, M. (2025). Enhancing global banking security: A novel approach integrating federated learning and CNN-GRU for effective anti-money laundering measures. *Journal of Information Systems Engineering and Management*, *10*(32s). 1053–1065. https://doi.org/10.52783/jisem.v10i32s.5449

Chhabra, R., Goswami, S. & Ranjan, R. K. (2024). A voting ensemble machine learning based credit card fraud detection using highly imbalance data. *Multimed Tools Appl*, *83*, 54729–54753. https://doi.org/10.1007/s11042-023-17766-9

Chidananda, A. (2025). *Deep learning for fraud detection in financial transactions using CNN-LSTM hybrid and GRU Model* [Master thesis]. California State University. https://scholarworks.calstate.edu/concern/theses/qf85nm65z

Detura, R., Ioshiura, C., Murphy, A., Richardson, B., Scheurle, S., Schweikert, E., & Vancauwenberghe, M. (2022, November 8). *A new approach to fighting fraud while enhancing customer experience*. McKinsey & Company. https://www.mckinsey.com/capabilities/risk-and-resilience/our-insights/a-new-approach-to-fighting-fraud-while-enhancing-customer-experience

ECB. (2023, May). *Annual report 2022*. European Central Bank. https://www.ecb.europa.eu/pub/pdf/annrep/ecb.ar2022~8ae51d163b.en.pdf

ECB. (2025, April). *Annual report 2024*. European Central Bank. https://www.ecb.europa.eu/pub/pdf/annrep/ecb.ar2024~8402d8191f.en.pdf

Ghosh, S., & Reilly, D. (1994). Credit card fraud detection with a neural-network. In *Proceedings 27th Hawaii International Conference on System Sciences: Decision support and knowledge-based systems* (vol. 3, pp. 621–630). https://doi.org/10.1109/HICSS.1994.323314

Khanda, H. A., Stefan, A., Yuhong, Li, & Ali, M. S. (2025). A credit card fraud detection approach based on ensemble machine learning classifier with hybrid data sampling. *Machine Learning with Applications*, *20*. https://doi.org/10.1016/j.mlwa.2025.100675

Kolli, C. S., Tatavarthi, U. D., & Raju, D. V. N. (2023). *Fraud detection in banking: AI strategies for financial institutions: Reduce complexity, increase productivity*. Lap Lambert Academic Publishing.

Mienye, I. D., & Swart, T. G. (2024). A hybrid deep learning approach with generative adversarial network for credit card fraud detection. *Technologies*, *12*(10), 186. https://doi.org/10.3390/technologies12100186

Misra, S., Thakur, S., Ghosh, M., & Saha, S. K. (2020). An autoencoder based model for detecting fraudulent credit card transactions. *Procedia Computer Science*, *167*, 254–262. https://doi.org/10.1016/j.procs.2020.03.219

Moturi, S. R., Matta, R, Pavurala, P. K, Kolli, S. K, & B. Nandan K. (2024). Optimizing credit card fraud detection using deep learning by smote-enn technique. *International Refereed Journal of Engineering and Science (IRJES)*, *13*(2), 190–200. https://www.irjes.com/Papers/vol13-issue2/1302190200.pdf

Naas, M. N., & H. Zouaoui (2024). Forecasting foreign exchange rate volatility using deep learning: Case of US dollar/Algerian dinar during the COVID-19 pandemic. *Research Papers in Economics and Finance*, *8*(1), 91–114. https://doi.org/10.18559/ref.2024.1.1172

Nilson Report. (2020). https://nilsonreport.com/newsletters/1187/

Ren, Y. (2023). Application of machine learning algorithms in detecting credit card fraud: A comparative analysis. *Highlights in Business, Economics and Management*, *21*, 733–739. https://doi.org/10.54097/hbem.v21i.14753

Sulaiman, S. S., Nadher, I., & Hameed, S. M. (2024). Credit card fraud detection using improved deep learning models. *Computers, Materials & Continua*, *78*(1), 1049–1069. https://doi.org/10.32604/cmc.2023.046051

Tayebi, M., & El Kafhali, S. (2025). A novel approach based on XGBoost classifier and Bayesian optimization for credit card fraud detection. *Cyber Security and Applications*, *3*. https://doi.org/10.1016/j.csa.2025.100093

Vadisena, V. K. R., Radha, V. K. R., Masthan, S. K. M., Balaji, K., Suresh, K. M., & Kolli C. S. (2024). Deep learning-based credit card fraud detection in federated learning. *Expert Systems with Applications*, *255*(A). https://doi.org/10.1016/j.eswa.2024.124493

Wahab, F., Khan, I., & Sabada, S. (2024). Credit card default prediction using ML and DL techniques. *Internet of Things and Cyber-Physical Systems*, *4*(1), 293–306. https://doi.org/10.1016/j.iotcps.2024.09.001

Zareapoor, M., & P. Shamsolmoali, P. (2015). Application of credit card fraud detection: Based on bagging ensemble classifier. *Procedia Computer Science*, *48*, 679–685. https://doi.org/10.1016/j.procs.2015.04.201

Zouaoui, H., & Naas, M. N. (2023). Option pricing using deep learning approach based on LSTM-GRU neural networks: Case of London stock exchange. *Data Science in Finance and Economics*, *3*(3), 267–284. https://doi.org/10.3934/DSFE.20230160