Erasmus University Rotterdam

Erasmus School of Economics

BSC^2 Thesis Econometrics and Economics

Tackling Credit Card Fraud: Evaluation of Classification Models for Imbalanced Data.

Lakshita Bhatti (575717)

Supervisor:	Dr Katherin Gruber
Second assessor:	Pieter Schoonees
Date final version:	1st July 2024

Abstract

In today's cashless society, credit cards are essential to the global economy. With the digitisation of money and financial transactions, banks and other financial institutions are vulnerable to fraudsters who now have numerous ways to perpetrate crime from behind a screen, anywhere in the world. The constant evolution of fraud schemes exposes these institutions to significant financial losses, hence it is imperative that the have strong fraud detection systems. Thus, this thesis seeks to assess different machine-learning classification models for detecting credit card fraud on an imbalanced dataset. This paper employes models including Random Forest, Adversarial Random Forest, One-Class SVM, and Random Forest Quantile Classifier. They are evaluated using various performance metrics like Accuracy rate, PR Curve, AUC-ROC, False Negative Rate (FNR), False Positive Rate (FPR), and F1 Score. The field of credit card fraud detection frequently encounters challenges such as limited access to datasets and a significant imbalance in the number of fraudulent and non-fraudulent transactions. Keeping this in mind, the methods mentioned above are specifically employed to test their effectiveness in identifying fraud in light of the class imbalance problem. Overall, the findings show that the Random Forest model outperformed the others, effectively defending against adversarial attacks. The One-Class SVM, while conservative, showed the highest false negative rate, whereas the Random Forest Quantile Classifier had the lowest false positive rate. Despite these strengths, both these methods showed room for improvement in other performance metrics.

1 Introduction

As our society becomes increasingly cashless, the rise in credit card usage marks an unprecedented transformation in global spending habits. Credit cards have quickly become the preferred payment method for both online and in-store purchases alike. Furthermore, a higher reliance on credit cards for online transactions has been spurred by the growing E-commerce sector. According to (Zareapoor, K.R. & Alam, 2012)), physical transactions are ones in which the buyer and seller deal directly where it is necessary to use a physical card at the point of sale. Conversely, online transactions, which are carried out either over the telephone or via the Internet, have become more prevalent than ever (Zareapoor et al., 2012). Undoubtedly, the advent of credit cards has streamlined online transactions, making them not only seamless but also more accessible and convenient (Adewumi & Akinyelu, 2016).

Therefore, in an era where one's wallet is more likely to hold cards than cash, the invisible threat of credit card fraud looms large; challenging both consumers and companies alike. Card Not Present (CNP) fraud is a type of credit card fraud that involves unauthorised and unlawful use of physical cards or card details without the consent of the cardholder (Zareapoor et al., 2012). Payments made via the phone, mail, or the internet accounted for a staggering 80% of the value of card fraud in 2019 (European Central Bank, 2023). This type of fraud noticeably poses significant challenges for banks and financial institutions. In stark contrast, fraudulent transactions at physical point-of-sale (POS) terminals, including face-to-face payments at retailers or restaurants, and at automated teller machines (ATMs), comprised only 15% and 5% of the total value of card fraud, respectively (European Central Bank, 2023). Hence, sensitive credit card information is particularly vulnerable on unsecured online platforms and web pages.

Additionally, identity theft schemes may expose or steal credit and debit card numbers, allowing fraudsters to access these details illicitly (Sulaiman, Schetinin & Sant, 2022).

According to the European Central Bank (ECB), the total losses from card fraud in the Single Euro Payments Area (SEPA) for cards issued were €1.53 billion in 2021, out of €5.40 trillion in total card transactions. Despite reaching its lowest levels in recent years due to regulatory efforts by the ECB, the financial aftermath for merchants is still severe. They face a 'double jeopardy' of absorbing the costs of lost goods and contending with chargeback fees, all while balancing on the brink of potentially having their merchant accounts shut down (Bhatla, Prabhu & Dua, 2003). Evidently, fraudulent transactions represent a minimal portion of total transactions, accounting for just 0.028% in 2021 (European Central Bank (2023)). This small ratio makes identifying fraudulent transactions accurately and efficiently a daunting task. As such, it becomes crucial to develop robust methods capable of discerning these 'rare' fraudulent activities amidst billions of legitimate transactions.

Adding on to this, 800,944 cybercrime complaints totalling $10.3 billion in losses have been filed to the Internet Crime Complaint Center (IC3) (IC3, 2022) in 2022. Credit card scammers stand out in this intricate web of cybercrime because of their diverse tactics and backgrounds. These fraudsters can be broadly categorised into three types: buyers of stolen credit card information, black hat hackers, and physical credit card thieves. Buyers typically lack advanced technical skills and purchase stolen card details from illegal websites to make unauthorised transactions. In contrast, black hat hackers are highly skilled in programming and networking. They breach security systems to steal sensitive information for personal gain, often selling it to less skilled fraudsters. Physical thieves steal cards directly, recording details to use for online purchases (Akhilomen, 2013).

These cybercriminals continuously target digital systems worldwide, attacking both large corporate networks and consumer's personal devices. Consequently, the sheer volume of suspected fraud cases makes manual checking impractical, necessitating the use of machine learning models in fraud detection systems. Fraud detection involves identifying rare events, a challenge described using terms such as outlier analysis, anomaly detection, exception mining, and mining imbalanced data (Sorournejad, Zojaji, Atani & Monadjemi, 2016). Given the rarity and subtlety of fraudulent activities, effective machine-learning models are crucial for distinguishing between normal and anomalous transactions. Classification models, in particular, are essential for designing credit card anomaly detection systems. These models learn from a set of training samples to classify unseen data into normal or fraudulent classes.

Nonetheless, these classification models are susceptible to adversarial attacks, which exploit weaknesses in the way the model makes decisions. Adversarial attacks entail carefully modifying input data in order to trick the model; this is often done by making subtle, imperceptible alterations to the input that lead to the model classifying data incorrectly. These attacks may originate from a variety of sources like malicious hackers and fraud artists looking to bypass security measures. Such attacks have the potential to cause major financial losses for businesses by making the model mistakenly categorise fraudulent transactions as genuine. As fraudulent tactics evolve globally, fraud detection systems are swiftly advancing to keep pace.

The integration of artificial intelligence (AI), particularly through machine learning and data mining, has proven effective in identifying fraud within the financial sector. The European Central Bank (2023) reports that four out of ten merchants make use of machine learning to enhance their fraud management and payment routing systems.

In light of the background and the ever-evolving challenges in detecting credit card fraud, this paper aims to analyse various classification methods to determine the most effective models for fraud detection on imbalanced datasets. Additionally, the paper examines the performance of these models when subjected to adversarial attacks by cybercriminals, contributing to the existing literature on enhancing the robustness of fraud detection systems.

In order to fulfil the aim of this research, various methods such as the Random Forest (RF), Adversarial Random Forest, One-Class Support Vector Machine, and the Random Forest Quantile Classifier (RFQC) are employed in this study. Due to the advancing dangers mentioned above, this study also examines various threat models characterised by attackers with different levels of knowledge. Specifically, it focuses on Black-Box attacks, where the attacker has no knowledge of the target system, and White-Box attacks, where the attacker has complete knowledge of the systems. This study employs a machine learning-based approach to conduct evasion attacks against fraud detection methods, simulating the behaviour of different types of fraudsters under varying degrees of knowledge. To assess the performance of the methods, various metrics are employed, including accuracy, precision-recall (PR) curve, sensitivity, specificity, false positive rate (FPR), false negative rate (FNR), precision, F1 score, and area under the receiver operating characteristic (AUC-ROC) curves. The results reveal that the RF model delivers the most outstanding performance, showcasing high classification accuracy even under adversarial attack conditions. This model effectively differentiates between fraudulent and non-fraudulent transactions. Moreover, the One-Class SVM and RFQC also demonstrate competitive and promising results. However, their performance may be slightly hindered by the inherent class imbalance in the dataset.

The rest of the paper is organised as follows: Section 2 offers a comprehensive literature review, highlighting the rationale behind selecting the various models for studying fraud detection systems. Section 3 describes the data used in this research. Section 4 provides an in-depth explanation of the methodology, offering readers detailed insights into the models employed. Section 5 presents the results of the study. Finally, Section 6 discusses the implications of the findings, addresses the limitations of this research, and suggests avenues for future investigation.

2 Literature Review

Prior to the advent of artificial intelligence, the battle against credit card fraud was waged with outdated tools, relying on manual reviews and rigid rule-based systems. These traditional methods used predefined rules to flag suspicious transactions, forming the initial foundation for fraud detection. However, they were often plagued by high false-positive rates and were vulnerable to the ever-evolving strategies of fraudsters. As the complexity and volume of transactions increased, the necessity for more sophisticated and scalable solutions became increasingly apparent.

Machine learning has since emerged as a powerful tool in the fight against credit card fraud. In R. Chen, Chiu, Huang and Chen's (2004) study, the authors introduced an innovative approach to detecting credit card fraud using a questionnaire-responded transaction model based on Support Vector Machines (SVM). Presented at the IDEAL2004 conference, the research integrated user responses with transaction data which showcased significant improvement in fraud detection accuracy compared to traditional rule-based systems. This early application of machine learning in fraud detection highlights the transition from static rules to dynamic and data-driven methods, underscoring the potential of SVM in enhancing fraud detection capabilities.

Maes, Tuyls, Vanschoenwinkel and Manderick (2002) investigated credit card fraud detection using Bayesian networks and Artificial Neural Networks (ANN). They found that neural networks provided superior accuracy in detecting fraudulent transactions, while Bayesian networks offered better interpretability and faster processing times. However, the study noted limitations in the neural network approach, including the need for extensive training data and higher computational resources. This emphasised the trade-off between accuracy and resource efficiency in different machine-learning techniques for fraud detection.

With these innovative methods paving the way, the exploration of advanced classification models marked the next leap forward, promising even greater potential to refine and strengthen fraud detection systems. Shen, Tong and Deng (2007) explored the application of classification models in credit card fraud detection, specifically harnessing decision trees, neural networks, and logistic regression. Their study evaluated these models on their ability to detect fraudulent transactions and found that neural networks demonstrated the highest accuracy. However, decision trees provided quicker processing times and a balance of accuracy and interpretability, while logistic regression offered a good trade-off between model simplicity and performance.

Adding to these findings, a recent 2023 study by Yundong, Zhulev and Ahmed focused on credit card fraud detection using Logistic Regression and Random Forest (RF) models. RF, an ensemble method that aggregates the results of multiple decision trees, demonstrated superior performance. The study found that Random Forest achieved higher accuracy in identifying fraudulent transactions and significantly reduced false positives compared to Logistic Regression. This enhanced performance is due to RF's capability to capture complex patterns and interactions within the data, making it a highly effective tool for fraud detection.

Drawing inspiration from Yundong et al. (2023), this research further explores RF due to their strong performance in classification tasks. Adversarial attacks have become a focal point in machine learning research, demonstrating their potential to reveal weaknesses in models that appear robust. These attacks, which involve making minor, deliberate changes to input data, show that even highly accurate models can be tricked into making errors. This is especially important in fraud detection, where failing to identify fraudulent transactions can have serious consequences. Ding et al. (2019) delve into the susceptibility of neural networks to adversarial examples, illustrating that even highly accurate models can be deceived by carefully crafted inputs. This vulnerability is particularly concerning in the context of fraud detection, where the consequences of missed fraudulent transactions can be severe.

While the majority of existing research, such as Ding et al. (2019) and Huang, Menkovski, Pei and Pechenizkiy (2022), has focused on the application of adversarial attacks in the context of image recognition, there is a notable gap in applying these techniques to financial datasets. This thesis strives to bridge this gap by leveraging adversarial attacks on a highly imbalanced credit card fraud dataset. This study makes use of both white-box attacks (Huang et al., 2022), where the attacker has full knowledge of the model, and black-box attacks (J. Chen, Jordan & Wainwright, 2020), where the attacker has no such knowledge. The intent of doing so is to provide a comprehensive evaluation of the RF model's resilience under different adversarial conditions, simulating real-world threat scenarios in the financial sector.

Meinshausen (2006) proposed Quantile Regression Forests (QRFs) as an improvement over regular Random Forests in their seminal work. Unlike conventional Random Forests, which focus mostly on predicting the mean outcome, QRFs estimate the response variable's whole conditional distribution. Because of its capability to estimate conditional quantiles in a non-parametric and accurate manner, QRFs are especially useful for tasks that require a thorough comprehension of the distribution of data. This includes determining prediction intervals and spotting outliers. Since QRFs store all of the observations at each leaf node rather than just the mean, they can retain more information at each node than standard RFs. This method works well for predictive modelling as it enables the construction of prediction intervals and provides a comprehensive way to understand prediction variability.

However, while Meinshausen's QRFs excel at capturing response variable distributions in detail, they are best suited for continuous outcomes and cannot directly handle binary classification problems. O'Brien and Ishwaran (2019) expanded on this idea to develop the Random Forest Quantile Classifier (RFQC), which handles categorical outputs effectively in order to get over this limitation. This modification is particularly helpful in the detection of credit card fraud, a binary classification issue with significantly unbalanced data. The RFQC offers a balanced and reliable prediction model, essential for precisely identifying fraudulent activity by concentrating on minimising the false positive rate while retaining high sensitivity.

3 Data

A significant challenge in the research of data mining applied to fraud detection is the limited availability of real-world data for conducting studies (Phua, Lee, Smith-Miles & Gayler, 2010). This is particularly true for credit card fraud datasets due to confidentiality concerns. The dataset consists of credit card transactions by European cardholders from September 2013. It was collected and analysed through a research collaboration between Worldline and the Machine Learning Group at ULB (Université Libre de Bruxelles), focusing on big data mining and fraud detection (ULB, 2018). The data spans two days and includes 284,807 transactions, of which 493 are fraudulent, resulting in a highly imbalanced dataset with frauds representing only 0.172%.

The dataset contains 31 numerical input variables that have been transformed using Principal Component Analysis (PCA), except for two variables: Time and Amount. The 'Time' variable measures the seconds since the first transaction, and the 'Amount' variable displays

the transaction value, which is useful for cost-sensitive learning. The target variable, 'Class' (renamed to isfraud), indicates whether the transaction is classified as fraud (1) or non-fraud (0). The original names of the variables/features and additional background information are not provided due to confidentiality issues. Therefore, these features are named as V1,V2...V28. A glimpse into the details of the variables can be seen in the table containing the descriptive statistics.

Table 1: Descriptive Statistics

Variable	**Min**	**1st Quantile.**	**Median**	**Mean**	**3rd Qu.**	**Max**
Time	0.00	54205	84693	94811	139298	172792
Amount	0.00	5.60	22	88.47	77.51	25691.16
isfraud	0.00	0.00	0.00	0.0017	0.00	1.00
V1	-56.41	-0.92	0.02	0.0059	1.32	2.45
V12	-18.68	-0.41	0.14	-0.0007	0.62	7.85
V14	-19.21	-0.43	0.05	0.0003	0.49	10.53
V17	-25.16	-0.48	-0.07	0.0002	0.39	9.25
V28	-15.43	-0.05	0.01	0.0006	0.08	33.85

Note: This table displays the descriptive statistics of selected variables. Due to confidentiality reasons, the details of the features are not revealed. Thus, only a few variables are included in this table.

Referring to Table 1 , **Time** represents the seconds elapsed between the first and subsequent transactions, ranging from 0 to 172,792 seconds, with an average of 94,811 seconds, indicating most transactions occur within this time frame. **Amount**, which is the value of the transaction, spans from €0.00 to €25,691.16, with a median of €22.00 and a mean of €88.47, showing that most transactions are small in value. **isfraud** is a binary indicator, representing whether a transaction is fraudulent or not, with a mean of 0.0017, highlighting the significant class imbalance with very few fraudulent transactions. V1, V12, V14, V17, and V28 are the features from the PCA transformation, each centred around zero, reflecting the standardised nature of these transformed variables.

Upon inspecting the dataset, it was found that there are no missing values. However, the dataset did contain some duplicate entries, which were removed prior to running any methods, to ensure the integrity of the data. For the purpose of this research, the 'Time' variable was excluded from the analysis, as it was determined to lack predictive value and only contributed to the noise. Consequently, the remaining 30 variables were used to train the machine-learning models.

Moreover, the correlation matrix depicted in Figure 1 provides a detailed view of the linear relationships between various features in the dataset. The matrix uses a colour gradient to represent correlation coefficients, with values ranging from -1 to 1. A value of 1, represented by the yellow diagonal, indicates perfect positive correlation where each variable is perfectly correlated with itself. Most features exhibit weak correlations with one another, as seen by the dominance of dark blue hues, suggesting low multicollinearity.

Figure 2 illustrates the imbalance present in the dataset. The methodologies employed in this research are specifically designed to address this imbalance. In fraud detection datasets,

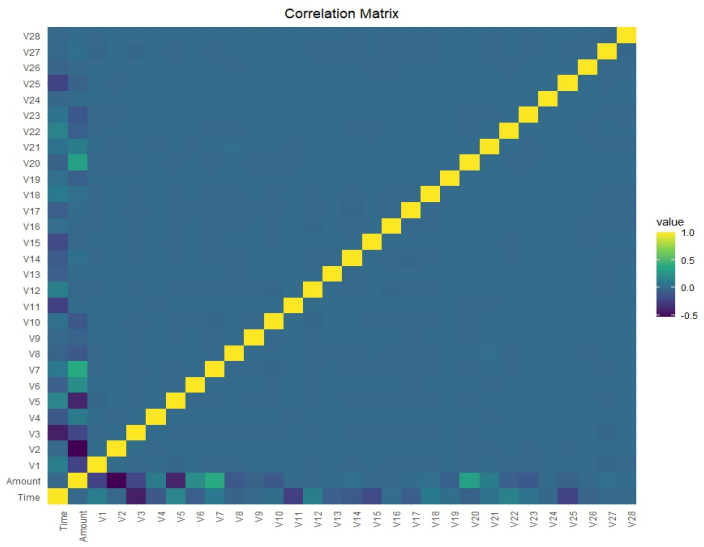


Figure 1: Correlation Matrix of Variables: Each cell shows the correlation coefficient between two variables, with values ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation). The diagonal yellow line with squares represents the correlation of each variable with itself, which is equal to 1.

the vast majority of instances represent non-fraudulent behaviour, labelled as 0. There are significantly fewer instances that are labelled as fraudulent due to the difficulty in accurately identifying and classifying such behaviour. Therefore, this thesis focuses on learning from such a highly imbalanced dataset, emphasising the importance of identifying rare fraudulent events.

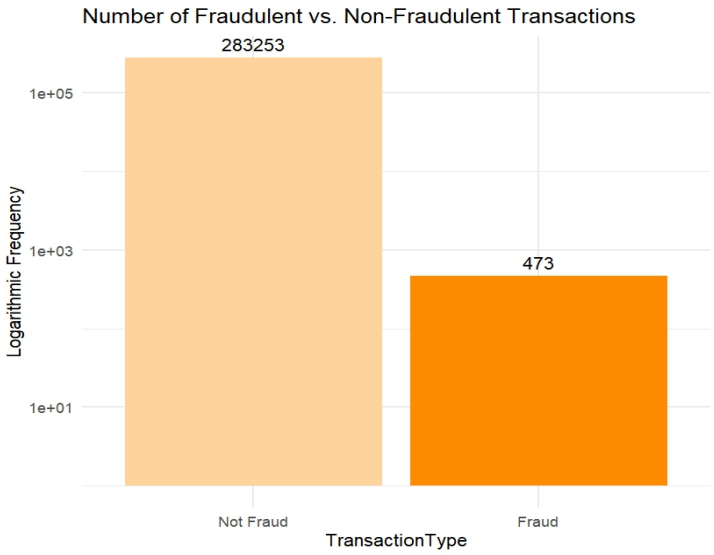


Figure 2: Fraudulent vs Non-Fraudulent Transactions. **Important to know:** This bar chart has been constructed after cleaning the data. Number of normal transactions = 283,253 , Number of fraud transactions = 473 (0.167% of total transactions)

4 Methodology

A typical fraud detection system (FDS) incorporates several layers of control, which can be either automated or supervised by humans (Carcillo et al., 2018). The automated layer often uses machine learning algorithms to create predictive models from labelled transaction data. Over the past decade, significant advancements in machine learning for credit card fraud detection have produced various techniques such as: supervised, unsupervised, and semi-supervised techniques (Sethi & Gera, 2014). Section 4.1 further explains this in detail and provides the methods used in this research paper.

4.1 Supervised and Unsupervised Learning Methods

Supervised learning, or predictive learning makes use of past examples to forecast the category of unknown objects. This technique makes use of historical data, sometimes known as 'training data', which contains instances of both legitimate and fraudulent transactions. In this study, supervised learning is employed to detect fraud by analysing various algorithms. This approach, often called 'misuse detection' (Sorournejad et al., 2016), is chosen for its proven reliability in identifying known fraud patterns. Supervised techniques leverage labelled past transactions to develop a fraud prediction model that estimates the probability of a new transaction being fraudulent.

To accurately predict fraudulent transactions, this paper explores supervised classification models, namely Random Forests and Quantile Classifiers. These models, a subset of supervised learning techniques, are specifically designed to categorise data into predefined classes. By training on a labelled dataset, supervised classification models learn the relationship between input features and target labels (fraud or not), enabling accurate prediction of class labels for new, unseen data.

The drawback of supervised classifiers is their reduced performance as a result of class imbalance. Conversely, unsupervised outlier detection techniques do not require labelled transactions. Instead, they focus on characterising the overall data distribution. These techniques operate on the assumption that outliers within the transaction distribution are indicative of fraud. Consequently, they can detect previously unseen types of fraud, as they do not depend on historical labels identifying fraudulent transactions.

This thesis focuses on the integration of both supervised and unsupervised learning methods to address the challenge of an imbalanced labelled dataset. Since unsupervised methods do not require labels, they were first trained on data that had been filtered to exclude the fraud class. This resulted in a dataset that was made up entirely of legitimate transactions. Then, using these trained models, fraud was identified in a separate set of credit card transactions. This approach is similar to One-Class Classifiers (OCC) (Perera, Oza & Patel, 2021), where only data from a single positive class is used during training. However, because fraudulent transactions were specifically filtered out in the training phase, it does not strictly qualify as an unsupervised method.

This thesis is unique in the fact that it simultaneously evaluates the efficacy of supervised methods in fraud detection and the accuracy of unsupervised methods on an imbalanced, labelled dataset. The goal of merging both these strategies is to improve fraud detection abilities and offer a thorough evaluation of both methods' performance in these particular conditions. The remainder of Sec.4 will provide an in-depth exploration of the methodologies employed in this study.

4.2 Random Forest

Breiman (2001) developed Random Forests (RF), an ensemble supervised learning methodology that combines numerous decision trees to improve classification accuracy. The first step in the procedure is bootstrap sampling, which creates several training datasets by selecting samples at random from the original dataset and replacing them. Each decision tree in the forest is trained on a different subset of the data, introducing variability among the trees.

During the construction of each tree, a random subset of features is considered at each node to determine the best split, rather than evaluating all features. This random feature selection reduces the correlation among trees and improves the robustness of the model. The trees are developed to their full depth without any pruning, and the Gini impurity is used to identify the best split at each node. Below is a visualisation of one of the decision trees used in the ensemble of the RF model. Figure 3 shows a decision tree where each node represents a split based on a feature value, and each leaf node represents a class prediction.

The visualisation helps in understanding the hierarchical decision-making process of the RF model. Each split is based on the feature that best separates the classes at that level. For instance, the first split in the tree is based on feature V17, and subsequent splits further refine the classification based on other features. This detailed view helps interpret the model's behaviour and identify the most influential features for fraud detection.

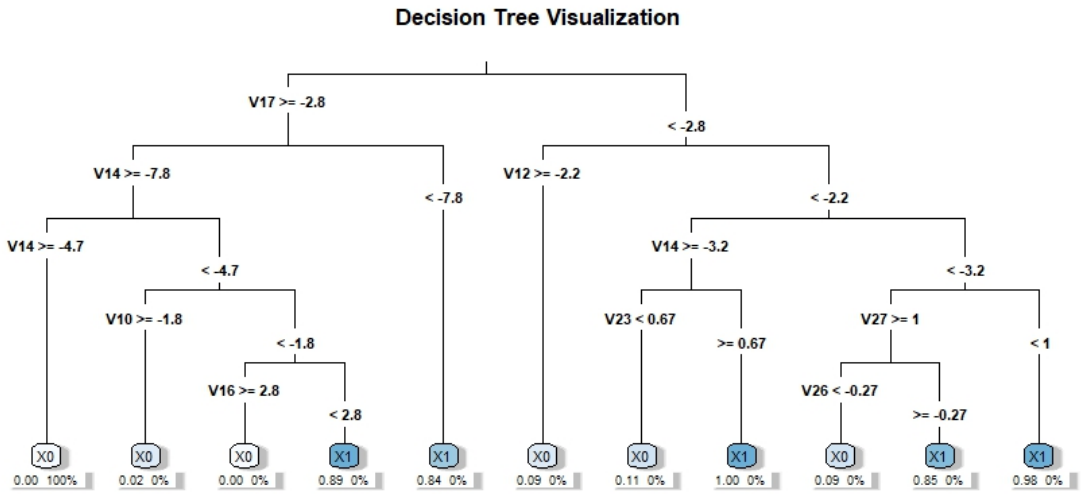


Figure 3: Decision Tree Visualisation. This figure illustrates how the Random Forest (RF) model splits the dataset at each node based on feature values. The paths from the root to the leaves show the conditions leading to the final classification, with leaf nodes representing the predicted class and the proportion of samples classified.

The RF algorithm works in the following way for binary classification tasks such as identifying fraudulent transactions: Each tree is trained on different subsets of data, and then all of the trees collectively generate predictions. For a new transaction, each tree in the forest votes on whether the transaction is fraudulent or not. The final prediction is made based on the majority vote, where the class with the most votes is selected as the final prediction (Breiman, 2001).

An individual tree's strength and correlation are important factors that affect how well the RF model performs. The model's generalisation error measures how well the procedure performs on new, untested data. It decreases as more trees are added to the forest. This is because each additional tree contributes to reducing the overall variance of the model. Thus, as the number of trees increases, the model's predictions become more stable and the error rate converges to a fixed value ensuring that overfitting does not pose as a problem. The margin function, which measures the confidence in the model's classification, is defined as:

$$\text{mg}(X, Y) = \frac{1}{K} \sum_{k=1}^{K} I(h_k(X) = Y) - \max_{j \neq Y} \frac{1}{K} \sum_{k=1}^{K} I(h_k(X) = j), \tag{1}$$

where I is the indicator function that returns 1 if the condition is true and 0 otherwise. $h_k(X)$ is the prediction of the k-th tree, X is the input vector, and Y is the true class label. The margin measures the difference between the average number of votes for the correct class and the highest average number of votes for any incorrect class. A higher margin indicates greater confidence in the model's classification decision.

The generalisation error PE^* is the probability that the margin is less than zero, given by:

$$\mathrm{PE}^* = P_{X,Y}(\mathrm{mg}(X,Y) < 0), \tag{2}$$

The performance of a RF depends on the strength of individual trees and their correlation, which is expressed as :

$$\mathrm{PE}^* \leq \frac{\bar{\rho}(1-s^2)}{s^2}, \tag{3}$$

where ρ is the average correlation between trees, and s is the strength of the trees.

To implement the RF algorithm in this research, the dataset was split into training (70%) and testing sets (30%). The model was created using R's *randomForest* package. The model was configured to use 100 trees ($ntree = 100$), to strike a balance between accuracy and computational efficiency. Every split (*mtry*) was limited to a maximum of five features. This was determined by taking the square root of the total number of features, which is the default value for classification tasks. To avoid overfitting and guarantee computational viability, the minimum size of terminal nodes (*nodesize*) was set to 10 and the maximum number of terminal nodes (*maxnodes*) in each tree was restricted to 30.

4.3 Adversarial Random Forest

In order to trick machine learning models into producing inaccurate predictions, adversarial attacks entail subtly altering the input data. These perturbations can drastically change the model's output, even though they often vary so little that observers cannot see them. Adversarial attacks are especially helpful in the context of this research since they aid in assessing and improving the strength of fraud detection systems. Fraud detection models are crucial in identifying fraudulent transactions and protecting users from financial loss. However, these models can be vulnerable to adversarial examples, which can exploit the models' weaknesses and bypass detection. One can identify these vulnerabilities by employing adversarial attacks, allowing for the development of more robust and secure fraud detection systems.

Relating these attacks to real-world scenarios, consider how hackers and cybercriminals might use similar techniques to bypass security systems. Just as adversarial attacks introduce subtle changes to fool machine learning models, hackers may attempt to manipulate transaction data or exploit system weaknesses to evade detection mechanisms. Understanding and defending against adversarial attacks can thus be seen as a proactive measure to thwart potential cyber threats and improve the overall security of financial systems.

4.3.1 White-Box Models

A white-box attack presumes that the attacker has all control over the model's internal operations, including its parameters, architecture, and training data. For instance, someone might gather all the data while working as an intern at a bank. With such extensive access, the attacker may compute gradients and other internal metrics, which makes it easier to create

adversarial examples designed particularly to take advantage of the weaknesses in the model. Keeping this in mind, this research explores White-box attacks since they are particularly effective because they leverage detailed information about the model to maximise the impact of the perturbations. According to the paper by Ma, Zhang, Shen, Marshall and Chang (2023) , white-box attacks exploit the model's detailed information to generate perturbations that deceive the model during the inference stage. Attackers use the model's gradients to create adversarial examples that maximise the perturbation's impact while minimising perceptibility.

Two common methods for generating adversarial examples in a white-box setting are the Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD).

Fast Gradient Sign Method (FGSM)

Goodfellow, Shlens and Szegedy (2015) introduced the Fast Gradient Sign Method (FGSM), a simple and computationally effective method for producing adversarial cases. To produce perturbations, FGSM uses the gradient of the loss function with respect to the input characteristics. The perturbation is applied in the direction of the gradient sign, ensuring that the changes are minimal yet effective in misleading the model. Specifically, the adversarial examples are generated by :

$$x' = x + \epsilon \cdot \text{sign}\left(\nabla_x J(f(x;\theta), l)\right), \tag{4}$$

where x is the original input, x' is the adversarial example, ϵ is the parameter for perturbation magnitude of value 0.2, $\nabla_x J(\theta, x, y)$ is the gradient of the loss function J with respect to the original input x, θ represents the model parameters, and y is the true label. Lastly, sign$(\cdot)$ returns 1 (-1) if the value of the gradient direction is greater (smaller) than 0.

Projected Gradient Descent (PGD)

The Projected Gradient Descent (PGD) method is an iterative extension of FGSM. PGD applies small perturbations iteratively and projects the perturbed sample back onto the feasible set defined by an ϵ-ball around the original input. The key difference between PGD and FGSM is that while FGSM applies a single-step perturbation, PGD uses multiple iterations to gradually adjust the input, ensuring that the perturbations stay within a constrained range around the original input. This makes PGD particularly useful for creating more sophisticated and effective adversarial examples that can bypass model defences more reliably (Ma et al., 2023).

Algorithm 1: PGD Algorithm

Input: signal x, label l or l^t, loss function $J(\cdot)$, size of the perturbation ϵ, the size on each iteration step α, number of iterations I

Output: An adversarial example x'_I with $\|x'_I - x\|_p \leq \epsilon$

Initialize $x'_0 = x$, $i = 1$;

for *i in range I* **do**

  $x'_i = \text{Clip}_{x,\alpha}\left(x'_{i-1} + \alpha \cdot \text{sign}(\nabla_x J(f(x;\theta), l^x))\right)$;

end

return x'_I

[a]

[a]Where ϵ is the perturbation magnitude and is set to 0.2, α is the step size of 0.01 and 10 iterations are performed. Reference : Goodfellow et al. (2015)

In this research, robust adversarial examples were created for RF model training by combining the FGSM and PGD methods to generate adversarial attacks. The R packages *randomForest* and *reticulate* were used to put this strategy into practice. The dataset was split into 70% training and 30% testing sets. The *generate_fgm* function created adversarial examples by perturbing the training data using FGM with an epsilon value of 0.2. Similarly, the *generate_pgd* function generated adversarial samples using PGD with an *epsilon* of 0.2, *alpha* of 0.01, and 10 iterations. These adversarial datasets were then combined with the original training data to train the adversarial RF model, configured to use 100 trees *(ntree = 100)*, 5 features per split *(mtry)*, 30 maximum terminal nodes *(maxnodes)*, and a minimum terminal node size *(nodesize)* of 10. Therefore, the goal of this combined strategy is leverage the advantage of FGSM's computational efficiency and PGD's robustness (Huang et al., 2022) in order to expose the RF model to a greater range of adversarial perturbations and thereby equip it to handle diverse attacks.

4.3.2 Black-Box Models

Black-box attacks simulate real-world scenarios where attackers such as hackers and cybercriminals do not have access to the internal workings of the fraud detection system. The attacker can only interact with the model by providing input data and observing the output. This limitation simulates realistic scenarios where attackers attempt to deceive the model without knowing the internal mechanics of the systems they are trying to exploit. Therefore, this is a valuable approach for evaluating the security of machine learning models used in fraud detection.

HopSkipJump Attack (HSJA)

The HopSkipJump Attack (HSJA) is an effective method for generating adversarial examples in a black-box setting. This method detailed by J. Chen et al. (2020), is particularly relevant for the fraud detection models used in this research as it requires minimal information about the model and focuses on reducing the number of queries needed to generate successful adversarial examples.

HSJA begins by finding an initial adversarial example through random sampling. This initial point is perturbed to lie within the decision boundary of the model. The attack then performs a binary search to find a point close to the decision boundary between the adversarial and non-adversarial regions, ensuring that the perturbation is minimal yet effective. Unlike white-box attacks that rely on gradient information, HSJA estimates the gradient by querying the model. This is achieved by slightly perturbing the input and observing the changes in the model's output. The attack adaptively adjusts the step size based on the success of previous perturbations, efficiently converging to a successful adversarial example with minimal queries. The process is repeated iteratively, refining the adversarial example with each iteration until the disturbance is sufficient to deceive the model reliably. The working of this method can be seen from Algorithm 2 and visualised by Fig.4.

Algorithm 2: HopSkipJump Attack

Classifier C, a sample x, constraint ℓ_p, initial batch size B_0, iterations T Perturbed image x_t Set θ [a]. Initialise at $\tilde{x}_0$ with $\phi_{x^*}(\tilde{x}_0) = 1$[b]. Compute $d_0 = \|\tilde{x}_0 - x^*\|_p$[c];

for t *in* $1, 2, \ldots, T-1$ **do**

`// Boundary search`

$x_t = \text{Bin-Search}(\tilde{x}_{t-1}, x, \theta, \phi_{x^*}, p)$;

`// Gradient-direction estimation`

Sample $B_t = B_0\sqrt{t}$ unit vectors $u_1, \ldots, u_{B_t}$;

Set δ_t [d];

Compute $v_t(x_t, \delta_t)$ [e];

`// Step size search`

Initialise step size $\xi_t = \|x_t - x^*\|_p/\sqrt{t}$;

while $\phi_{x^*}(x_t + \xi_t v_t) = 0$ **do**

$\xi_t \leftarrow \xi_t/2$;

end

Set $\tilde{x}_t = x_t + \xi_t v_t$;

Compute $d_t = \|\tilde{x}_t - x^*\|_p$;

end

Output $x_t = \text{Bin-Search}(\tilde{x}_{t-1}, x, \theta, \phi_{x^*}, p)$;

[f]

[a] θ is a parameter set used in the binary search and gradient estimation processes.
[b] $\phi_{x^*}(\tilde{x}_0) = 1$ indicates the initial condition that $\tilde{x}_0$ is within the decision boundary.
[c] d_0 is the initial distance from the adversarial example to the original input under the ℓ_p norm.
[d] δ_t is a perturbation parameter that scales the direction vectors u_i.
[e] $v_t(x_t, \delta_t)$ is the estimated gradient direction.
[f] Reference: J. Chen et al. (2020)

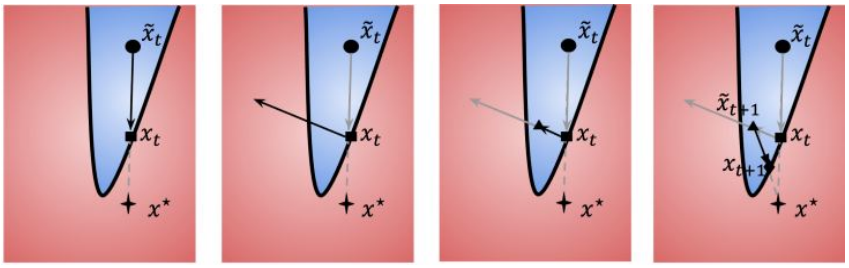


Figure 4: Intuitive explanation of HopSkipJumpAttack : (a) Perform a binary search to locate the boundary and update $\tilde{x}_t \rightarrow x_t$. (b) Estimate the gradient at the boundary point x_t. (c) Apply geometric progression to update $x_t \rightarrow \tilde{x}_{t+1}$. (d) Conduct another binary search and update $\tilde{x}_{t+1} \rightarrow x_{t+1}$. Image from J. Chen et al. (2020)

To implement the HSJA in this study, the process began by generating adversarial examples in Python using the *art* library and *sklearn* package. The data was split into training (70%) and testing (30%) sets. A RF model was trained as the Oracle with 100 trees *(n_estimators=100)*, using *sqrt* for *max_features*, 30 maximum leaf nodes *(max_leaf_nodes=30)*, and a minimum of 10 samples per leaf *(min_samples_leaf=10)*. This model was then wrapped with the *art.estimators.classification.SklearnClassifier* to apply the attack. The HSJA was initialised and adversarial examples were generated from the test data.

Further, these generated adversarial examples were then imported into R for model evaluation. The dataset was once again split into training (70%) and testing (30%) sets using the *createDataPartition* function from the *caret* package. The RF model was trained using the *randomForest* package, configured with 100 trees *(ntree=100)*, 5 features per split *(mtry)*, 30 maximum terminal nodes *(maxnodes=30)*, and a minimum terminal node size *(nodesize=10)*. The model's performance was evaluated on both clean test data and the adversarial examples, ensuring the predictions and actual labels were factors with the same levels.

4.4 One-Class Support Vector Machine

One-Class Support Vector Machine (SVM) is a highly effective approach for anomaly detection, particularly effective in identifying fraudulent transactions. Introduced by Schölkopf and Smola (2002) and Müller, Mika and Räshc (2001), One-Class SVM is an unsupervised learning technique that differentiates normal data points from outliers by mapping the training data from input space to a higher-dimensional feature space using a kernel function.

In this feature space, the algorithm identifies a hyperplane that maximises the margin to separate the majority of the data from the origin. This separation effectively creates a boundary around the normal data points, with points outside this boundary identified as anomalies ie. the fraudulent transactions (Hejazi & Singh, 2013). This can be visualised in Fig.5, where the regular training observations (white circles) represent the normal data used to train the model. New regular observations (purple circles) are additional normal data points that were not part of the training set but still fall within the learned boundary, indicating they are correctly identified as non-anomalous. New abnormal observations (yellow circles) are data points outside the learned boundary, indicating anomalies.

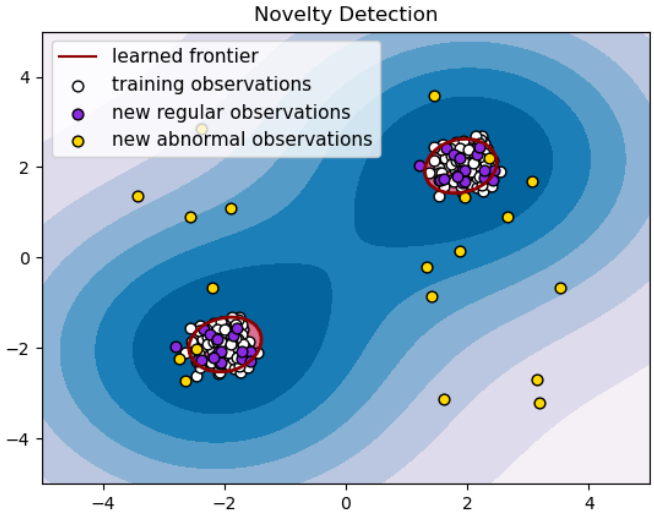


Figure 5: Novelty Detection using One-Class SVM. The plot shows the learned decision frontier (red line) which separates normal observations (inside) from anomalies (outside). Regular training observations are represented by white circles, new regular observations by purple circles, and new abnormal observations by yellow circles. Image from scikit-learn (2024).

As One-Class SVM may detect data points that considerably differ from conventional patterns, it is useful in context of this research. Given this, it can be used to identify fraudulent transactions that deviate from normal user behavior. Additionally, the kernel-based approach allows One-Class SVM to capture complex relationships in the data, enhancing its ability to detect subtle anomalies.

One-Class SVM uses kernel functions, such as linear, polynomial, or radial basis function (RBF) kernels, to map the training data to a feature space. This method mathematically computes the hypersphere with centre c and radius r in Eq.5. Here, the function Φ is the hypersphere transformation of x samples (Hejazi & Singh, 2013).

$$\begin{aligned} \min_{r,c,\zeta} \quad & r^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \zeta_i \\ \text{subject to,} \quad & \|\Phi(x_i) - c\|^2 \leq r^2 + \zeta_i \quad \forall i = 1, 2, \ldots, n. \end{aligned} \tag{5}$$

Initially, the dataset was divided into training and testing subsets with a 70-30 split ratio. The One-Class SVM model was trained using just non-fraudulent transactions from the training set. This choice helps the model learn the traits of genuine transactions, which is essential for spotting anomalies. To enhance the model's performance, the features of the normal training data were scaled. Scaling ensures that the optimisation process of the SVM converges more efficiently by normalising the feature values. The scaling was performed using a preprocessing step that centres and scales the features of the training data. The One-Class SVM model was then trained on the scaled normal data using the RBF kernel. Specific parameters, such as the

regularisation parameter ν and the kernel parameter γ, were set to 0.01 and 0.1, respectively. These parameters were chosen to balance the trade-off between maximising the margin and allowing for some flexibility in handling outliers. After training the model, predictions were made on the scaled test data. The model predicted whether each transaction was similar to the normal transactions (non-fraudulent) or an outlier (potentially fraudulent). Following that, the predictions were converted to binary labels, where fraudulent transactions were labeled as 1 and non-fraudulent transactions as 0.

4.5 Quantile Regression Forest

Meinshausen (2006) created Quantile Regression Forests (QRF), which extend the RF methodology (4.2) by estimating the full conditional distribution of the response variable. This approach makes it possible to compute conditional quantiles, which offers a more thorough comprehension of the distribution of the data. Similar to conventional RF, multiple decision trees are constructed in QRF using bootstrap samples and random subsets of predictor variables. However, instead of averaging the terminal node responses, QRF retains the entire distribution of responses, allowing for the estimation of quantiles.

For a given input $X = x$, QRF calculates weights $w_i(x)$ for each observation based on the proportion of trees where the observation falls into the same terminal node as x. The conditional distribution function $\hat{F}(y \mid X = x)$ is then estimated using these weights. Specifically, the quantiles $Q_\alpha(x)$ are derived from this distribution function and are defined as:

$$Q_\alpha(x) = \inf \{y : F(y \mid X = x) \geq \alpha\}, \tag{6}$$

where $F(y \mid X = x)$ is the CDF. . This methodology is especially useful in the context of fraud detection since it makes it possible to identify transactions at different risk levels (Meinshausen, 2006). QRF helps in assessing the uncertainty in predictions and more precisely identifying high-risk fraudulent transactions by providing comprehensive probability distributions and quantiles. This method's reliability has been validated, and it has been extended for use in this research. The replication details are available in Appendix A.

4.6 Random Forest Quantile Classifier

Building on the concept of QRF, O'Brien and Ishwaran (2019) introduced the Random Forest Quantile Classifier (RFQC), which extends the quantile model to handle categorical outputs, making it highly suitable for binary classification problems like fraud detection. The RFQC adapts the quantile regression approach to classification tasks by generalising the concept of conditional quantiles to categorical data. This extension is particularly relevant for handling class-imbalanced datasets in scenarios like fraud detection. The motivation for using this method stems from its ability to provide more informative metrics for evaluating model performance in the presence of imbalanced classes, thereby enhancing the detection of fraudulent transactions.

Traditional classifiers often struggle with imbalanced data, favouring the majority class. To counter this, the q^*-classifier (RFQC) assigns samples to the minority class if their conditional probability exceeds a specified quantile threshold (q^*): $0 < q^* < 1$, where q^* equals the unconditional probability of observing a minority class sample. This approach by O'Brien and Ishwaran (2019) is motivated by a density-based methodology, resulting in the dual optimisation of the q^*-classifier (RFQC): maximising true positive rate (TPR) and the true negative rate (TNR). By optimising both TNR and TPR, the q^*-classifier ensures a balanced detection of fraudulent and non-fraudulent transactions, thereby reducing both false positives and false negatives as can be seen in Eq.7.

$$r(\delta_{q^*}, \pi, 1-\pi) = E[\min\{(1-\pi)p(X), \pi(1-p(X))\}] \leq E[\pi(1-p(X))] \leq \pi. \tag{7}$$

A cost-weighted Bayes classifier framework can be used to characterise the q^*-classifier, minimising the weighted risk. Here in Eq.7, *p(X)* is the model-predicted probability for the sample X and π is the minority class probability (fraudulent transactions). This equation shows that the risk is minimised for both marginally and conditionally imbalanced data, unlike the traditional Bayes rule which does not ensure the joint optimisation of TNR and TPR.

Unlike traditional classifiers that may focus on optimising overall accuracy, the RFQC is tailored to enhance the detection of minority class instances, thereby minimising the false positive rate (FPR). This is crucial in fraud detection, where falsely flagging legitimate transactions can be costly. The RFQC focuses on minimising the weighted risk rather than optimising overall accuracy, leveraging the G-mean metric (Eq.8), which balances the TPR and TNR, thereby providing a more comprehensive performance evaluation in imbalanced datasets.

$$\text{G-Mean} = \sqrt{\text{Sensitivity} \times \text{Specificity}}. \tag{8}$$

To implement the RFQC in this research, the *imbalanced function* from the *randomForestSRC* package by O'Brien and Ishwaran (2019) is adopted. It provides a Random Forest implementation of the q^*-classifier for the two-class imbalance problem and is used in this research to train the model on the training data. This function is specifically designed to handle class imbalance, optimising the model's ability to distinguish between fraudulent and non-fraudulent transactions. The model is trained with 100 trees ($ntree = 100$). To assess the model's performance, the dataset was divided into a training set (70%) and a testing set (30%), following the standard approach as done previously in other methods. After training, the model's predictions on the test set were used to assess its effectiveness in identifying fraudulent transactions.

Moreover,the G-mean (Eq.8) and Variable Importance Measures (VIMP) are used to extract the significance of each variable in the model (Ishwaran, O'Brien, Lu & Kogalur, 2021). Moreover, the importance of each variable in the model is extracted using the G-mean (Eq.8) and Variable Importance Measures (VIMP) (Ishwaran et al., 2021). This adds to the assessment of model's overall performance by highlighting the features that are crucial for detecting fraudulent behaviour. Unlike other performance measures such as accuracy, F1-score, and AUC-ROC

(detailed in Section 5.1), which evaluate the model's overall predictive power, VIMP focuses on the contribution of individual features. This would help understand which features of transactions are crucial for flagging suspicious activity in credit cards.

5 Results

This section displays the results obtained following the methods detailed in Section 4. The analysis was conducted on the real-world dataset containing credit card transactions as presented in Section 3. Each methodology is evaluated based on various performance measures presented in Section 5.1.

5.1 Performance Measures

The performance of classification models, whether binary or multi-class, is often evaluated using a confusion matrix. This matrix includes four key metrics: True Positives (correctly identified class examples), True Negatives (correctly identified non-class examples), False Positives (incorrectly identified as class examples), and False Negatives (class examples not recognised). These metrics form the confusion matrix, illustrated in Table 2. For this research on fraud detection, which is a binary classification problem, the confusion matrix is a 2x2 table showing these four possible outcomes.

Table 2: Confusion Matrix

		Actual Values	
		Positive (1)	**Negative (0)**
Predicted Values	**Positive (1)**	TP	FP
	Negative (0)	FN	TN

Note: The table displays the confusion matrix used to evaluate the performance of different methods. TN - True Negative, TP - True Positive, FP - False Positive, FN - False Negative.

The values of this confusion matrix are used in the performance measures in the table below.

Table 3: Performance Measures and their Definitions

Formula	**Explanation**
Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$	Total correct predictions out of all predictions.
True Positive Rate (TPR) $= \frac{TP}{TP+FN}$	Proportion of actual frauds correctly identified by the model (Sensitivity/Recall).
True Negative Rate (TNR) $= \frac{TN}{TN+FP}$	Proportion of actual non-frauds correctly identified by the model (Specificity).
False Positive Rate (FPR) $= \frac{FP}{FP+TN}$	Proportion of non-frauds incorrectly identified as frauds by the model.
False Negative Rate (FNR) $= \frac{FN}{FN+TP}$	Proportion of frauds incorrectly identified as normal transactions by the model.
Precision $= \frac{TP}{TP+FP}$	Proportion of transactions identified as fraud that actually are frauds.
F1-Score $= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$	Harmonic mean of precision and recall.
AUC-ROC $= \int_0^1 \text{ROC}(x)\, dx$	Area under the curve, measuring the model's ability to distinguish between fraudulent and non-fraudulent transactions.
Precision-Recall Curve (PRC)	Plot of precision against recall, useful for imbalanced datasets .

Note: This table provides the performance measures, their formulas, and explanations in the context of this study. The metrics range from 0 to 1. More details about these methods can be found in Appendix B.

While the accuracy rate is straightforward to understand and compute, it is not always a reliable metric. In datasets where one class is significantly more prevalent than the other, a model can achieve high accuracy by simply predicting the majority class for all instances. This would result in high accuracy but poor performance of the model in detecting the minority class (fraudulent transactions). Therefore, solely relying on the accuracy rate to measure the performance of a model is not advisable since the main purpose of a fraud detection system is to be able to flag fraudulent activity.

On the other hand, the Precision-Recall (PR) curve is particularly effective for imbalanced datasets like fraud detection, where fraudulent transactions (the positive class) are much rarer than non-fraudulent ones (the negative class). Unlike the ROC curve, which includes the true negative rate (TNR) and can give an overly optimistic view of performance, the PR curve focuses on positive class performance. Saito and Rehmsmeier (2015) point out that PR curves provide a more intuitive and accurate interpretation of classifier performance and highlight model susceptibility to class imbalance. Therefore, while all performance measures are considered, special emphasis is placed on the PR curve to evaluate model effectiveness in this research.

Table 4 provides an overview of the results obtained in this study. The next sections will dive into detail about them.

Table 4: Performance Measure Results of Different Methods

	Accuracy	PR Curve	Sensitivity	Specificity	FPR	FNR	Precision	F1-Score	AUC-ROC
Random Forest	0.9995	0.80	0.9999	0.7447	0.00044	0.07895	0.9996	0.9997	0.9322
Random Forest with White-box Model	0.9994	0.82	0.9999	0.6950	0.00051	0.03922	0.9995	0.9997	0.9108
Random Forest with Black-box Model	0.9994	0.84	0.9999	0.5969	0.00062	0.02532	0.99938	0.9997	0.7984
One-class SVM	0.9473	NA	0.9475	0.8308	0.00027	0.9764	0.9997	0.9729	0.9324
Random Forest Quantile Classifier	0.9736	0.47	0.9737	0.8936	0.00018	0.9466	0.9998	0.9866	0.9337

Note: The table displays the performance measures of different methods. Accuracy, PR-Curve (Precision-Recall Curve), Sensitivity, Specificity, FPR (False Positive Rate), FNR (False Negative Rate), Precision, F1-Score, and AUC-ROC (Area Under the Receiver Operating Characteristic Curve) are the metrics used for evaluation.

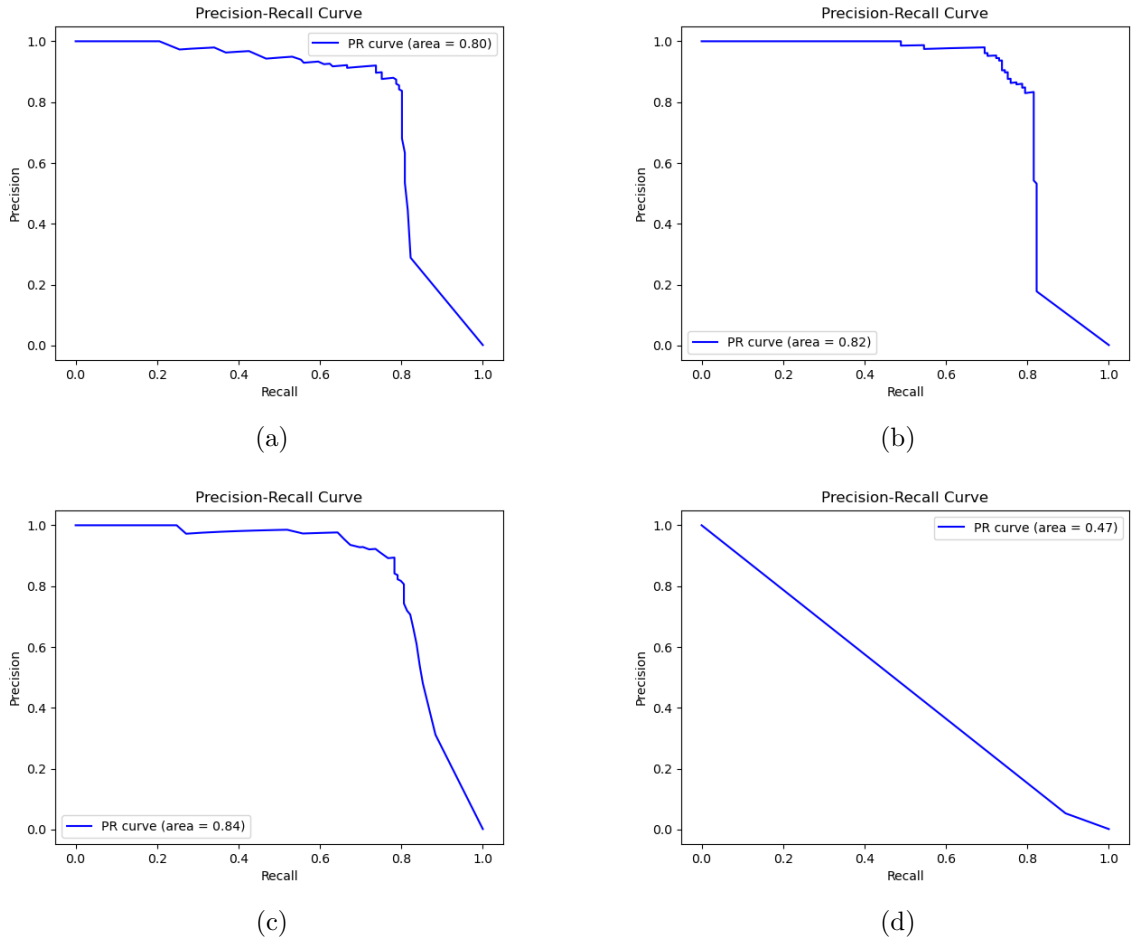


Figure 6: PR Curves for Different Models: (a) Random Forest, (b) Random Forest with White-box Model, (c) Random Forest with Black-box Model, (d) Random Forest Quantile Classifier. On the graphs, the x-axis represents the recall and the y-axis represents the precision.

5.2 Random Forest

With an incredibly high accuracy of 99.95% and sensitivity of 99.99%, the standard RF model strongly depicts that it can correctly detect fraudulent transactions. However, the 74.47% specificity suggests a greater false positive rate of 0.044%. This means that while the model is excellent at detecting frauds, it may incorrectly flag some legitimate transactions as fraudulent. This could lead to customer dissatisfaction due to unnecessary alerts or actions. The high sensitivity indicates that it catches almost all fraudulent transactions, making it very effective in fraud detection, but the moderate specificity suggests room for improvement in reducing false positives.

5.3 Adversarial Random Forest

White-Box model

When analysed using white-box adversarial attack methods such as FGSM and PGD, the Random Forest model shows improved performance. The specificity is at 69.50%, reducing the false positive rate to 0.0051%, and the false negative rate drops to 3.922%. This indicates fewer missed fraudulent transactions. The robustness against adversarial attacks implies that the model can maintain high detection rates even when fraudsters attempt to evade detection using sophisticated attacks. This method's improved balance between sensitivity and specificity makes it a reliable option for detecting credit card fraud while maintaining customer trust by reducing false positives. Additionally, the model's ability to handle cyberattacks enhances its utility in real-world fraud detection scenarios. The PR curve for this model shows an area of 0.82, indicating a better precision-recall balance compared to the standard model, which further confirms its robustness under adversarial conditions.

Black-Box model

The analysis using the HSJA method reveals that the RF model maintains high accuracy and sensitivity but has a significantly lower specificity of 59.69%. The false positive rate of 0.062% and moderate AUC-ROC of 79.84% suggest that the model struggles more to distinguish between fraudulent and non-fraudulent transactions when subjected to black-box attacks. This vulnerability implies that while the model is still effective in detecting fraud, it may generate more false alarms under adversarial conditions, potentially overwhelming fraud detection systems and causing inconvenience to legitimate users. Despite this, the model's ability to detect fraudulent transactions remains high, demonstrating its potential in handling complex cyberattacks. The PR curve for this model shows an area of 0.84, indicating an even better balance between precision and recall under black-box attack conditions.

Table 5: Adversarial Attack Metrics for Black-Box Model

Metric	**Value**
Injection Rate	50.34%
Evasion Rate	41.09%
Attack Detection Rate	58.91%

Additionally, the injection rate for the adversarial examples is 50.34%, with an evasion rate of 41.09% and an attack detection rate of 58.91%. These metrics provide further insight into the model's performance and limitations in an adversarial context. The injection rate indicates the proportion of adversarial examples within the dataset, reflecting the extent to which the dataset is manipulated by adversarial tactics. The evasion rate highlights the percentage of actual fraudulent transactions that the model failed to detect, demonstrating the model's vulnerability to sophisticated adversarial strategies. Conversely, the attack detection rate measures the model's success in identifying adversarial examples, showcasing its ability to withstand and

respond to adversarial attacks. In a broader context, this displays the importance of developing fraud detection systems capable of maintaining high performance even when faced with advanced adversarial techniques, ensuring the reliability and security of financial systems.

5.4 One-Class SVM

The One-Class SVM model shows good performance with an accuracy of 94.73% and sensitivity of 94.75%. The higher specificity of 83.08% indicates a low false positive rate of 0.0027%, meaning it effectively identifies normal transactions. However, the high false negative rate of 97.64% implies many fraudulent transactions might be missed. This model is suitable for environments where minimising false positives is more critical than catching all frauds. However, its vulnerability to missing frauds is a concern, suggesting the need for additional measures to improve its toughness against cyberattacks.

In Table 3, it can be seen that the PR curve cannot be generated for this model. One-Class SVM is designed for anomaly detection, learning from normal data to identify outliers. It does not offer explicit probabilities for class membership, which is necessary for creating PR curves, in contrast to binary classifiers. PR curves, which require continuous probability scores for both positive and negative classes, plot accuracy against recall at various threshold values. One-Class SVM typically outputs binary decisions (anomaly or not) without these intermediate probability scores.

Additionally, 4569 anomalies, or 5.3679% of all transactions, are predicted by the model. The test sample has a very low percentage of anomalies at just 0.1527%.. This discrepancy indicates the model's conservative approach in flagging potential fraud, suggesting it errs on the side of caution. This approach is beneficial in high-risk environments where the cost of missing a fraudulent transaction is significantly higher than the inconvenience of a false positive. However, this conservative stance also calls for the need of continuous monitoring and potential adjustment of the model to balance detection rates with the rate of false positives effectively.

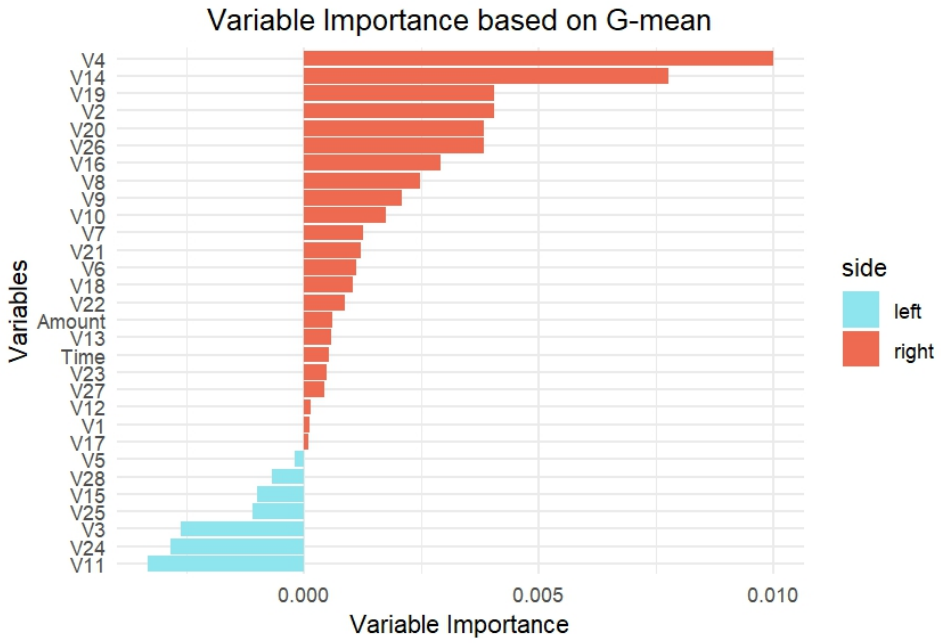
5.5 Random Forest Quantile Classifier

This classifier shows a balanced and robust performance with an accuracy of 97.36% and a high sensitivity of 97.37%. The specificity of 89.36% and low false positive rate of 0.018% indicate that it is reliable in detecting fraudulent transactions without raising many false alarms. The false negative rate of 94.66% and high AUC-ROC of 93.37% suggest that this method is highly effective in distinguishing between legitimate and fraudulent transactions. Its inherent balance and optimisation make it a highly effective model for practical fraud detection. The PR curve for this model shows an area of 0.47, indicating a moderate precision-recall balance, which stresses the importance of evaluating both precision and recall for imbalanced datasets.

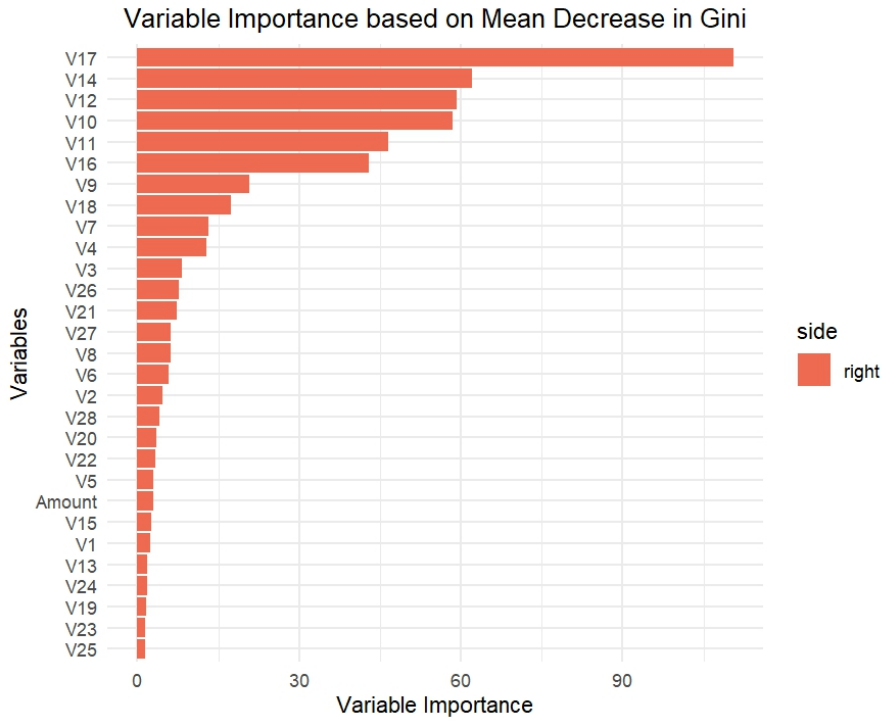
This classifier was also used to assess the importance of various features in detecting fraudulent transactions. Figure 7a presents the variable importance (VIMP) scores based on the G-mean metric. This metric helps identify the most influential variables in the model's decision-

making process. For comparison, Fig.7 also includes the VIMP based on the mean decrease in Gini by the RF model. The G-mean metric emphasises balancing sensitivity and specificity, making it particularly valuable for imbalanced datasets like fraud detection, where both classes need equal attention. In contrast, the mean decrease in Gini impurity seen in focuses on how well a feature splits the data into homogeneous groups, thereby reducing uncertainty about the target variable.

In Fig.7a, variables like V4, V14, and V19 show high positive importance, indicating their strong influence in detecting fraudulent transactions based on the G-mean metric. Negative importance suggests less relevance or detrimental impact. On the other hand, Fig.7b shows variable importance in RF measured by Mean Decrease in Gini, where variables such as V17, V14, and V12 are significant. This importance was also visualised previously in Fig.3, in the form of a decision tree. Essentially, the differences in the graphs (7) are owed to the fact that the G-mean prioritises balanced performance across classes, while the mean decrease in Gini measures overall impurity reduction in decision trees.



(a) Quantile Classifier VIMP



(b) Random Forest VIMP

Figure 7: Variable Importance based on different methods. The red bars indicate variables that have a positive importance, while the blue bars indicate variables with negative importance

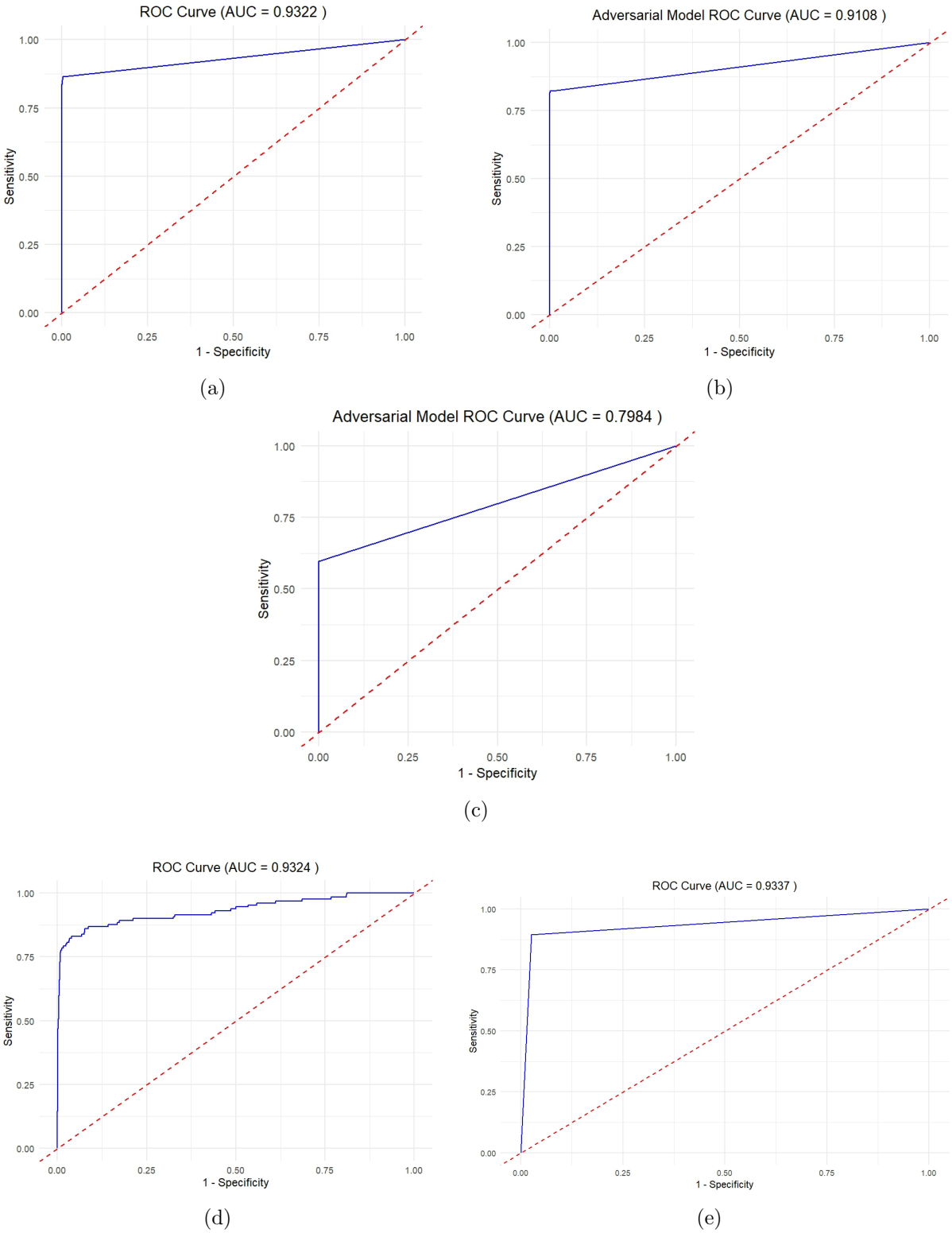


Figure 8: AUC-ROC Curves for Different Models : (a) Random Forest, (b) Random Forest with White-box Model, (c) Random Forest with Black-box Model, (d) One-class SVM, and (e) Random Forest Quantile Classifier. On the graphs, the x-axis represents the false positive rate and the y-axis represents the true positive rate.

6 Conclusion and Discussion

This thesis examined the efficacy of various classification models in detecting credit card fraud within an imbalanced data. The primary goal was to assess their performance and propose improvements that can enhance them further. The purpose of this thorough investigation was to contribute to the existing body of work by investigating novel approaches such as adversarial attacks and quantile classifiers. These methods had not been substantially applied to this dataset before.

The Random Forest (RF) model emerged as a powerful performer, showing great success in detecting fraudulent transactions with an F1-Score of 99.97% and an AUC-ROC of 93.22% (Fig.8). Moreover, its PR curve displayed an area of 0.80, indicating a balanced performance between precision and recall. However, its false positive rate (FPR) of 0.044% suggests an incidental misclassification of legitimate transactions as fraudulent. While this can lead to customer discontentment, the model's high sensitivity ensures that almost all suspicious activities are detected.

Enhancing the RF model with white-box adversarial attack methods such as FGSM and PGD led to significant performance enhancements; its PR curve indicated significant gains with an area of 0.82. This displayed an enhanced balance between precision and recall under adversarial conditions, making this model effective in detecting credit card fraud while upholding consumer trust by limiting false positives. The durability against adversarial attacks signifies that the model can ensure high detection rates even when fraudsters attempt to evade detection. Conversely, black-box adversarial methods analysed using the HSJA technique revealed vulnerabilities. While the model displayed strong detection capabilities with an F1-Score of 99.38%, its increased FPR of 0.062% and lower AUC-ROC of 79.84%(Fig.8) suggest challenges in distinguishing fraudulent transactions from normal ones under adversarial conditions. Additionally, the injection rate for adversarial examples is 50.34% with an evasion rate of 41.09% and an attack detection rate of 58.91%. Hence, this conveys a need for additional countermeasures that booster the model against sophisticated fraud techniques that exploit its weaknesses.

With a success rate of 94.73% for accuracy and 83.58% for specificity in recognising valid transactions, the One-Class SVM model displayed an effective balance. Its elevated false negative rate (FNR) suggested that a large number of fraudulent ones would evade detection, though. Additionally, during the testing phase, the model predicted 4569 anomalies, which accounted for 5.36799% of transactions, and an anomaly rate of just 0.1527%, indicating probable over-flagging issues. These findings further underscore the conservative character of the model.

The Random Forest Quantile Classifier (RFQC) endorsed impressive performance, boasting an F1-Score of 98.66% and an AUC-ROC of 93.37% (Fig.8). With a FPR rate of 0.018% and a FNR limit of 94.66%, its reliable detection capabilities without creating unnecessary false alarms stand out as features of note for further research in fraud detection. However, the PR curve has demonstrated average performance when distinguishing fraudulent from non-fraudulent transactions in the context of imbalanced datasets.

In conclusion, this thesis has demonstrated that while traditional models like Random Forests perform well in detecting fraud, their resilience can be considerably strengthened through adversarial training techniques.The exploration of novel approaches such as the RFQC and adversarial attacks offers further perspective into improving fraud detection systems. Drawing attention to the other methods used, the One-Class SVM approach, for instance, ensures that the model remains on the side of caution, preferring to flag potential anomalies rather than risk missing fraudulent transactions. This approach has proven especially successful when applied in high-stakes financial environments - as evidenced by its higher false negative rate. This strategy would result in reduced fraud whilst protecting the integrity of customers' data. In order to detect real frauds more effectively, it tends to overlook fewer fraudulent transactions but may mistakenly classify more legal ones as fraudulent, indicating a high sensitivity (true positive rate) at the expense of poorer specificity. The conservative aspect of this model, however, also suggests that there is potential for improvement in terms of identifying fraudulent transactions without sacrificing its ability to identify legitimate ones.

Moreover, among the models examined, the RFQC is notable for having the lowest false positive rate. This showcases that the classifier is quite good at correctly detecting normal transactions (true negatives) in addition to identifying fraudulent ones (true positives). While its performance in minimising false positives stands out, its performance among other metrics can still be improved.

Therefore, the results of this paper highlight the significance of employing cutting-edge methods to remain ahead in the ever-evolving field of fraud detection, opening the door for more research and advancement in this crucial sector. As we continue to innovate and adapt, our goal remains unchanged: creating a safer and fraud-free financial world.

Limitations and Future Research

A significant limitation encountered in this research is the limited availability of real-world datasets for credit card fraud detection, as discussed in various sections of this paper. Accessible datasets, like the one used in this study, are frequently anonymised to preserve privacy. The model's interpretability decreases when variables are anonymous, particularly when examining the impact of these variables and how they affect the results. This restriction makes it difficult to apply the model's takeaways to real-world scenarios, where it is paramount to understand the precise effects of such variables.

The Isolation Forest was used in this study in addition to the previously mentioned techniques. However, it had a meagre 0.0016 accuracy. This performance was below expectations for several reasons. The isolation method was mostly an unsupervised technique that did not work well with the labelled dataset; for best results, a supervised learning approach is needed. Its utility in this environment was further hampered by its incapacity to distinguish between subtle anomalies and legitimate transactions.

Furthermore, future studies should focus on enhancing the capabilities of these models, particularly when confronted with adverse circumstances. Adding more techniques will be crucial to lowering false positives and increasing detection rates. More research into combining these models with additional complex algorithms and real-time data processing could result in more flexible and all-encompassing fraud detection systems. By opening the door for the creation of more reliable and accurate fraud detection models, the study's findings support ongoing efforts to protect financial transactions..

References

Adewumi, A. O. & Akinyelu, A. A. (2016). A survey of machine-learning and nature-inspired based credit card fraud detection techniques. *International Journal of System Assurance Engineering and Management*, *8*(S2), 937–953. Retrieved from https://doi.org/10.1007/s13198-016-0551-y doi: 10.1007/s13198-016-0551-y

Akhilomen, J. (2013). Data mining application for cyber credit-card fraud detection system. In *Advances in Data Mining Applications and Theoretical Aspects: 13th Industrial Conference, ICDM 2013, New York, NY, USA, July 16-21, 2013. Proceedings 13* (pp. 218–228).

Bhatla, T. P., Prabhu, V. & Dua, A. (2003). *Understanding Credit Card Frauds.* Tata Consultancy Services.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32. Retrieved from https://doi.org/10.1023/A:1010933404324 doi: 10.1023/A:1010933404324

Carcillo, F., Pozzolo, A. D., Borgne, Y. A. L., Caelen, O., Mazzer, Y. & Bontempi, G. (2018). SCARFF: a Scalable Framework for Streaming Credit Card Fraud Detection with Spark. *Information Fusion*, *41*, 182-194.

Chen, J., Jordan, M. I. & Wainwright, M. J. (2020). Hopskipjumpattack: A query-efficient decision-based attack. In *2020 ieee symposium on security and privacy (sp)* (pp. 1277–1294).

Chen, R., Chiu, M., Huang, Y. & Chen, L. (2004). Detecting Credit Card Fraud by Using Questionnaire-Responded Transaction Model Based on Support Vector Machines. In *Proceedings of IDEAL2004* (pp. 800–806).

Ding, Y., Wang, L., Zhang, H., Yi, J., Fan, D. & Gong, B. (2019). *Defending Against Adversarial Attacks Using Random Forests.* Retrieved from https://arxiv.org/abs/1906.06765

European Central Bank. (2023). *European Central Bank.* Retrieved from https://www.ecb.europa.eu (Accessed: 15-05-2024)

Goodfellow, I. J., Shlens, J. & Szegedy, C. (2015). Explaining and Harnessing Adversarial examples. In *International Conference on Learning Representations (ICLR).* San Diego, CA, USA.

Hejazi, M. & Singh, Y. P. (2013). One-class support vector machines approach to anomaly detection. *Applied Artificial Intelligence*, *27*(5), 351–366. Retrieved from https://doi.org/10.1080/08839514.2013.785791 doi: 10.1080/08839514.2013.785791

Huang, T., Menkovski, V., Pei, Y. & Pechenizkiy, M. (2022). *Bridging the Performance Gap between FGSM and PGD Adversarial Training.* Retrieved from https://arxiv.org/abs/2011.05157

IC3, I. C. C. C. (2022). *Internet Crime Complaint Center (IC3) — Home Page.* Retrieved from https://www.ic3.gov/ (Accessed: 16 May 2024)

Ishwaran, H., O'Brien, R., Lu, M. & Kogalur, U. B. (2021). *randomForestSRC: Random Forests Quantile Classifier (RFQ) vignette.* http://randomforestsrc.org/articles/imbalance.html. Retrieved from http://randomforestsrc.org/articles/imbalance.html

Ma, J., Zhang, J., Shen, G., Marshall, A. & Chang, C.-H. (2023). White-Box Adversarial

Attacks on Deep Learning-Based Radio Frequency Fingerprint Identification. In *ICC 2023-IEEE International Conference on Communications* (pp. 3714–3719).

Maes, S., Tuyls, K., Vanschoenwinkel, B. & Manderick, B. (2002). Credit Card Fraud Detection using Bayesian and Neural Networks. In *Proceedings of the 1st International NAISO Congress on Neuro Fuzzy Technologies.*

Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine Learning Research*, *7*(35), 983–999.

Müller, K., Mika, S. & Räshc, T. (2001). K., Schölkopf, B.: An Introduction to Kernel-based Learning Algorithms. *IEEE Transactions on Neural Networks*, *12*(2), 181–201.

O'Brien, R. & Ishwaran, H. (2019). A random forests quantile classifier for class imbalanced data. *Pattern Recognition*, *90*, 232–249. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0031320319300536` doi: 10.1016/j.patcog.2019.01.036

Perera, P., Oza, P. & Patel, V. M. (2021). One-Class Classification: A Survey. *CoRR*, *abs/2101.03064*. Retrieved from `https://arxiv.org/abs/2101.03064`

Phua, C., Lee, V. C. S., Smith-Miles, K. & Gayler, R. W. (2010). A Comprehensive Survey of Data Mining-based Fraud Detection Research. *CoRR*, *abs/1009.6119*. Retrieved from `http://arxiv.org/abs/1009.6119`

Saito, T. & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, *10*(3), e0118432. Retrieved from `https://doi.org/10.1371/journal.pone.0118432` doi: 10.1371/journal.pone.0118432

Schölkopf, B. & Smola, A. J. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

scikit-learn. (2024). *One-class SVM with non-linear kernel (RBF).* `https://scikit-learn.org/stable/auto_examples/svm/plot_oneclass.html#sphx-glr-auto-examples-svm-plot-oneclass-py`. (Accessed: 26 May 2024)

Sethi, N. & Gera, A. (2014). A revived survey of various credit card fraud detection techniques. *International Journal of Computer Science and Mobile Computing*, *3*(4), 780-791.

Shen, A., Tong, R. & Deng, Y. (2007). Application of Classification Models on Credit Card Fraud Detection. In *International Conference on Service Systems and Service Management* (pp. 1–4). doi: 10.1109/ICSSSM.2007.4280163

Sorournejad, S., Zojaji, Z., Atani, R. E. & Monadjemi, A. H. (2016). *A Survey of Credit Card Fraud Detection Techniques: Data and Technique Oriented Perspective* (Vol. abs/1611.06439). Retrieved from `http://arxiv.org/abs/1611.06439` (Available at arXiv:1611.06439)

Sulaiman, R. B., Schetinin, V. & Sant, P. (2022). Review of machine learning approach on credit card fraud detection. *Human-Centric Intelligent Systems*, *2*(1-2), 55–68.

ULB, M. L. G. (2018). *Credit Card Fraud Detection.* Kaggle. Retrieved from `https://www.kaggle.com/mlg-ulb/creditcardfraud`

Yundong, W., Zhulev, A. & Ahmed, O. G. (2023). Credit Card Fraud Identification Using Logistic Regression and Random Forest. *Wasit Journal of Computer and Mathematics Science*, *2*(3), 1–8. Retrieved from `https://doi.org/10.31185/wjcms.184` doi: 10

.31185/wjcms.184

Zareapoor, M., K.R., S. & Alam, M. A. (2012). Analysis on credit card fraud detection techniques: Based on certain design criteria. *International Journal of Computer Applications*, *52*(3), 35–42. Retrieved from `https://doi.org/10.5120/8184-1538` doi: 10.5120/8184-1538

A Quantile Regression Forest Replication

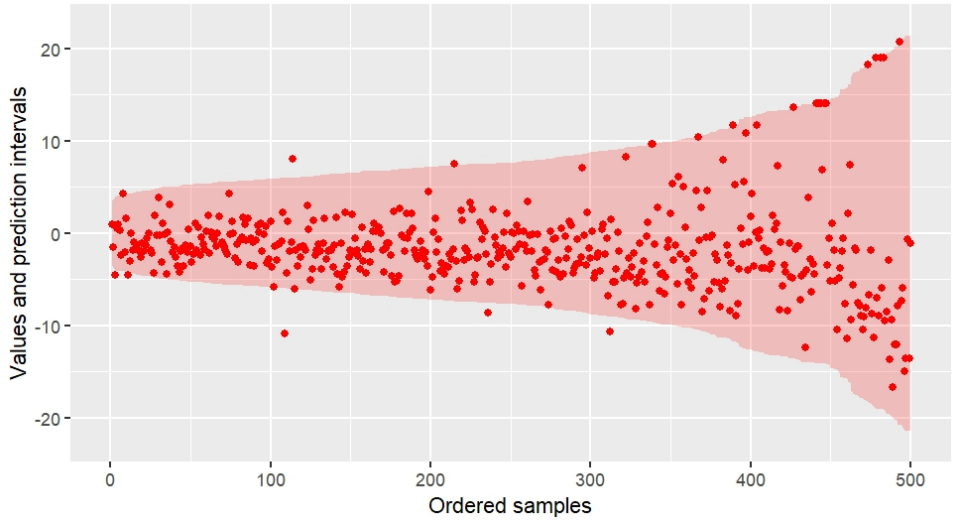


Figure 9: Prediction Intervals Using Quantile Regression Forests

The graph illustrates the prediction intervals generated by the Quantile Regression Forests (QRF) method for the Boston Housing dataset, replicating the approach outlined by Meinshausen (2006). The x-axis represents the ordered samples, while the y-axis shows the values and prediction intervals.

In this graph, each red dot represents an observed value from the dataset, and the shaded region indicates the prediction intervals for those observations. The width of the shaded area reflects the uncertainty in the predictions, with wider intervals indicating greater uncertainty. Impressively, only 8 points fall outside the confidence interval, translating to just 1.58% of the total values in the dataset. This outcome not only perfectly aligns with the findings of Meinshausen (2006), but it also highlights the robustness and precision of the method. Therefore, this replication not only confirms the success of the method but also affirms its reproducibility making it a promising tool for real-world applications.

As can be seen from Fig.9, the intervals widen as the sample index increases. This highlights the increasing uncertainty in the predictions for those samples. This visual representation demonstrates the QRF's ability to capture variability and uncertainty in the data, providing more informative predictions compared to traditional point estimates.

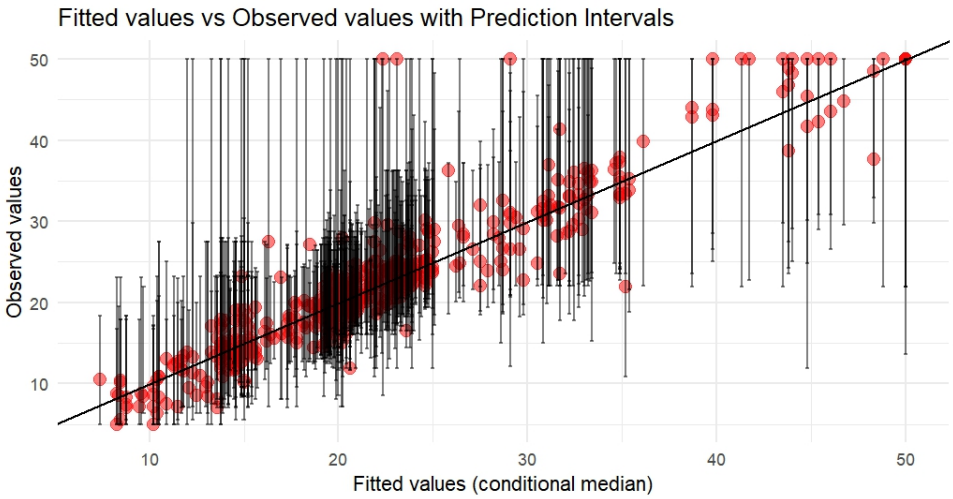


Figure 10: Fitted Values vs Observed values with Prediction intervals

This graph illustrates the relationship between the fitted values and the observed values, along with their prediction intervals, for the Boston Housing dataset. The x-axis represents the fitted values (conditional median), while the y-axis represents the observed values. The red dots represent the actual observations, and the black vertical lines denote the prediction intervals around the fitted values.

The black diagonal line represents a perfect fit where the observed values equal the fitted values. The spread of the red dots and the length of the prediction intervals indicate the model's uncertainty and the variability in the data. Ideally, most points should cluster around the diagonal line, indicating good model performance. However, wider intervals and significant deviations from the line suggest areas where the model's predictions are less reliable.

B Performance Measures

Table 6: Confusion Matrix

		Actual Values	
		Positive (1)	**Negative (0)**
Predicted Values	**Positive (1)**	TP	FP
	Negative (0)	FN	TN

Note: The table displays the confusion matrix used to evaluate the performance of different methods. TN - True Negative, TP - True Positive, FP - False Positive, FN - False Negative.

In the context of credit card fraud detection, a True Positive (TP) occurs when a transaction is correctly predicted as fraudulent. A False Positive (FP) happens when a transaction is incorrectly predicted as fraudulent, but it is actually normal. Conversely, a True Negative (TN) is when a transaction is correctly predicted as normal, and a False Negative (FN) occurs when a transaction is incorrectly predicted as normal but is actually fraudulent.

Accuracy rate

Accuracy rate is a common metric used to evaluate the performance of a classification model. It is calculated as the ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

True Positive Rate (TPR)

The true positive rate (TPR) is also known as Sensitivity or Recall or Hit rate. It measures the proportion of actual frauds correctly identified by the model. It is calculated as :

$$\text{TPR} = \frac{TP}{TP + FN}. \tag{10}$$

A high TPR indicates that the model is effective at detecting fraudulent transactions.

True Negative Rate (TNR)

The True Negative Rate (TNR), also known as Specificity, measures the proportion of actual non-frauds correctly identified by the model. It is calculated as :

$$\text{TNR} = \frac{TN}{TN + FP}. \tag{11}$$

A high TNR implies that the model is well-capable at identifying non-fraudulent transactions.

False Positive Rate (FPR)

The False Positive Rate (FPR) measures the proportion of non-frauds incorrectly identified as frauds by the model. This is shown by calculating the following :

$$\text{FPR} = \frac{FP}{FP + TN}. \tag{12}$$

A high FPR signals that many non-fraudulent transactions are incorrectly flagged as fraud.

False Negative Rate (FNR)

The False Negative Rate (FNR) determines the proportion of frauds incorrectly identified as normal transactions by the model. This is shown using the following formula :

$$\text{FNR} = \frac{FN}{FN + TP}. \tag{13}$$

A high FNR signifies that many fraudulent transactions were missed by the model.

Precision

Precision quantifies the proportion of transactions identified as fraud that actually turn out to be frauds. This is done by calculating the following :

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{14}$$

A high precision would imply that the model rarely incorrectly flags non-fraudulent transactions as fraudulent. On the other hand, a low precision would show that many normal transactions are incorrectly flagged as fraudulent.

F1-Score

The F1-Score is the harmonic mean of precision and recall, providing a balance between the two metrics. It is evaluated by :

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{15}$$

A high F1-Score usually depicts a good balance between precision and recall. While a low F1-Score, indicates poor balance with either low precision, low recall or both.

AUC-ROC

The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) is a metric that evaluates how well a model can differentiate between fraudulent and non-fraudulent transactions. The ROC curve itself is a plot that shows the TPR (y-axis) against the FPR (x-axis) at various threshold settings. Essentially, the AUC (Area Under the Curve) value represents the likelihood that the model will correctly identify a randomly chosen fraudulent transaction as more suspicious than a randomly chosen non-fraudulent transaction.

$$\text{AUC-ROC} = \int_0^1 \text{ROC}(x)\, dx \tag{16}$$

The AUC value is the area under the ROC curve. A high AUC value signifies that the model has a good ability to distinguish between fraudulent and non-fraudulent transactions. The AUC-ROC theoretically ranges from 0 to 1. However, since a random classifier's ROC curve forms a diagonal line from (0,0) to (1,1), its AUC is 0.5. Therefore, no realistic model will have an AUC below 0.5, and hence AUC-ROC values typically range from 0.5 to 1.

C Programming Code

The description below explains the code I used to replicate Meinshausen (2006)'s methodology for the Quantile Regression Forest:

- The 'Replication 1' R Source File uses the *quantregForest* package to train the model. The code contains details to plot the Fig.9. It uses the Boston Housing dataset that is available in R attained using package *MASS*.
- The 'Replication 2' R Source File also uses the *quantregForest* package and the Boston Housing dataset. The code contains details to plot the Fig.10.

Next, the list below explains the code I used to generate the results of this thesis (excluding the replication). For this part, the 'creditcard.csv' file was used as the primary dataset.

1. **Random Forest model**: the file 'Rf_new' file has the code used to execute this model.
    - The file includes the data preprocessing process before conducting any of the methods such as removing duplicates, renaming the 'Class' variable to 'isfraud' and removing the 'Time' column as mentioned in Section 3. It also has the code for the plots in Section 3.
    - The code then moves on to training the Random Forest model using the *randomForest* package. Further, the visualisation of the decision tree used in this method is obtained using the *rpart* package.
    - Next, the code PR curve for the results section (5) is provided. The true labels of the test data and the predicted probabilities that the Random Forest assigned are extracted from R- in file 'rf_predictions'. Then, the Python file 'PCR_curve' is used to visualise the PR curve. This is done for each method by replacing the name of the file in the data path.
    - Lastly, the code for the VIMP plot (Fig.7b) is provided.
2. **Adversarial Random Forest** uses the R source file 'Adversarial_new'.
    - **White Box Attack**: First, the code for using the FGSM and PGD methods is provided. Then, as carried out in this research the code proceeds to train the model by combining both methods by using *rbind* package. They were then trained using *randomForest* package.
    - As mentioned before for the PR curve, the file 'white_predictions' is generated using R and is used in the Python file 'PCR_curve' to visualise the plot.
    - **Black Box Attack**: To carry out the HSJA, the Python file 'Black_box' is used. The original dataset 'creditcard.csv' is cleaned again.
    - The Random Forest model is first trained on the training dataset. Next, the model is wrapped with the *Adversarial Robustness Toolbox (ART)* to simulate the adversarial attack.
    - The HSJA is initialised and used to generate adversarial examples from the test dataset. These adversarial examples are then saved and exported to a CSV file- 'adv_examples'.

- The Random Forest is then evaluated on classifying fraudulent and non-fraudulent transactions using the 'adv_examples' dataset. Again, as done previously the file 'black_predictions' is generated in R and is used in the Python file 'PCR_cruve' to visualise the plot.

3. **One-Class SVM**: the code for this method lies in the 'SVM_new' R Source file.

   - As mentioned in Sec.4.4, the code separates normal transactions for training the method. The features are then scaled and are used to train the model with the *e1071* package.
   - Unlike the other methods, the One-Class SVM cannot provide probabilities for class membership which is needed to plot the PR Curve.

4. **Random Forest Quantile Classifier**: the code to carry out this method can be found in the R Source file-'Quantile_new'.

   - To train the model the *imbalance* function from the *randomForestSRC* package is used.
   - Next, the VIMP plot is created as visualised in Fig.7a.
   - As mentioned before, the file 'quantile_predictions' is generated using R and is then used in the Python file 'PCR_curve' to plot the PR Curve.