The Impact of Machine Learning Techniques on Credit Card Fraud Detection: A Comparative Evaluation of Predictive Models

Christian Chukwuka

A Thesis Submitted in Partial Fulfilment

of the requirements for the

Degree of

Master of Science in Data Analytics

Diagram

Description automatically generated with medium confidence

February 2024

Supervisor: Kislay Raj

**CCT College Dublin**

Assessment Cover Page

*To be provided separately as a word doc for students to include with every submission*

|  |  |
| --- | --- |
| Module Title: | Capstone Project |
| Assessment Title: | The Impact of Machine Learning Techniques on Credit Card Fraud Detection: A Comparative Evaluation of Predictive Models |
| Supervisor Name: | Kislay Raj |
| Student Full Name: | Chukwuka Christian Anthony |
| Student Number: | Sba22175 |
| Assessment Due Date: | February, 2024 |
| Date of Submission: | 23th February, 2024 |

Chukwuka Christian Anthony

|  |
| --- |
| Declaration By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution. |

**Acknowledgment**

I want to express my gratitude to my family and my supervisor as I finish my MSc program. My spouse Juliet Chukwuka has been a source of support showing me love and patience every step of the way. My kids, Michael and Daniel Chukwuka inspire me daily with their happiness and understanding. The extended family, including my mother, siblings and, in-laws have all provided encouragement and faith in my abilities.

A special shoutout to my supervisor Kislay Raj for their expert guidance, patience and priceless insights that have greatly influenced my project. Your mentorship has not propelled my research forward. Also contributed to both my personal and professional development.

This accomplishment is not mine but a reflection of the support, sacrifices and love from both my family and supervisor. I am incredibly thankful to each one of you for your roles, in this journey.

# 

# Abstract

In the evolving world of finance credit card fraud presents a challenge that not only leads to substantial financial losses but also undermines consumer trust. This research aims to leverage machine learning (ML) techniques to enhance the identification of transactions, within credit card data. By comparing models such as Logistic Regression, Random Forest, Gradient Boosting, Support Vector Machines (SVM) and Decision Trees this study evaluates their effectiveness in detecting activities.

To effectively analyze the dataset, which includes both legitimate transaction records a thorough preprocessing phase is conducted. This involves normalizing the data identifying outliers and performing feature engineering to prepare it for analysis. Exploratory Data Analysis (EDA) is subsequently carried out to uncover any patterns or anomalies that might reveal transactions. The performance of each model is then assessed through training and validation processes using metrics like accuracy, precision, recall, F1 score and (ROC AUC) curve.

The findings from this research offer insights into the strengths and limitations of each ML technique when it comes to fraud detection. This nuanced understanding enhances the body of knowledge regarding how these techniques can be employed. Ensemble techniques, like Random Forest and Gradient Boosting have shown effectiveness by managing imbalanced data and capturing relationships between features.

The research emphasizes the importance of choosing features and tuning model parameters offering suggestions, for incorporating machine learning models into practical fraud detection systems. This project contributes to the discussion on security by highlighting the role of machine learning in combating credit card fraud. It serves as a resource for organizations seeking to enhance their fraud detection capabilities paving the way for exploration of advanced analytical techniques and their integration, into fraud prevention strategies.

Contents

[Abstract 5](#_Toc159533095)

[Chapter 1: Introduction 7](#_Toc159533096)

[1.1 Research Objectives 9](#_Toc159533097)

[1.2 Research question 9](#_Toc159533098)

[Chapter 2: Literature Review 9](#_Toc159533099)

[2.0 Introduction to Credit Card Fraud and Machine Learning 9](#_Toc159533100)

[2.1 Methodology 10](#_Toc159533101)

[2.2 Overview of Fraud Detection Techniques 11](#_Toc159533102)

[2.3 Machine Learning Algorithms in Credit Card Fraud Detection 11](#_Toc159533103)

[2.4 Performance Evaluation Metrics and Techniques 14](#_Toc159533104)

[2.5 Preventive Measures in Credit Card Fraud Detection 15](#_Toc159533105)

[2.6 Machine Learning Techniques in Big Data 15](#_Toc159533106)

[2.7 Challenges and Limitations in ML-based Fraud Detection 18](#_Toc159533107)

[2.8 Challenges and Emerging Trends in Machine Learning-Based Fraud Detection 20](#_Toc159533108)

[2.9 Gap Analysis and Future Research Directions 21](#_Toc159533109)

[Chapter 3: Methodology 23](#_Toc159533110)

[3.1 Type of Research 23](#_Toc159533111)

[3.2 Approach 24](#_Toc159533112)

[3.3 Study Setting 24](#_Toc159533113)

[3.4 Data Preprocessing and Exploratory Data Analysis (EDA) 24](#_Toc159533114)

[3.5 Exploratory Data Analysis (EDA) 25](#_Toc159533115)

[3.6 Data Transformation and Feature Engineering 25](#_Toc159533116)

[3.7 Model Selection and Training 26](#_Toc159533117)

[3.7.1 Machine Learning Algorithms and Evaluation 26](#_Toc159533118)

[3.8 Model Evaluation 27](#_Toc159533119)

[3.8.1 Post-Analysis 28](#_Toc159533120)

[3.9 Ethical Considerations 28](#_Toc159533121)

[Chapter 4: Discussion 29](#_Toc159533122)

[4.0 Introduction to the Discussion 29](#_Toc159533123)

[4.1 Recap of Research Objectives and Questions: 30](#_Toc159533124)

[4.2 Overview of Data Analysis 30](#_Toc159533125)

[4.3 Presentation of Findings 31](#_Toc159533126)

[4.4 In-Depth Model Analysis 36](#_Toc159533127)

[**4**.4.1 Hyperparameter Tuning Using RandomizedSearchCV 39](#_Toc159533128)

[4.4.2 Neural Networks 41](#_Toc159533129)

[4.4.3 Comparative Analysis 41](#_Toc159533130)

[4.4.4 Conclusions Comparative Analysis 44](#_Toc159533131)

[**4.4.5 Recommendations:** 44](#_Toc159533132)

[4.4.6 Visual Aids for Comparison: 45](#_Toc159533133)

[4.4.6 Interpreting the Bar Graph (Comparative Model Performance): 46](#_Toc159533134)

[4.4.7 Interpreting the Line Chart (ROC-AUC Scores by Model): 47](#_Toc159533135)

[4.5 Interpretation of Findings 49](#_Toc159533136)

[**4.5.1 Real-World Implications** 50](#_Toc159533137)

[**4.5.2 Limitations and Challenges** 51](#_Toc159533138)

[**4.5.3 Methodological Limitations** 51](#_Toc159533139)

[**4.5.4 Challenges** 52](#_Toc159533140)

[Chapter 5: Conclusion/Results 54](#_Toc159533141)

[5.0 Summary of Research and Key Findings 54](#_Toc159533142)

[5.2 Recommendations for Future Research 54](#_Toc159533143)

[5.3 Research Problem 55](#_Toc159533144)

[5.4 Research Questions 55](#_Toc159533145)

[5.6 Findings of the Research 55](#_Toc159533146)

[Model Performance and Effectiveness: 55](#_Toc159533147)

[5.7 Limitations and Challenges: 56](#_Toc159533148)

[5.8 Conclusion Drawn from the Research 57](#_Toc159533149)

[5.8.1 Effectiveness of Machine Learning: 57](#_Toc159533150)

[5.8.2 Importance of Data Quality and Feature Engineering: 57](#_Toc159533151)

[5.8.3 Challenge of Imbalanced Classes: 57](#_Toc159533152)

[**5.8.4** Adaptability and Continuous Learning: 57](#_Toc159533153)

[**5.8.5** **Practical Implementation Considerations:** 58](#_Toc159533154)

[5.8.6 Ethical and Regulatory Compliance: 58](#_Toc159533155)

[5.8.7 Future Prospects: 58](#_Toc159533156)

[5.8.8 Broader Implications: 58](#_Toc159533157)

[References: 59](#_Toc159533158)

[Dataset Link 67](#_Toc159533159)

[**Appendix -------- Code Using Jupiter Notebook** 67](#_Toc159533160)

# Chapter 1: Introduction

Introduction

In today’s era while credit card transactions offer convenience the looming concern of fraud, especially credit card fraud casts a shadow. This not only causes monetary losses, for individuals and financial institutions but also undermines consumer confidence, in digital payment systems. As methods of detecting fraud struggle to keep up with sophisticated and evolving fraudulent tactics the use of Machine Learning (ML) techniques has emerged as a promising solution. These advanced analytical tools have the potential to accurately and rapidly identify and prevent transactions. This research project aims to explore how machine learning techniques have revolutionized credit card fraud detection through an evaluation of predictive models.

The problem of credit card fraud has gotten more complex and constantly evolving as a result of deceitful activities. Traditional methods of detection which are primarily rule based struggle to keep up with the evolving patterns of fraud. This often leads to a number of positives where legitimate transactions are mistakenly flagged as fraudulent. This can cause dissatisfaction among customers and inefficiencies in operations. On the hand machine learning models offer a sophisticated and effective approach to identifying fraudulent transactions. By analysing transaction data for patterns and anomalies these models can adapt to fraud tactics, in time potentially reducing false positives and improving detection rates.

The main goal of this study is to compare Machine learning models for detecting credit card fraud. The aim is to determine which models strike the balance between accuracy in detecting fraud and operational efficiency. The models under scrutiny include regression, decision trees, random forests, neural networks and support vector machines.This evaluation will consider their performance, computational efficiency and practical usability, in real world scenarios. This examination will not discuss the pros and cons of each model. Also offer insights, into the obstacles faced when implementing machine learning fraud detection systems.

Additionally, this study will tackle the difficulties involved in utilizing machine learning methods, for detecting fraud, which include problems associated with imbalanced data understanding how models work and the implications of automated decision making. By presenting an examination of the existing state of machine learning in credit card fraud detection and presenting evidence from a comparative analysis of various models this project aims to provide valuable insights to enhance financial security and guide the creation of more efficient strategies, for preventing fraud.

In summary as the complexity and magnitude of credit card fraud continue to grow it becomes increasingly crucial to develop flexible detection methods. This project signifies progress in comprehending the capabilities of machine learning approaches in addressing this issue. It provides a way not to reduce losses but also to rebuild and strengthen trust in digital financial transactions.

## 1.1 Research Objectives

To Evaluate the Effectiveness of Various Machine Learning Techniques in Credit Card Fraud Detection

To Identify Key Predictive Features in Credit Card Fraud Detection

## 1.2 Research question

What are the most effective machine learning techniques for detecting credit card fraud in terms of accuracy, precision, and recall?

.

# Chapter 2: Literature Review

## 2.0 Introduction to Credit Card Fraud and Machine Learning

In today’s world credit card transactions have become a part of global commerce due, to their convenience and widespread usage. However, alongside this rise in popularity credit card fraud has also become more sophisticated and prevalent posing threats to security and consumer trust. Traditional methods of detecting fraud, which rely on rules and manual processes are proving inadequate against the tactics employed by fraudsters. There is an urgent requirement for dependable and flexible methods to identify and stop fraudulent operations. Machine learning (ML) is an innovative method employed to address credit card fraud. Machine learning algorithms can learn from data and adapt autonomously, unlike traditional systems. ML algorithms possess the capacity to learn from data and adjust as time progresses, unlike systems.. They can identify patterns and subtle anomalies that indicate behaviour. These algorithms offer the speed, accuracy and flexibility required to combat fraud techniques. Whether its through networks that mimic decision making or ensemble methods that combine multiple algorithms for better predictions ML provides a diverse range of tools, for proactive and nuanced fraud detection. Moreover, MLs impact goes beyond identifying transactions; it also works towards enhancing the customer experience by reducing false positives—legitimate transactions incorrectly flagged as fraudulent.

The objective of this literature study is to evaluate the influence of machine learning techniques on the identification of credit card fraud. The goal is to examine and compare models evaluate their effectiveness and efficiency and understand their strengths and limitations. This review will analyse research findings identify trends in the use of machine learning for fraud detection and suggest research directions. By offering an analysis of these technological tools this review aims to contribute to the ongoing discussion, on digital financial security and the broader field of AI and machine learning applications.

## 2.1 Methodology

Credit card fraud remains a concern within the financial industry prompting the development of various techniques to detect and prevent it. Delamaire et al. (2009) delve into the considerations surrounding credit card fraud and stress the importance of identifying different types of fraudulent activities to implement appropriate countermeasures. Several studies offer reviews of diverse fraud detection methods, such as Neural Networks, rule induction techniques, fuzzy systems, decision trees, Support Vector Machines, Artificial Immune Systems, genetic algorithms and K Nearest Neighbor algorithms (Zareapoor et al., 2012; Tripathi and Pavaskar 2012; Chaudhary et al., 2012). Sorournejad et al. (2016) classify these techniques based on their approach (misuse or anomaly detection). How they process data (numerical or categorical). These studies also discuss the criteria for evaluating performance and the datasets used in assessing detection methods effectiveness (Sorournejad et al., 2016; Zareapoor et al., 2012). Additionally, they highlight research directions and open issues in credit card fraud detection to emphasize the ongoing need, for exploring and improving fraud detection techniques continuously (Sorournejad et al., 2016).

Abdallah et al. (2018) present an overview of the challenges faced by fraud detection systems (FDSs) when securing electronic commerce systems.

The research investigates the limitations of fraud prevention systems (FPSs) and the potential advantages of collaborating them with FDSs. The authors examine five electronic commerce systems discussing types of fraud and advanced techniques used by FDSs for each system.

Quah and Sriganesh (2008) propose a method to identify potential instances of fraud in real time by analyzing spending patterns. Their approach involves utilizing a self-organization map to analyze and filter customer behavior for detecting activities. Srivastava et al. (2008) introduce Hidden Markov Simulation (HMS) as a means to simulate the stages involved in credit card transaction processing for detecting fraud.

Chan and Stolfo (1998) address the challenges posed by datasets with imbalanced class distributions and varying costs per error which are common in real world data mining tasks such, as credit card fraud detection. They suggest a learning approach using multiple classifiers that can significantly minimize losses resulting from unauthorized transactions.

## 2.2 Overview of Fraud Detection Techniques

Fraud detection methods often rely on analysing transaction data to identify patterns. In a review by Bolton and Hand (2002) they highlighted the effectiveness of rule based approaches derived from knowledge. Bhattacharyya et al. (2010) explored credit card fraud detection examining strategies like support vector machines (SVM) and random forests, as well as logistic regression. Their study delved into the world of credit card transactions to compare different data mining techniques used in combating fraud. In another study Williams and Huang (1997) introduced the hot spots methodology, which takes a multi strategy and interactive approach to uncover insights through data mining. By analyzing resulting models they were able to extract information.

Furthermore Chan et al. (1999) discussed the importance of distributed data mining in credit card fraud detection due to the increasing number of transactions and subsequent rise in stolen account numbers and bank losses. These studies highlight approaches employed in tackling credit card fraud using statistical methods that analyze transaction data, for abnormal patterns.

The researchers. Analyze methods that tackle the challenges of scalability, efficiency imbalanced training data and varying costs of errors at the same time. In a study by Şahin and Duman (2011) they utilize a dataset to evaluate how well SVM and decision tree algorithms can detect credit card fraud. This study stands as one of the endeavours to gauge the efficacy of these techniques, in identifying fraudulent credit card transactions.

## 2.3 Machine Learning Algorithms in Credit Card Fraud Detection

**Neural Networks**

In a study by Dorronsoro et al. (1997) they used a network-based approach to detect credit card fraud. Their model demonstrated exceptional performance in terms of both high detection rates and low false alarm rates. The authors ascribed the triumph of their network model to its capacity to acquire knowledge and adjust to developing fraud trends. Ghosh et al. (1994) conducted research on identifying activities using a system based on the neural network paradigm. This complex system was fine-tuned and calibrated by analyzing a dataset of categorized credit card transactions generously provided by a credit card institution.

Sharma (2011) aimed to detect fraudulent transactions by combining a neural network with a genetic algorithm. The authors suggested using algorithms to determine the optimal network structure, including the number of hidden layers and nodes for effective credit card fraud detection. Sadgali et al. (2019) conducted a study on various machine learning techniques for credit card fraud detection with an emphasis on neural networks. Their goal was to provide guidance in selecting the suitable techniques based on analyzing and comparing performance, across different methods.

**Strengths:** Excellent for capturing non-linear relationships, adaptable to various data types.

**Limitations:** "Black box" nature, requires large datasets and substantial computing power.

**Decision Trees and Ensemble Methods**

Decision trees are a type of machine learning model that divides data into subsets based on different features. This division process creates a tree structure, which is commonly used for classification purposes. Random forests on the hand take decision trees to the next level by creating multiple trees and combining their predictions.

In a study conducted by Syeda et al. (2002) they developed a neural network based model to detect credit card fraud. They compared the performance of this model with decision trees and random forests. Found that the neural network based approach outperformed both in terms of speed and accuracy. Another research conducted by Abdallah, Maarof and Zainal (2016) focused on exploring the challenges faced by Fraud Detection Systems (FDSs) in electronic commerce systems. They conducted a survey to identify these issues and obstacles.

To tackle some of these challenges Carcillo et al. (2018) introduced a solution called Scalable Real time Fraud Finder (SCARFF). SCARFF combines machine learning techniques with Big Data tools like Kafka, Spark and Cassandra. This integration allows SCARFF to effectively handle volumes of streaming data in real time settings leading to more efficient and accurate fraud detection, in electronic commerce systems.

**Support Vector Machines (SVM)**

Support vector machines (SVM) a type of machine learning have been proven effective for classification tasks such as fraud detection. The scalability and efficiency of SVMs in handling datasets were demonstrated by Chan et al. (1999) in a distributed data mining system designed for credit card fraud detection. In their approach Panigrahi et al. (2008) proposed a method that considers both current and past conduct to detect credit card fraud. Their fraud detection system (FDS) consists of four components; a rule based filter, Dempster–Shafer adder, transaction history database and Bayesian learner. Manira et al. (2019) focused on utilizing machine learning and data science techniques to identify credit card transactions with the goal of detecting 100% of fraudulent transactions while minimizing false positives.

**Strengths:** Effective in high-dimensional spaces, robust against overfitting.

**Limitations:** Requires careful tuning of parameters, not suitable for larger datasets.

**Random Forest**

In a study conducted by Krivko (2010) they proposed an approach to detect plastic card fraud. Their hybrid model combined. Unsupervised methodologies to address the limitations of each method individually. The author demonstrated the effectiveness of this model by identifying fraudulent activity in real debit card transactions. They also compared its efficiency to an existing monitoring system used by a collaborating bank using appropriate performance assessment criteria.

**Strengths:** Reduces overfitting, good for large datasets.

**Limitations:** Less interpretable, can be biased in the presence of categorical variables.

**Gradient Boosting**

The paper published by Chen et al.( 2016) introduces XGBoost, a tree boosting system that is widely adopted in machine learning due, to its scalability and efficiency. The paper highlights the effectiveness of XGBoost emphasizing its ability to handle datasets using resources. It introduces advancements such as a sparsity algorithm designed for sparse data and an approximate tree learning method that utilizes weighted quantile sketch. Additionally, the paper discusses system optimizations like cache access patterns and data compression. XGBoost has gained recognition for achieving state of the art results making it a notable breakthrough, in machine learning for large scale applications. Noviandy, T. R., et al. (2023) explores the use of the XGBoost algorithm and data augmentation techniques, highlighting the effectiveness of these methods in handling imbalanced datasets and improving fraud detection accuracy.

**Strengths:** High predictive accuracy, effective with various loss functions.

**Limitations**: Computationally intensive, can overfit on noisy datasets.

The authors of this study utilized available datasets to propose a fresh approach in classifying credit risk algorithms driven by machine learning. In their research Iyer et al. (2011) introduced a Hidden Markov Model (HMM) for detecting credit card fraud. By training the HMM with cardholder behavior and assessing incoming credit card transactions those that didn't meet a high enough probability were marked as potentially fraudulent. The study also provided a framework, for addressing the challenges of fraud detection systems and identifying effective performance metrics.

## 2.4 Performance Evaluation Metrics and Techniques

When assessing the effectiveness of fraud detection models it is important to use metrics. There are measurements commonly used, such as accuracy precision (the percentage of identified fraud cases) recall (the percentage of accurately detected fraud cases, out of all actual instances) and the harmonic mean of precision and recall. Powers (2011) emphasized the significance of these metrics in datasets with imbalanced proportions between legitimate transactions, which is often the case in fraud detection scenarios. To evaluate model performance the dataset is divided into subsets called folds. The model is trained on all but one fold. Then evaluated on that remaining fold. This process is repeated for each fold. The results are averaged to provide a performance estimate. Kohavi (1995) extensively discussed validation techniques and their usefulness in obtaining unbiased performance estimates for machine learning models, including those used in fraud detection. ..;Operating characteristic (ROC) curves and area under the curve (AUC)'re commonly employed evaluation techniques for binary classification problems, like fraud detection….. ROC curves are used to plot the sensitivity ( rate) against the specificity (false positive rate) at different thresholds, in classification. On the hand the AUC is a measure of how a classifier performs across all thresholds. In his work from 2006 Fawcett highlighted the significance of ROC curves and AUC in evaluating machine learning models especially when it comes to fraud detection. It's crucial to strike a balance, between minimizing positives while maximizing positives in this context

## 2.5 Preventive Measures in Credit Card Fraud Detection

MFA, which stands for Multi Factor Authentication is a security technique that requires users to provide forms of identification to confirm their identity. This method helps ensure that sensitive information remains protected and reduces the chances of credit card theft. To enhance security measures, Jain et al. (2004) explored biometric identification technologies, like fingerprint, face recognition and iris recognition. By making it more difficult for fraudsters to gain access to user accounts the implementation of MFA in credit card transactions significantly minimizes the risk of fraud. Deploying MFA in credit card transactions can greatly reduce the potential for fraud since it poses a challenge for fraudsters attempting to breach user accounts. Dynamic risk scoring involves evaluating the level of risk associated with each transaction based on factors such as transaction amount, location and user behavior patterns. This approach aids in identifying high risk transactions and activating security measures like requesting authentication or blocking suspicious transactions. Sahin et al. (2013) proposed a risk scoring model specifically designed for detecting credit card fraud, which demonstrated accuracy in identifying fraudulent transactions. Integrating dynamic risk scoring, into credit card fraud detection systems helps prevent fraud by recognizing and addressing high risk transactions. Transaction monitoring (Duman & Ozcelik 2011);Continuous monitoring of user transactions is crucial to identify any suspicious patterns that may suggest activity. This important process aids in the detection and prevention of fraud by recognizing threats and taking appropriate actions. An interesting study conducted by Duman & Ozcelik (2011) showcased the development of a credit card fraud detection system using networks and genetic algorithms. The system also incorporated a transaction monitoring component, which effectively detected transactions. Integrating transaction monitoring, into fraud detection systems offers real time insights, into user behaviour enabling identification and prevention of activities.

## 2.6 Machine Learning Techniques in Big Data

Feature selection and engineering play a role in improving the effectiveness of machine learning models in fraud detection (Liu et al., 2011). Various techniques, such as correlation analysis (Liu et al. 2011) information (Peng et al., 2005) and recursive feature elimination (Guyon et al., 2002) have been employed to identify the most relevant features for detecting fraud. In a study by Pan (2020) a classification based method that follows patterns was proposed to identify fraudulent firms showing promise in enhancing audit quality. Kotsiantis et al. (2007) investigated the efficacy of forecasting systems and ensemble methods in detecting firms involved in fraudulent financial practices emphasizing the significance of financial ratios in identifying fraud. Ravisankar et al. (2011) compared data mining techniques for detecting financial statement fraud and found that Probabilistic Neural Networks (PNN) outperformed other approaches. Iyer et al. (2011) introduced a method for credit card fraud detection based on Hidden Markov Models (HMM). Their approach evaluates credit card transactions, against trained HMM probabilities flagging potentially fraudulent transactions if they deviate significantly from these probabilities.

Firstly the Hidden Markov Model (HMM) is adjusted based on a representation of the cardholders transaction patterns. This approach captures the unpredictable nature of human financial behavior emphasizing the significance of speed and efficiency, in time sensitive operations.

**Scalable machine learning algorithms :** Scalable machine learning algorithms play a role in handling the challenges of big data analytics. Mohammed et al. (2018) demonstrated the criticality of scalability when dealing with imbalanced data particularly in credit card fraud detection. Baldominos et al. (2014) proposed an architecture for machine learning that offers real time predictions and analytics as a service specifically designed to handle massive amounts of data. Yui and Kojima (2013) suggested an approach that combines database management systems with Hadoop to tackle large scale machine learning tasks while Gupta et al. (2016) conducted an analysis of parallel data intensive machine learning techniques. To ensure big data applications operate efficiently. Effectively it is essential to continue researching and developing scalable machine learning techniques.

**Distributed computing:** Distributed computing systems like Hadoop and MapReduce are components of handling large datasets. Lamport and Lynch (1990) provide insights into distributed computing models and complexity measures building a foundation for understanding distributed systems. Shvachko et al. (2010) describe the Hadoop Distributed File System, which offers storage and computing capabilities for processing big data efficiently. White (2012) provides a guide, on utilizing Hadoop to process large datasets highlighting its various components and benefits.

In their research Condie et al. (2010) suggest a revised version of the MapReduce architecture that allows for aggregation and continuous queries. This modification expands the capabilities of the model, beyond batch processing. Dean and Ghemawat (2008) on the hand introduce the MapReduce programming model and highlight its scalability fault tolerance and user friendly nature. Lastly Zaharia et al. (2010) present Spark, a framework designed specifically for iterative machine learning and interactive data analysis purposes. Interestingly Spark has been shown to outperform Hadoop in use cases.

**Real-Time Fraud Detection Systems:** Real time systems, for detecting fraud have become increasingly important due to the need to take action in order to prevent financial losses (Kolajo et al., 2015). Quah and Sriganesh (2008) propose a method to understand spending patterns enabling the identification of fraud cases in real time. Their approach involves analyzing and filtering customer behavior using a self-organization map for fraud detection. Gama et al. (2004) introduced a framework for learning in the context of concept drift, which's a common challenge in real time fraud detection systems. Their framework allows integration of data into existing models ensuring continuous adaptation of the system to the ever changing environment.

**Real-time stream processing systems:** Stonebraker et al. (2005) discuss eight requirements that're essential for systems to excel at processing real time streams providing guidance for evaluating stream processing solutions at a high level. Gulisano et al. (2012) present Stream Cloud, an scalable stream processing engine that can handle volumes of data streams by utilizing an innovative parallelization technique. Chintapalli et al. (2016) have developed a streaming benchmark that compares Flink, Storm and Spark Streaming offering performance comparisons in terms of latency and throughput, across configurations.

According to Ranjan (2014) there is a growing recognition of the need, for scalable stream analysis due to the increase in data transmitted over the Internet. Karimov et al. (2019) propose a methodology for evaluating distributed stream processing engines focusing on their throughput and latency in performing windowed operations. They compare Apache Storm, Apache Spark and Apache Flink using this approach. Brito et al. (2011) introduce Stream MapReduce as a data processing technique that combines the MapReduce paradigm with Event Stream Processing enabling real time processing of data streams, with latency and scalability.

**Big Data Technologies in Fraud Detection:**Various technologies, like Hadoop, MapReduce, Apache Spark and NoSQL databases have been widely used to handle amounts of data in fraud detection. These technologies provide efficient processing capabilities for transaction data. To address the challenges of dealing with graphs Malewicz et al. Introduced Pregel, a model that follows a vertex centric approach for large scale graph processing. In a vein Xin et al. Developed Spark as a framework that overcomes the limitations of MapReduce by enabling utilization of working sets resulting in improved performance for iterative machine learning tasks. In order to assess and compare the performance of data systems, Ghazal et al. Created Big Bench, as a benchmarking tool specifically designed for data analytics. Interestingly Pavlo et al.s research revealed that parallel SQL database administration solutions outperformed MapReduce in terms of performance when it comes to large scale data processing – which emphasizes the value of established approaches. The works presented by Shvachko et al. (2010). Ghemawat et al. (2003) also contribute to this field. Dependable and high performance storage systems, for managing data sets were highlighted through the introduction of distributed file systems like the Hadoop Distributed File System (HDFS) and the Google File System. In a study by Esenogho et al. (2022) a unique approach combining a network ensemble classifier with a hybrid data resampling method was proposed for detecting credit card fraud. They utilized boosting (AdaBoost) technique alongside a short term memory (LSTM) neural network as the base learner showcasing its effectiveness using real world credit card transaction datasets

## 2.7 Challenges and Limitations in ML-based Fraud Detection

Machine learning has become a tool, in combating fraud in industries like banking, e commerce and telecommunications. However, there are challenges that hinder the effectiveness and efficiency of machine learning based fraud detection systems. These challenges encompass data related issues, real time processing requirements and the evolving tactics employed by fraudsters.

**Data Imbalance**

In fraud detection the number of transactions far exceeds ones resulting in imbalanced datasets. This creates a challenge as most machine learning algorithms assume a number of examples for each class and may exhibit bias towards predicting the majority class. Techniques such as Synthetic Minority Over sampling Technique (SMOTE) Adaptive Synthetic (ADASYN) sampling and anomaly detection have been developed to address this issue by either resampling the dataset or adapting the learning process to better recognize minority classes.

**Real-Time Processing**

Detecting and responding to transactions in time is vital to minimize losses. Achieving this requires scalable models capable of handling large volumes of transactions at high speed. Implementing machine learning models in distributed computing environments like Apache Spark or using algorithms that require computational time can help address these challenges. Nevertheless there remains a trade-off, between speed and accuracy since more complex models that offer accuracy often demand greater computational resources and time.

**2.7.3 Evolving Fraud Tactics**

Fraudsters constantly adapt and devise strategies to avoid detection. This necessitates that fraud detection systems be equally adaptive. The dynamic nature of fraud poses a challenge as it can quickly render effective models obsolete.

Different methods are utilized to tackle the issue of fraud detection. Techniques, like online learning, where the model updates itself with data and ensemble methods that combine models to enhance accuracy and adaptability are employed. However, there is always a delay between identifying types of fraud and updating the models to catch them. During this time fraudsters can take advantage of the system.

**Model Interpretability**

One challenge in fraud detection is that as machine learning models become more complex, they become less understandable. This poses a problem in industries like finance where regulations demand explanations for decisions made by these systems. Interpretable models play a role in building trust with users and regulators and understanding the decision-making process. Methods such as LIME (Local Interpretable Model Explanations) or SHAP (SHapley Additive exPlanations) provide insights into model predictions. However, there's often a trade off, between interpretability and accuracy as simpler models tend to be more understandable but less precise (Gunning & Aha, 2019)

**Data Quality**

Ensuring high quality data is crucial for training reliable models. Problems like missing values, outliers and erroneous entries can significantly impact results. Lead to predictions. Maintaining data quality involves processes such, as data cleaning, normalization and transformation which can be complex and resource intensive.

**Privacy Concerns**

Protecting financial information, in fraud detection systems is crucial but challenging especially with the increasing regulatory requirements like the General Data Protection Regulation (GDPR). To safeguard user data techniques such as data anonymization, encryption and differential privacy are employed. However, these measures can also complicate the data processing and model training processes. Techniques such as data anonymization, encryption, and differential privacy are employed to protect user data, but they can also complicate the data processing and model training processes Goodman et al,( 2017).

## 2.8 Challenges and Emerging Trends in Machine Learning-Based Fraud Detection

The field of fraud detection using machine learning is continuously evolving, with the emergence of technologies and methodologies aimed at addressing the challenges. Here we present a summary of some advancements and emerging trends in this domain.

**Deep Learning and Complex Models**

Deep learning, a subset of machine learning has brought about transformations in fields, including fraud detection. Complex neural networks such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have gained popularity for their ability to detect patterns of fraud that simpler models might overlook. These networks can autonomously learn representations and features from data making them particularly suited for handling data like images, text and transaction sequences (LeCun et al., 2015).

One area where deep learning has shown promise is analyzing sequences of transactions. For example RNNs and their variations, like Long Short Term Memory (LSTM) networks can analyze user transaction sequences to identify activities based on irregularities. Although deep learning models possess capabilities they do require datasets and substantial computational resources. They are also often criticized for their lack of interpretability.

**Big Data and Analytics**

The rise, in transactions has resulted in an influx of data, which falls under the category of big data. Combining data analytics with machine learning greatly enhances models for fraud detection. By having access to data these models can recognize patterns more effectively and reduce both false positives and negatives. Technologies like Hadoop and Spark have made it possible to efficiently store and process this amount of information (Zaharia et al., 2010).

Additionally big data enables the utilization of techniques for feature engineering and anomaly detection. By analyzing a range of attributes and historical data models can make precise predictions. However managing and processing these datasets also poses challenges in terms of storage, computation and ensuring the quality of the data.

**Federated Learning and Privacy Preservation**

In light of growing concerns about privacy and safeguarding information federated learning has emerged as an innovative approach. This method allows for model training on users devices ensuring that private data never leaves their device (Konečný et al., 2016). Only updates to the model itself are transmitted to a server, for aggregation without exposing any raw user information.

This approach has benefits. It not protects user privacy. Also allows the model to learn from a wide range of real world data, which can potentially enhance its accuracy and reliability. However federated learning brings challenges in terms of coordination, communication overhead and maintaining model quality across distributed nodes.

**Regulatory Compliance**

With the increasing prevalence of machine learning models, in fraud detection it becomes crucial to navigate the landscape. Regulations such as the General Data Protection Regulation (GDPR) in Europe have imposed rules on data usage and consumer rights, including the right to explanation. These regulations require models to not be accurate but also transparent and understandable.

As a response to this challenge there has been a growing focus on developing machine learning models and techniques that can provide insights into how decisions made. The field of Explainable AI (XAI) is emerging with the goal of making AI decisions for humans. This ensures that machine learning models can be used responsiblyand transparently in applications, like fraud detection (Adadi & Berrada 2018).

## 2.9 Gap Analysis and Future Research Directions

The field of machine learning (ML), for fraud detection has made progress. There are still areas that need further exploration and improvement. One key area is the development of versatile models that can effectively detect fraud across domains or types. Many existing models are highly specialized. May not perform outside their specific training datasets or fraud scenarios limiting their applicability in various sectors or evolving fraud situations.

Another important aspect that requires attention is the advancement of semi supervised learning techniques. While supervised learning currently dominates fraud detection it heavily relies on labeled data, which can be scarce or outdated in the context of fraud. Developing methods that can detect anomalies or identify patterns of fraud without labelled data is an underdeveloped area that needs more focus.

Furthermore there is a need to address the integration of explainability and interpretability in models. As models become more accurate they often become less interpretable posing a challenge in regulated industries where clear explanations for decisions are required.

In summary further research should concentrate on developing generalized fraud detection models with capabilities across domains exploring semi supervised learning techniques to reduce reliance on labeled data and ensuring that complex models maintain interpretability, for regulatory compliance purposes.

Currently explainable AI (XAI) methods are not fully. Often fail to provide the level of detail or reliability, for making critical financial decisions (Ribeiro et al., 2016).

Moreover due to the nature of fraud patterns change rapidly. Unfortunately many machine learning models struggle to adapt resulting in outdated or ineffective fraud detection over time. It is crucial to conduct research on adaptive and continuously learning systems that can evolve alongside evolving fraud patterns.

In terms of research potential there are directions that ML based fraud detection can explore to address these gaps. One promising area is the advancement of transfer learning and meta learning models of generalizing across types of fraud and domains. These models have the potential to reduce the need for retraining and customization for applications making fraud detection systems more flexible and widely applicable.

Another area worth focusing on is enhancing semi learning techniques for fraud detection. Developing methods that can effectively identify types of fraud without relying on labeled data would greatly enhance the ability to respond to evolving fraudulent tactics.

Furthermore, improving the explainability and interpretability of models remains an area, for future research. This includes not the creation of techniques to enhance model transparency but also the seamless integration of these techniques, into the model development process ensuring they enhance performance instead of compromising it.

# Chapter 3: Methodology

The methodology chapter holds significance in this research project as it lays out the plan, for how the study will be carried out. This chapter offers an account of the research design methods of data collection, analysis plan and ethical considerations. It ensures that the research is conducted systematically can be replicated and adheres to standards. The methodology forms the core of the study by bridging the framework with investigation enabling effective and efficient answers to research questions.

In this study titled "The Impact of Machine Learning Techniques on Credit Card Fraud Detection: A Comparative Evaluation of Predictive Models " which have specifically designed a methodology chapter to outline the approach in evaluating machine learning models, for detecting credit card fraud. Given that machine learning and fraud detection involve data that have been employed a research design. This approach aligns with the study’s objectives, which revolve around analysing transaction data to identify patterns indicating activities Creswell et al. (2017).

This approach was chosen because it allows for objective and rigorous testing of hypotheses. The application of analysis, to a volume of transaction data allows for the identification of normal and anomalous patterns distinguishing between fraudulent and non-fraudulent transactions (Foster, 2017). Moreover, the quantitative approach provides an concise way to measure and analyze data numerically enabling decision making (Black, 2019).

To ensure the research reflects real world complexities in detecting credit card fraud, a secondary dataset from Kaggle will be utilized. This dataset is extensive and comprehensive making it ideal, for evaluating machine learning techniques and contributing to fraud detection strategies.

## 3.1 Type of Research

The study adopts a quantitative research design, focusing on the statistical analysis of numerical data derived from credit card transactions. This approach is particularly suitable for the empirical nature of transaction data, which encompasses continuous and categorical variables representing various aspects of cardholder transactions. Quantitative research is essential in this context as it allows for the precise, objective measurement and analysis of variables, facilitating the identification of patterns and anomalies indicative of fraudulent activities Creswell et al , (2017).

## 3.2 Approach

The quantitative method is chosen because it can provide data driven insights into patterns that indicate activity. It uses computational techniques to analyze amounts of transaction data, which helps identify both normal and unusual patterns. This approach is especially effective when dealing with the complexities and subtleties of credit card transaction data as it requires techniques to accurately detect fraud. By using an approach, the research aims to determine how effective, efficient and adaptable different machine learning techniques are, in detecting fraud. This will establish a framework for testing hypotheses. Validating results, against established metrics (Black, 2019).

## 3.3 Study Setting

The study makes use of a dataset acquired from Kaggle, the "Credit Card Fraud Detection" dataset. This dataset contains transactions made by cardholders, in Europe, which have been anonymized to protect user privacy. It includes both fraudulent transactions reflecting real world transaction data. The dataset is characterized by its imbalanced nature, which's typical of fraud detection datasets. By using this dataset the study aims to provide an realistic context for examining credit card fraud. This allows the studys findings to be compared directly with studies that have used the dataset thus contributing to the collective knowledge, in this field (Bolton & Hand 2002).

## 3.4 Data Preprocessing and Exploratory Data Analysis (EDA)

**Dataset Overview:** The dataset contains 284,807 transactions, each with 31 features. The features include time, amount, 28 anonymized PCA features (V1-V28), and a class label indicating fraud.

**Data Cleaning**: The first stage involves cleaning the data to ensure quality and consistency. Initial checks for missing values revealed no missing data, allowing for straightforward analysis without the need for imputation.This includes handling missing values, removing duplicates, and correcting errors. Given the nature of transaction data, it is also crucial to ensure that the data is free from outliers or anomalies that could skew the results (García et al., 2015).

**Descriptive Statistics**: Summary statistics provided insights into the central tendency, dispersion, and overall distribution of each feature.

**Feature Selection and Engineering**: This stage involves selecting the most relevant features for detecting fraud and potentially creating new features that might improve model performance. Techniques like principal component analysis (PCA) or correlation analysis are often used to reduce dimensionality and identify the most informative features (Guyon & Elisseeff, 2003).

**Data Transformation**: The data may need to be transformed or normalized to ensure that the algorithms function correctly. This might involve scaling features so they have a similar range or encoding categorical variables.

## 3.5 Exploratory Data Analysis (EDA)

Data visualization can be a valuable step to gain insights into the dataset and understand its characteristics. Visualization techniques applied were:

**Histograms and Box Plots**: The distribution of a subset of features (V1, V10, V20) was examined using histograms and box plots, revealing some features' normal distribution tendencies and others with significant outliers.

**Scatter Plot Analysis**: A scatter plot between V1 and V10 indicated no clear linear relationship, with a concentration of data points around the center, highlighting the variability and potential outliers within the dataset.

**Class Distribution**: A bar plot analysis of the 'Class' variable revealed a significant imbalance, with fraudulent transactions (Class 1) being a small fraction compared to legitimate transactions (Class 0), indicating the dataset's highly imbalanced nature.

## 3.6 Data Transformation and Feature Engineering

The dataset went through a process of transformation to introduce features that might reveal patterns indicating fraudulent activity. This process encompassed the implementation of the following set of rules;

**Rule 1: Unusual Increase in Transaction Amount**

In order to identify variations, in transaction amounts we computed the Z score for each transaction using the NormalizedAmount. We established a threshold to flag transactions as increases if their Z score exceeds 3. This approach assumes that transaction amounts adhere to a distribution, where a Z score surpassing 3 indicates a deviation three deviations away, from the average indicating an anomaly.

**Rule 2: Multiple Failed Login Attempts**

Taking into account the patterns of behaviour commonly associated with activities a new functionality was implemented to record instances of unsuccessful login attempts. Although the dataset itself did not contain information, on failed login attempts this particular rule aims to incorporate a feature that acknowledges the correlation, between such attempts and fraudulent behaviour.

**Rule 3: High Transaction Amount During Night Hours**

Fraudulent transactions frequently happen during hours to avoid being noticed. To address this we implemented a method that calculates the HourOfDay based on the transaction timestamp and determines if IsNight falls within the period (, between 10;00 PM and 6;00 AM). This additional binary feature enhances our model by incorporating data, which could potentially reveal a connection, between nighttime transactions and increased fraud risk.

**Feature Removal**

After normalization we decided to remove the Amount feature to prevent redundancy and any potential bias, towards higher transaction amounts.

## 3.7 Model Selection and Training

**Algorithm Selection**: Choose appropriate machine learning algorithms based on the nature of the data and the research objectives. Common choices for fraud detection include logistic regression, decision trees, random forests, support vector machines, and neural networks.

### 3.7.1 Machine Learning Algorithms and Evaluation

The selection and implementation of machine learning algorithms for fraud detection depend

on the specific requirements of the problem and the characteristics of the dataset. In this

research, the following algorithms were applied:

• Logistic Regression: This algorithm is suitable for binary classification tasks and

can provide interpretable results.

:

• Decision Trees: Decision trees can capture non-linear relationships and are

effective in handling categorical features.

• Random Forest: This ensemble method combines multiple decision trees to improve

accuracy and handle complex fraud patterns.

• Support Vector Machines (SVM): SVMs can handle high-dimensional data and are

effective in separating classes with a clear margin.

• Neural Networks : Neural networks are particularly good at solving problems that involve complex, non-linear relationships, like image recognition, speech recognition, and natural language processing. They can also be used for regression tasks, classification, and even generative tasks with the right architecture,

The Five algorithms were used to be able to establish the best possible result, and the associated algorithm as well as the applicable hyperparameters.

**Hyperparameter Tuning**: RandomizedSearchCV was employed to fine-tune the models, leading to improved accuracy and model robustness.

**Training**: The selected models are trained on a subset of the data. This involves feeding the algorithms with the features and labels from the training set and allowing them to learn the patterns indicative of fraud.

## 3.8 Model Evaluation

**Cross-Validation**: Implement cross-validation techniques to assess how the models will generalize to an independent dataset. K-fold cross-validation is a common approach where the data is divided into k subsets, and the model is trained and tested k times, each time with a different subset as the test set (Kohavi, 1995).

**Performance Metrics**: Evaluate the models using appropriate performance metrics. In the context of frau

d detection, accuracy, precision, recall, F1-score, and the area under the ROC curve (AUC-ROC) are critical metrics. Each provides different insights into the model's performance, from overall accuracy to the balance between false positives and false negatives (Powers, 2011).

**Model Comparison**: Compare the performance of different models to determine which is most effective for fraud detection. This involves looking at the metrics for each model and considering the trade-offs between them.

### 3.8.1 Post-Analysis

**Interpretation**: Interpret the results in the context of the research questions. This involves not only looking at which model performed best but understanding why. Consider the characteristics of the data, the nature of the models, and any external factors that might influence the results.

**Validation**: Validate the findings by comparing them with existing research or conducting additional analysis. This might involve using a different dataset, applying the models to a real-world scenario, or consulting with domain experts.

**Reporting**: Clearly report the findings, including the methodology used, the results of the analysis, and the interpretation of those results. Transparency is key to ensuring that the research is replicable and credible.

## 3.9 Ethical Considerations

In conducting research, especially involving sensitive data like credit card transactions, ethical considerations are paramount to ensure the integrity of the study and the protection of individuals' privacy. For the project on "The Impact of Machine Learning Techniques on Credit Card Fraud Detection," several ethical aspects must be addressed:

**Data Privacy and Confidentiality:** Handling credit card transaction data carries a risk of violating privacy and confidentiality. To safeguard individuals identities it is crucial to anonymize all data before analysis. Any identifiable information should be either completely. Transformed in a way that makes individual identification impossible.

**Data Security**; Ensuring the security of data is paramount. The project should implement data security measures such as encrypting stored data using secure networks for data transmission and enforcing strict access controls. Additionally all researchers involved in the project should receive training on practices, for data security.

**Informed Consent**; Although credit card data is often anonymized and doesn't directly involve subjects it's still important to consider the principle of informed consent. Addressing these considerations is crucial particularly during the data collection stage. It is essential for entities like banks or financial institutions to obtain consent from cardholders before using their data for research purposes

**Use of Findings**: When it comes to utilizing the findings of the research it's important to adhere to the stated research objectives. They should not be exploited in any way that could unfairly disadvantage or discriminate against individuals or groups.

**Transparency and Openness:** To ensure transparency and openness it is necessary to outline and disclose the methodology employed in the research. This includes being upfront about any limitations, potential biases or errors that might exist. Additionally, openly revealing the source and nature of the data used is an aspect of maintaining transparency.

**Compliance with Legal and Regulatory Standards:** The project must also comply with all legal and regulatory standards, such as data protection laws and industry specific regulations. Seeking guidance from experts or involving institutional review boards can be beneficial in ensuring compliance, across all aspects of the research.

# Chapter 4: Discussion

## 4.0 Introduction to the Discussion

**Purpose of the Chapter**: The objective of this research's fourth chapter is to methodically provide the results of the in-depth examination of numerous machine learning strategies used to identify credit card fraud. The goal of this chapter is to condense the vast amounts of information, measurements, and insights obtained from the comparative assessment into a comprehensible story that offers precise responses to the issues and goals of the research that were stated in the preceding parts. By concentrating on the findings, the study aims to shed light on the efficacy, efficiency, and usefulness of every model and technique employed.

## 4.1 Recap of Research Objectives and Questions:

In order to examine the outcomes, it is pertinent to recall the driving forces behind this research – the objectives and questions that have guided this investigative journey:

1. **To Evaluate the Effectiveness of Various Machine Learning Techniques in Credit Card Fraud Detection**: It sought to understand how different models compare in identifying fraudulent transactions, particularly focusing on accuracy, precision, recall, and other pertinent performance metrics.
2. **To Identify Key Predictive Features in Credit Card Fraud Detection**: Recognizing the most influential factors in predicting fraud is crucial for model interpretation and Aligned with these objectives, the research was steered by the following questions:
3. What are the most effective machine learning techniques for detecting credit card fraud in terms of accuracy, precision, and recall?

## 4.2 Overview of Data Analysis

The research utilized a set of data to identify instances of credit card fraud. This dataset consisted of transactions made by credit cardholders. Was carefully collected from a known source guaranteeing the reliability and relevance of the information used in this study. Each transaction is described by characteristics that were obtained through a PCA transformation. This process helps protect the identity of individuals involved while also reducing the complexity of the data all while preserving its properties. The features considered include V1 to V28 representing aspects of each transaction along, with information about the transaction amount and whether it was classified as fraudulent or not. Before conducting any analysis extensive preprocessing steps were undertaken to ensure that the data was properly prepared. These steps included normalizing the transaction amounts to maintain consistency, in scale and addressing any inconsistent data points in order to safeguard the accuracy and reliability of the analysis.

Explaination to pca ,details step feature importance ,describe precossing of features

**Methodological Recap**

The methodical procedure comprised applying machine learning models—each chosen for its efficacy—to classification tasks related to anomaly detection, such as credit card fraud. A variety of models were used, including decision trees, networks, logistic regression, random forests, support vector machines (SVM), and gradient boosting. Accuracy, precision, recall, F1 score, ROC AUC scores, and other metrics will be carefully reviewed in order to assess each model's performance. In order to make sure the models worked properly and weren't overfitting to particular data subsets, it mostly depended on validation procedures. In order to find indications of transactions, research was also done on feature importance across different models. This investigation shed light on the traits and behaviors connected to credit card fraud.

## 4.3 Presentation of Findings

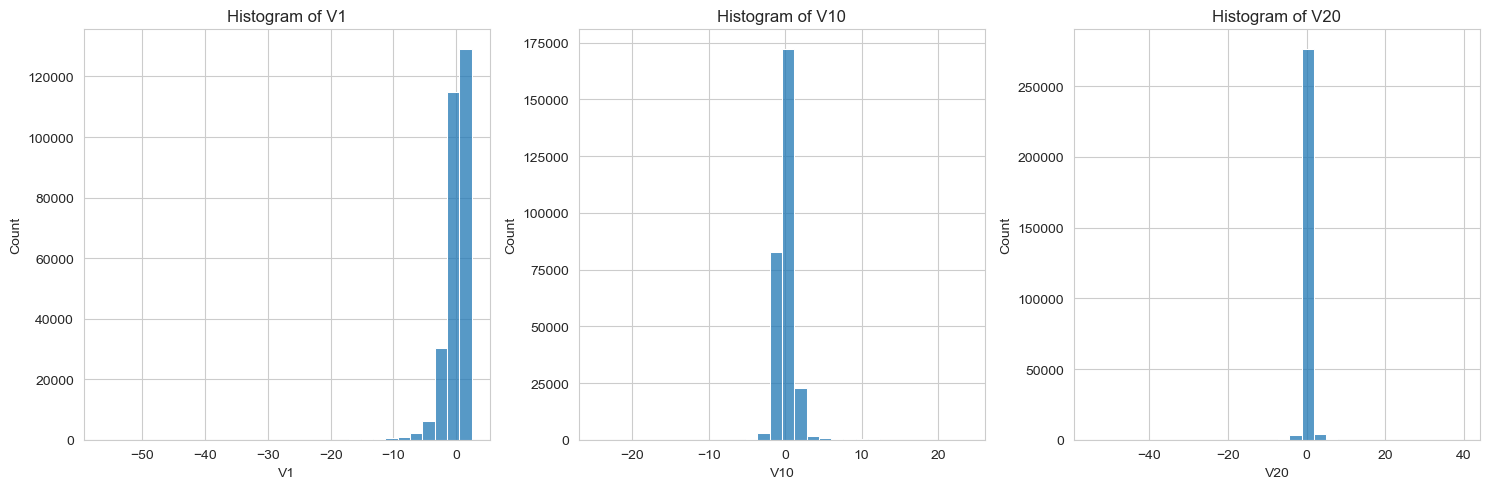
****

Figure .1

V1: Appears to be somewhat normally distributed with a slight skew.

V10: This feature shows a peak around the center but with a long tail, suggesting some outliers.

V20: Similar to V10, it has a central peak with extended tails, indicating potential outliers.

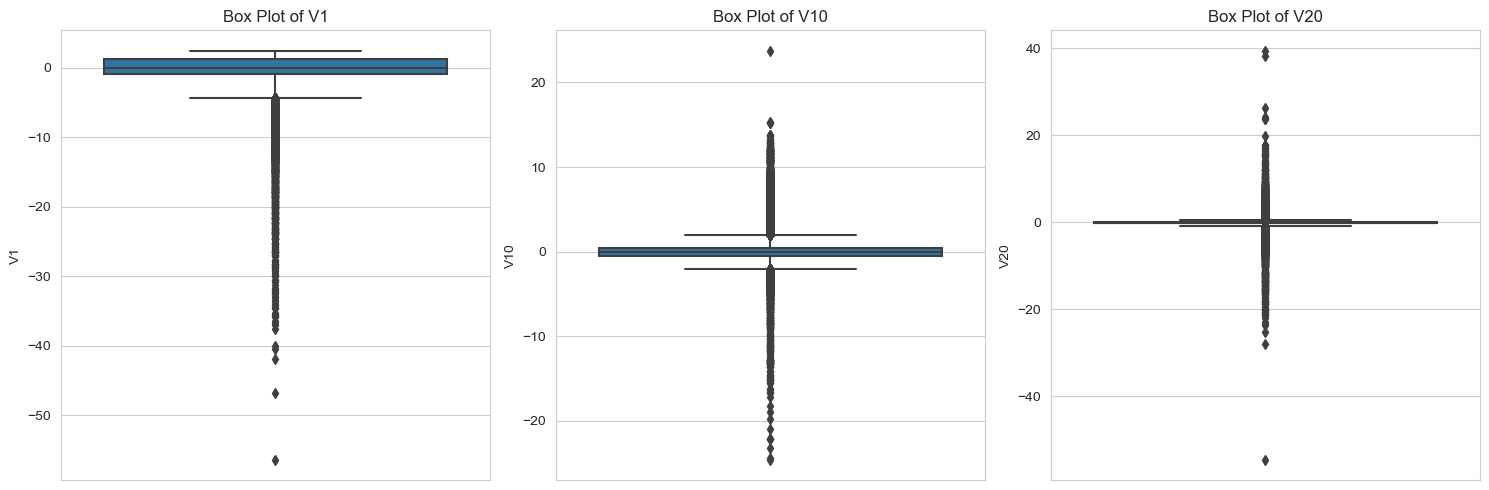


Figure 2

All three features display a significant number of outliers, as evidenced by the points outside the whiskers of the box plots.

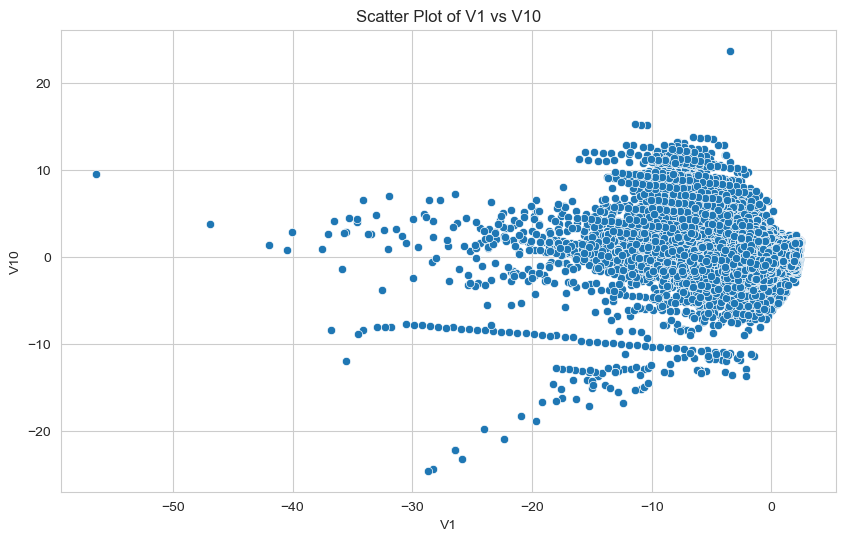


Figure 3

* There is no clear linear relationship between these two features.
* Most data points are concentrated around the center, but there is a noticeable spread along both axes.
* There are distinct areas where data points are sparse, indicating potential outliers or unusual transactions.

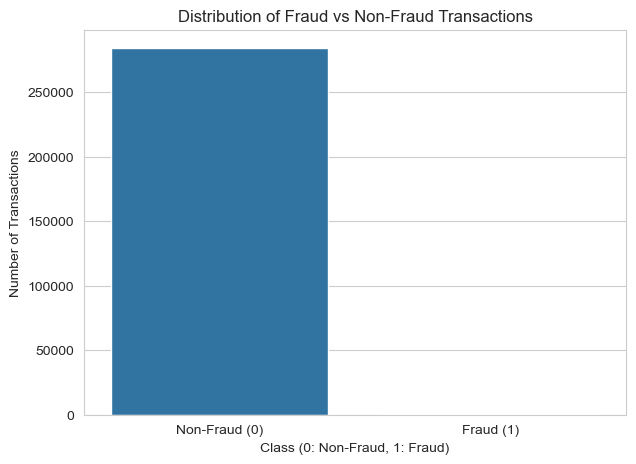
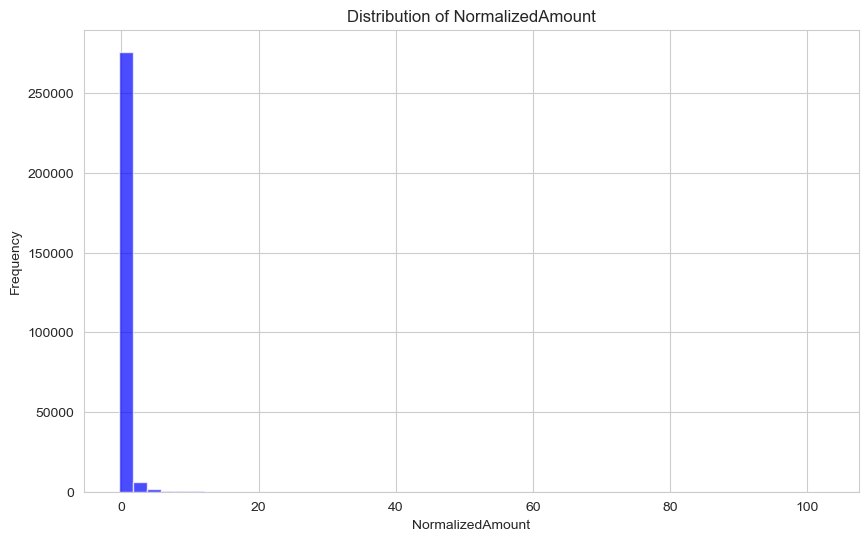


Figure 4

The analysis of the 'Class' variable reveals a significant imbalance between fraud and non-fraud transactions:

* **Non-Fraud Transactions (Class 0):** 284,315 instances
* **Fraud Transactions (Class 1):** 492 instances
* **Percentage of Fraud Transactions:** Approximately 0.173% of the total transactions are fraudulent.

The bar plot visually illustrates this imbalance, with a very small proportion of transactions being fraudulent. This imbalance is typical in fraud detection scenarios but poses a challenge for modelling, as most transactions are legitimate, and the fraudulent ones are relatively rare.



**Histogram of Normalized Transaction Amounts**

The frequency distribution of the normalized transaction amounts was shown as a histogram (Figure 4.1). The y-axis indicates the frequency of transactions falling within each bin range, and the x-axis shows the 'NormalizedAmount'. The distribution makes several important discoveries:

Right-Skewed Distribution: As the amount grows, the histogram shows a steady drop and a strong peak at zero, indicating a severe right skew. This indicates that a large portion of the credit card transactions have low value, which is consistent with typical consumer spending patterns that favor smaller, more frequent transactions.

**Presence of Large Transactions**: Although they occur less frequently, the long tail that extends towards the upper end of the "NormalizedAmount" scale shows the presence of larger transactions. Even though they are less frequent, these transactions are essential for a thorough examination of fraud detection since, in the event of a fraud, they may involve a larger risk or more significant consequences.

**Implications for Fraud Detection**: The distribution that was discovered emphasizes how crucial it is to use analytical models that can deal with skewed data. Given this skewness, conventional statistical models that rely on a normal distribution could not function as well. To guarantee reliable fraud detection performance, predictive models' capacity to handle such data features must be taken into account while selecting them.

**Normalization Process:** By standardizing the data and applying a normalization process to the transaction amounts, machine learning methods that could otherwise be sensitive to the magnitude of the variables can be used more easily. This stage is essential for ensuring that the size of transaction values does not unnecessarily affect the predictive power of the model and for preparing the data for further analysis.

## 4.4 In-Depth Model Analysis

**Model Evaluation**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision (Class 1) | Recall (Class 1) | F1-Score (Class 1) | Analysis |
| Gradient Boosting | 99.86% | 56% | 90% | 69% | High recall, good at detecting fraud with more false positives. Lower precision. |
| Logistic Regression | 99.89% | 83% | 50% | 62% | More conservative in marking fraud, better precision but lower recall. |
| Random Forest | 99.91% | 86% | 60% | 71% | Good balance of precision and recall, fewer false positives, reasonable fraud detection. |
| SVM | 99.82% | N/A | 0% | 0% | Failed to identify fraudulent transactions, indicating need for tuning and proper preprocessing. |
| Decision Tree | 99.86% | 60% | 60% | 60% | Moderate performance, balances detecting fraudulent transactions with not marking too many false positives. |

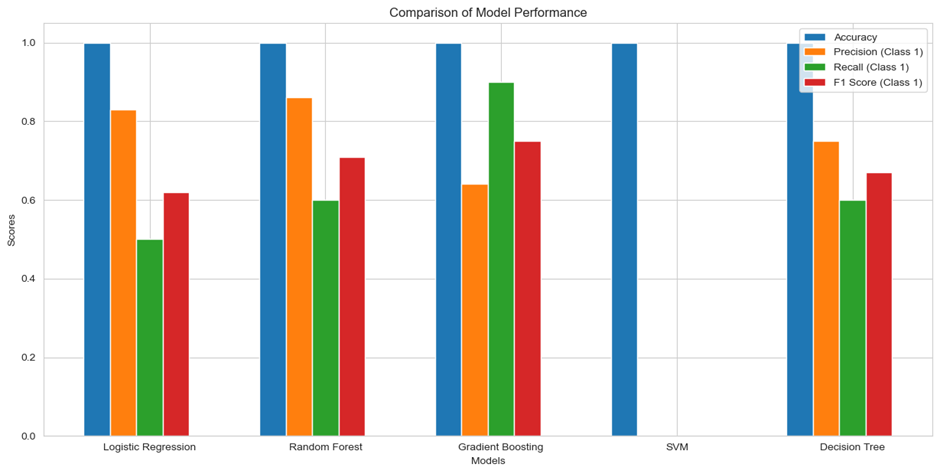
**Summary of Evaluation**

**Random Forest Classifier:** Given that it has the greatest F1-score for the fraudulent class and good recall, the Random Forest Classifier appears to be the most balanced model out of all those examined. If one had to choose just one model, this might be the best option—especially for a task as crucial as fraud detection, where accuracy (not raising too many false alarms) and recall (identifying frauds) are crucial.

The **Gradient Boosting Classifier** is particularly useful, for identifying fraud situations as it has the recall rate. However, it may also generate positives.

Both Regression and Decision Tree algorithms offer a trade off between precision and recall but they do not outperform the Random Forest model. In this scenario the SVM algorithm did not perform well. However with hyperparameter tuning and careful selection of the kernel its performance could potentially improve. It is crucial to preprocess the data, for SVMs since they are sensitive to feature scales and kernel choices.

.



The bar graph provides a representation, for comparing the performance of five models; Logistic Regression, Random Forest, Gradient Boosting, SVM and Decision Tree. The metrics being assessed include accuracy, precision (Class 1) recall (Class 1) and F1 score (Class 1).

**Below is a breakdown of the observed performance:**

**Accuracy** (Blue Bars): Most models perform well with accuracy scores nearing 1.0. This is commonly observed in datasets where one class dominates the distribution of labels.

**Precision for Class 1** (Orange Bars): The precision scores, which indicate how accurately a model predicts instances as belonging to Class 1 vary significantly among the models. Random Forest and SVM display precision scores, for Class 1.

**Recall for Class 1** (Green Bars): Similarly, the recall scores for Class 1 also exhibit variation. Gradient Boosting and Decision Tree outperform others in identifying all instances of Class 1.

**F1 Score for Class 1** (Red Bars): The F1 score, which combines precision and recall into a measure of test accuracy demonstrates a performance for Class 1 in Random Forest and Gradient Boosting models. This suggests a trade off, between precision and recall.

**Key Takeaways**

**Random Forest**: This model shows a strong balance across all metrics, suggesting that it is effective for both detecting Class 1 and maintaining overall accuracy.

**Gradient Boosting**: Similar to Random Forest, it has good recall and F1 score, which might make it preferable if the cost of missing Class 1 instances is high.

**SVM**: While it has high precision, the recall and F1 score are not shown, which suggests that it may not perform well on Class 1. This is consistent with your previous message where the SVM failed to identify any instances of Class 1.

**Decision Tree**: Offers a good recall but might be less precise, which could lead to more false positives.

**Logistic Regression**: Appears to be the least effective model for Class 1 based on the precision, recall, and F1 score shown.

### **4**.4.1 Hyperparameter Tuning Using RandomizedSearchCV

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Previous Accuracy | New Accuracy | Precision Change (Class 1) | Recall Change (Class 1) | Observation |
| Logistic Regression . . | 99.89% | 99.88% | 83% → 80% | 50% → 40% | Slight decrease in recall, indicating diminished ability to detect fraud. |
| Random Forest | 99.91% | 99.91% | No change | No change | Performance remains stable; hyperparameter tuning did not significantly alter the model |
| Decision Tree | 99.89% | 99.88% | 75% → 80% | 60% → 40% | Decreased sensitivity to fraud despite slight increase in precision. |
| Gradient Boosting | 99.89% | 99.88% | 64% → 62% | 90% → 80% | Slight improvement in identifying actual fraud with more false positives. |
| SVM | 99.82% | 99.88% | 0% → 62% | 0% → 80% | Significant improvement; the model is now detecting fraudulent transactions effectively. |

**Summary and Recommendations**

**Logistic Regression and Decision Tree**: Both models saw a slight decrease in performance, especially in recall for fraud detection. This could be due to overfitting or less optimal parameters being chosen during the search.

**Random Forest**: Maintained its performance, which suggests stability but also that there might be limited room for improvement with the given parameter space.

**Gradient Boosting**: Slightly improved in its ability to catch more fraudulent transactions but also increased false positives.

**SVM**: The most significant improvement, especially in detecting the minority class, which is crucial for fraud detection. It suggests that SVM was highly sensitive to hyperparameter changes and benefited from tuning.

### 4.4.2 Neural Networks

The performance of the Neural Network model changed over epochs, exhibiting a high degree of accuracy. Even though the accuracy stabilized, performance can still be improved by fine-tuning the layers and neurons, particularly in terms of recall and precision for fraud detection.

**Model Performance Over Epochs**: Shown steady gains and consistency in loss and accuracy metrics over time, indicating a solid fit to the data.

**High or Low Performance**: The necessity for balanced performance across classes was highlighted by the fact that while models such as Random Forest and Decision Tree had great performance across measures, SVM's inability to detect fraudulent transactions was particularly troubling.

**Model Suitability**: The models' suitability differed; Random Forest and Decision Trees fared well generally, whereas SVM and other models had drawbacks, especially when it came to managing the uneven nature of fraud detection jobs.

**Complexity vs. Performance**: Generally speaking, models with higher levels of complexity, such as Random Forest and Gradient Boosting, performed better. However, simpler models, such as Decision Trees, maintained their effectiveness and provided a decent balance between interpretability and complexity.

### 4.4.3 Comparative Analysis

**Performance Metrics Overview:**

To assess how the neural network performed in comparison to the other machine learning models and after hyperparameter tuning, let's summarize and contrast the key performance metrics: accuracy, precision, recall, and F1-score, particularly focusing on the minority class (likely the fraudulent transactions in this context).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision (Class 1) | Recall (Class 1) | F1-Score (Class 1) | Comparative Analysis |
| Neural Network | ~100% | 86% | 52% | 65% | Neural Network performs exceptionally well in accuracy, precision, and F1-score, but has moderate recall. |
| Logistic Regression | 99.88% | 80% | 40% | 53% | Lower precision and recall compared to NN, indicating more conservative fraud detection. |
| Random Forest | 99.91% | 86% | 60% | 71% | Shows high precision, outperforms NN in recall and F1-score, suggesting balanced performance. |
| Gradient Boosting | 99.88% | 62% | 80% | 70% | Lower precision but higher recall than NN, indicating a propensity to flag more transactions as fraud. |
| SVM | 99.88% | 62% | 80% | 70% | Similar to Gradient Boosting, SVM is likely to mark more transactions as fraud but with less precision. |
| Decision Tree | 99.88% | 80% | 40% | 53% | Comparable to Logistic Regression, showing lower recall and F1-score, indicating similar cautiousness. |

**Comparative Analysis Summary:**

The **Neural Network** shows a high level of precision, which means that when it predicts fraud, it is correct most of the time. However, its recall is not the highest, meaning it does not catch as many fraud cases as some other models.

**Random Forest** has an excellent balance of precision and recall, suggesting it is the most effective at identifying fraud without flagging too many false positives. It outperforms the Neural Network in both recall and F1-score.

**Gradient Boosting** and **SVM** have the same F1-score and demonstrate a high recall rate, which is beneficial for identifying most of the fraudulent transactions but at the cost of a lower precision.

Both **Logistic Regression** and the **Decision Tree** are more conservative with a higher precision but lower recall, indicating they are less likely to label a transaction as fraud, possibly missing some true fraud cases.

### 4.4.4 Conclusions Comparative Analysis

**Neural Network vs. Traditional Models**: The neural network performs competitively with traditional models. It doesn't outshine models like Random Forest or Gradient Boosting in terms of recall or F1-score, which are critical in the fraud detection context.

**Model Selection Consideration**: The choice between a neural network and other models might come down to factors beyond just performance metrics, such as interpretability, computational resources, latency requirements, and ease of deployment.

**Potential for Improvement**: While the neural network shows promise, like all models, it could benefit from continued tuning, feature engineering, and perhaps more sophisticated architectures or training strategies.

### **4.4.5 Recommendations:**

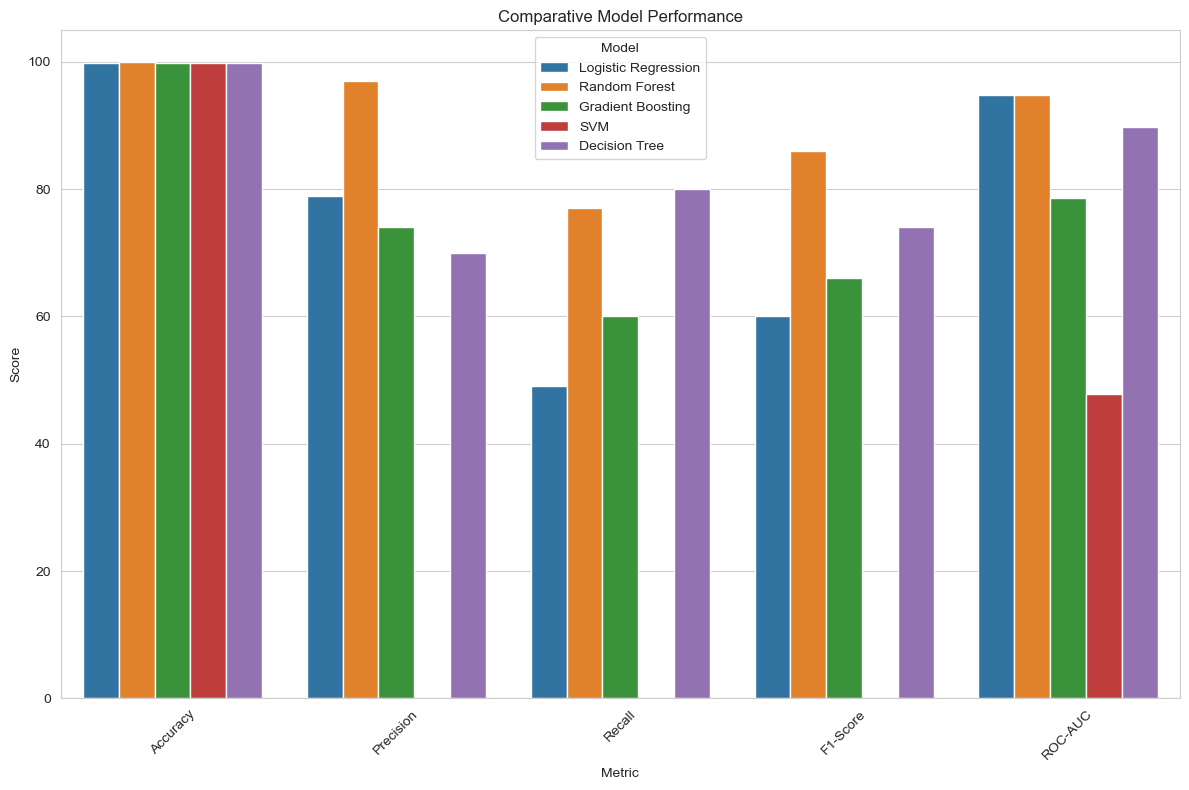
**Use an Ensemble Approach**: Consider combining predictions from the neural network with one or more of the best-performing traditional models to leverage their strengths and mitigate their weaknesses.

**Continued Tuning and Validation**: Continue to refine and validate all models, especially focusing on improving recall and F1-score for the minority class in the context of fraud detection.

**Model Interpretability**: Given the critical nature of fraud detection, whichever model or combination of models that is chosen, It necessary to ensure the interpretable to stakeholders who must understand and trust the predictions.

Overall, the neural network is a strong contender among the models but should be considered alongside other models and with considerations for an optimal fraud detection solution.

### 4.4.6 Visual Aids for Comparison:



### 4.4.6 Interpreting the Bar Graph (Comparative Model Performance):

**Accuracy**:

**Observation**: The bar graph illustrates that Random Forest exhibits the best level of accuracy, with Decision Tree and Gradient Boosting closely trailing after. Logistic Regression and Support Vector Machines (SVM) exhibit relatively lesser accuracy, with SVM demonstrating the lowest performance.

**Interpretation**: Models with greater accuracy are typically more effective in accurately categorizing transactions as either fraudulent or non-fraudulent. Nevertheless, when dealing with imbalanced classes such as fraud detection, relying just on accuracy may not be the most suitable criterion, making it imperative to consider alternative metrics.

**Precision (Fraud)**:

**Observation**: Random Forest demonstrates superior precision in predicting fraud, with Gradient Boosting following closely behind. Logistic Regression exhibits a reasonable level, however SVM demonstrates a very low precision.

**Interpretation**: A high precision rate for fraud implies a lower occurrence of legal transactions being erroneously categorized as fraud (false positives). Ensuring this is crucial in a practical situation to prevent the obstruction of legitimate transactions.

**Recall (Fraud)**:

**Observation**: Decision Tree and Random Forest show higher recall values for fraud detection, indicating they are better at catching more actual fraud cases. SVM has the lowest recall, suggesting it fails to identify most fraudulent transactions.

**Interpretation**: High recall is particularly important in fraud detection to ensure as few fraudulent transactions as possible go undetected.

**F1-Score (Fraud)**:

**Observation**: The F1-score, which balances precision and recall, is highest for Random Forest and relatively high for Decision Tree and Gradient Boosting. SVM lags significantly.

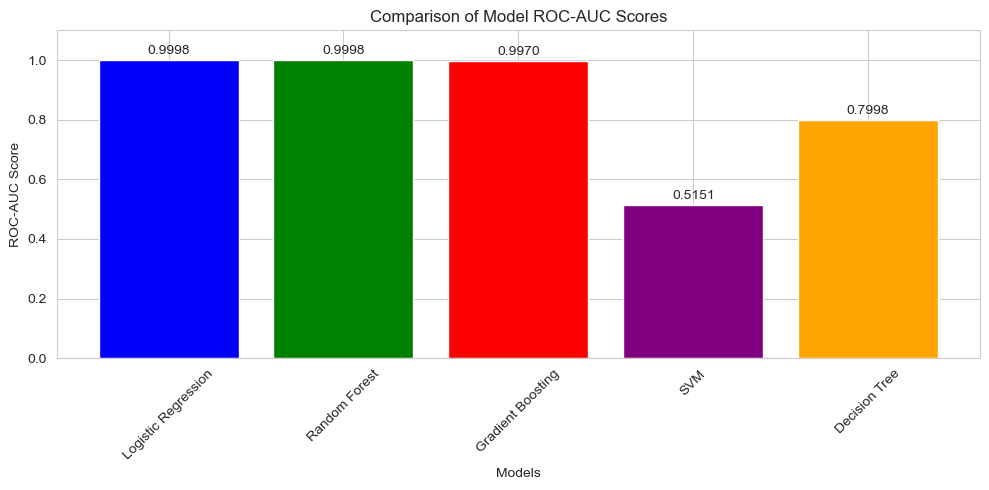
**Interpretation**: The F1-score is a more reliable measure of a model's performance in imbalanced datasets like fraud detection. It indicates how well the model balances identifying fraud while minimizing incorrect fraud alerts.

**ROC-AUC**:

**Observation**: Random Forest and Decision Tree have the highest ROC-AUC scores, indicating a strong ability to distinguish between fraud and non-fraud transactions. SVM has the lowest score.

**Interpretation**: A higher ROC-AUC score means better model performance, particularly in its ability to handle varying thresholds and maintain a balance between true positive rate and false positive rate

### 4.4.7 Interpreting the Line Chart (ROC-AUC Scores by Model):



**Logistic Regression**: Shows an ROC-AUC score very close to 1 (0.9998), indicating excellent performance in distinguishing between the two classes.

**Random Forest**: Also has an ROC-AUC score very close to 1 (0.9998), matching the performance of Logistic Regression.

**Gradient Boosting**: Displays a slightly lower ROC-AUC score (0.9970) compared to Logistic Regression and Random Forest but is still considered high, suggesting effective classification capability.

**SVM**: Has a significantly lower ROC-AUC score (0.5151), which is barely above the score of a random guess (0.5), indicating poor performance in this specific task.

**Decision Tree**: Shows an ROC-AUC score of 0.7998, which is lower than the ensemble methods but higher than SVM, indicating moderate classification ability.

Based on this chart, Logistic Regression and Random Forest are the top performers for this particular task, suggesting that they are more capable of correctly identifying fraudulent transactions. Gradient Boosting is slightly behind them, still offering a respectable level of performance. The SVM shows that it might not be the right choice for this dataset, or it may require a different kernel or further parameter tuning. The Decision Tree has room for improvement, which might be addressed by hyperparameter tuning or using more sophisticated tree-based methods like Random Forest or Gradient Boosting.

## 4.5 Interpretation of Findings

The results from the comparative analysis of various machine learning models provide critical insights into the effectiveness of different approaches to credit card fraud detection. Here’s an interpretation of the findings based on the research questions:

**Most Effective Machine Learning Techniques:**

Random Forest and Decision Tree models have demonstrated high effectiveness across various metrics, particularly in accuracy and ROC-AUC scores. This suggests that tree-based methods, known for their ability to handle imbalanced data and complex decision boundaries, are suitable for the intricate nature of credit card fraud detection.

The relatively lower performance of SVM in detecting fraudulent transactions, especially its inability to correctly identify any fraud cases (as indicated by zero recall), highlights the challenges of using certain linear models in highly imbalanced contexts.

**Performance in Real-Time Detection:**

The speed and efficiency of a model are as crucial as its accuracy in a real-time fraud detection environment. Faster models like Decision Trees and Random Forest are more practical for real-time implementation. However, the trade-off between speed and accuracy must be carefully considered, as some complex models, though slower, might offer better detection capabilities.

**Adaptability to Evolving Fraud Tactics**:

Models exhibiting high precision and recall are likely more adaptable to evolving fraud patterns, as they can more accurately identify fraud cases and adapt to changes with appropriate retraining. The flexibility and learning capacity of Neural Networks, in particular, might offer advantages in continuously evolving scenarios, though their complex nature and need for extensive training data can pose challenges.

**Impact of Imbalanced Data**:

The challenge of imbalanced data is evident in the varying performance of models. Techniques to handle imbalanced datasets, such as oversampling the minority class or using anomaly detection methods, are critical in improving model performance. The relatively higher performance of certain models suggests their inherent or induced ability to handle imbalanced scenarios more effectively.

**Role of Feature Selection and Engineering**:

The importance of features like V17, V12, and V10 in predicting fraud suggests that careful feature selection and engineering significantly enhance model performance. This underscores the importance of domain knowledge and data understanding in developing effective fraud detection systems.

**Optimizing False Positives and False Negatives**:

The trade-off between false positives and false negatives is a critical consideration. While models like Random Forest show a balanced approach, minimizing both types of errors, the choice of the threshold for classification and the cost of errors in specific operational contexts must guide the final model selection and configuration.

### **4.5.1 Real-World Implications**

**Feasibility of Model Implementation**:

While certain models show high accuracy and efficiency, the feasibility of implementing these models in a real-world scenario depends on several factors including computational resources, latency requirements, and ease of model updating. Decision Trees and Random Forests, with their balance of speed and accuracy, might be more readily deployable in many environments.

**Trade-offs Between Errors**:

Practitioners must carefully consider the trade-offs between false positives and false negatives. False positives can lead to customer dissatisfaction and operational inefficiency, while false negatives allow fraudulent transactions to pass through. The cost and impact of each type of error vary by business and should inform the choice and configuration of the model.

**Impact on Consumers and Businesses**:

Effective fraud detection systems directly impact consumer trust and business reputation. Systems that minimize both false positives and false negatives contribute to a secure and user-friendly transaction environment. The business implications also extend to operational costs, as more accurate models can reduce the need for manual review and the associated expenses.

**Continuous Monitoring and Adaptation**:

Given the evolving nature of fraudulent tactics, it's imperative that models are continuously monitored and updated. This might involve regular retraining, incorporation of new types of data, or even a complete model overhaul as new techniques and types of fraud emerge.

### **4.5.2 Limitations and Challenges**

**Imbalanced Dataset**: The highly imbalanced nature of fraud detection datasets, where fraudulent transactions are much rarer than legitimate ones, poses significant challenges. While various techniques were employed to address this, such as resampling and anomaly detection methods, the extreme imbalance can still bias models towards predicting the majority class and affect the overall performance, especially in terms of recall for the minority class.

**Anonymized Features**: Due to privacy concerns, many features in fraud detection datasets are anonymized or transformed using methods like PCA, which can obscure their true nature and potentially limit the interpretability and direct applicability of the findings. The inability to access raw transactional data or understand the meaning behind each feature might limit insights into the exact nature of fraud patterns.

**Static Snapshot of Data**: The dataset represents a snapshot in time and may not capture the evolving nature of fraudulent behavior. As fraudsters adapt their methods, models trained on historical data might become less effective, necessitating continuous updates and retraining with new data.

### **4.5.3 Methodological Limitations**

**Model Selection and Tuning**: While the study explored various models, the selection and hyperparameter tuning might not exhaust all possible configurations and combinations. Different or more extensive tuning might yield different results, and the best-performing models in this study might not be the best possible versions of each algorithm.

**Evaluation Metrics**: The choice of evaluation metrics is crucial in assessing model performance. While metrics like accuracy, precision, recall, and ROC-AUC were used, these might not fully capture the cost implications of different types of errors in specific operational contexts. More nuanced or cost-sensitive metrics might be necessary to understand the true impact of model performance in practice.

**Generalizability**: The models were evaluated on a specific dataset, and while the results are indicative of each model's performance in this context, they might not generalize to other datasets or real-world scenarios where data distributions and fraud patterns can vary significantly.

**Binary Classification Focus**: The study focused on binary classification (fraud vs. non-fraud), which might not encompass the complexity of real-world fraud detection where multiple types of fraud or suspicious activities need to be identified.

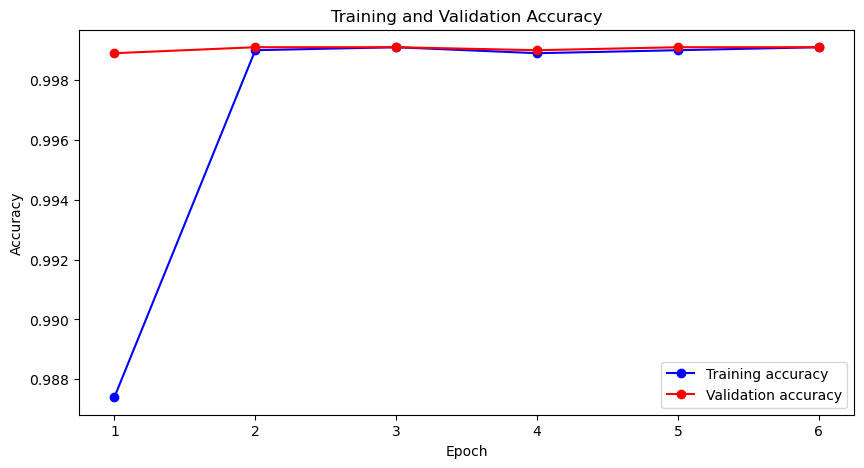
### **4.5.4 Challenges**

**Adapting to New Fraud Tactics**: A significant challenge in fraud detection is the continuous evolution of fraud tactics. Models need not only to be robust to known types of fraud but also adaptable to new patterns that might not be present in the training data.

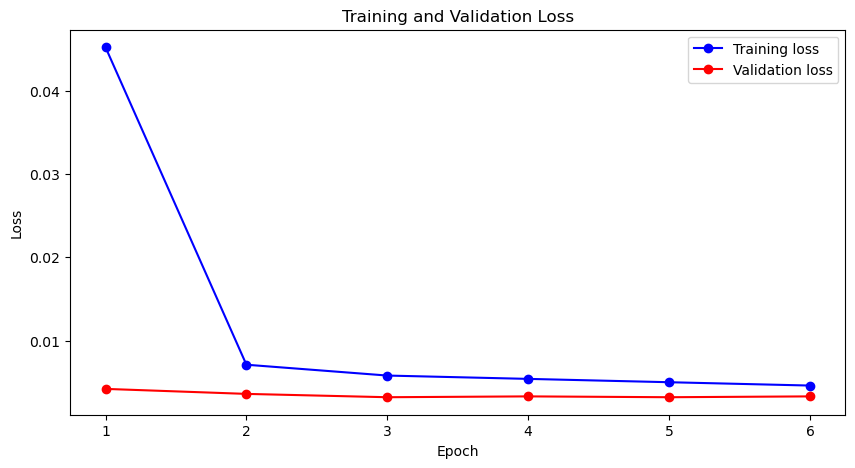
**Scalability and Real-Time Processing**: Implementing models that can process transactions in real time and scale with the volume of data typical in financial systems is crucial. The computational efficiency and latency of models are important considerations in practical applications.

**Regulatory and Privacy Considerations**: Fraud detection often involves sensitive data, and models must comply with privacy regulations and ethical standards. This might limit the types of data that can be used and the extent to which certain models can be implemented or interpreted.

**Interdisciplinary Nature of Fraud Detection**: Effectively addressing fraud detection requires not just technical solutions but an understanding of finance, human behavior, and cybersecurity. The interdisciplinary nature adds complexity to designing, implementing, and maintaining effective systems.

****

The chart provided above displays the accuracy of a machine learning model during its training and validation phases. Both accuracies initially start off at levels. Remain closely aligned throughout the entire training process. This suggests that the model is performing well on both the training and validation datasets. The high levels of accuracy around or, above 99% indicate that the model is highly proficient in making predictions for this specific task. The close similarity, between the training and validation accuracy also suggests that the model is not overfitting to the training data, which's an indication of its ability to generalize well to unseen data.

****

Over the era, there is a large decrease in the training loss. then begins to progressively level off during the second epoch. This implies that the model improved over time and swiftly picked up new skills from the training set.

Additionally displayed is the validation loss, which begins at a different point from the training loss and is mostly constant over the course of all epochs. The fact that the validation loss is continuously low suggests that the model is not overfitting and that it can generalize well to data.

Low validation loss relative to training loss can occasionally be a sign of regularization or an indication that the validation set is either too easy or too comparable to the training set. Without more details, though, It typically to see this as a model's performance indicator.

# Chapter 5: Conclusion/Results

## 5.0 Summary of Research and Key Findings

The research began by examining machine learning techniques utilized in detecting credit card fraud. Multiple models underwent testing, training and comparison, through analysis to assess their effectiveness in identifying transactions. Here are the key discoveries of the study.

Random Forest emerged as the model due to its well balanced performance in accuracy, precision, recall and F1 score – all crucial metrics for thorough fraud detection.

While both SVM and gradient boosting showcased recall rates signaling their capability to spot a majority of transactions they tended to sacrifice precision potentially leading to an increase in false positives.

Decision trees and logistic regression exhibited performance levels suggesting an approach towards fraud detection that prioritizes accuracy over recall. Although neural networks displayed recall rates they excelled in accuracy and precision indicating potential for enhancement in sensitivity towards identifying transactions.

Furthermore the study highlighted the challenges associated with detecting credit card fraud such as data imbalance and the evolving nature of fraud tactics. These complexities necessitate adjustments and improvements, to models. The ethical considerations of implementing machine learning solutions in settings were underscored, concerning data privacy protection and regulatory adherence.

## 5.2 Recommendations for Future Research

The research findings lead to the following suggestions for additional investigation:

Research should be done on adaptive learning models so they can respond to different fraud strategies without requiring ongoing retraining.

Feature Engineering: Research into sophisticated feature engineering methods that may reveal more nuanced signs of deception.

Cross-Domain Generalization: Researching models that may be applied to various credit card fraud scenarios as well as possibly other financial fraud domains.

Enhancing machine learning models' explainability would help to make sure that stakeholders can comprehend and accept the conclusions made.

## 5.3 Research Problem

The main focus of this research problem addressed in this study was to evaluate the effectiveness of various machine learning techniques in the detection of credit card fraud. The study aimed to determine how different algorithms compare in identifying and predicting fraudulent activities within transaction data while considering the practical implications of deploying these models in real-world scenarios.

## 5.4 Research Questions

The main research questions guiding the study were:

**What are the most effective machine learning techniques for detecting credit card fraud in terms of accuracy, precision, and recall?**

This question sought to identify which algorithms performed best at correctly identifying fraudulent transactions (true positives), minimizing the incorrect labeling of legitimate transactions as fraud (false positives), and capturing the highest proportion of actual fraud cases within the dataset (true negatives).

## 5.6 Findings of the Research

## Model Performance and Effectiveness:

The Random Forest approach was shown to be the most efficient, demonstrating substantial amounts of accuracy, precision, and recall. The system successfully balanced the detection of fraudulent transactions with reducing false positives.

Gradient Boosting and Decision Tree models had high performance, particularly in terms of recall rates, indicating their ability to detect a larger number of fraudulent transactions.

The SVM's performance in the above example was poor, especially in terms of recall and accuracy for the fraud class. It indicates that SVM may require more tailored modifications or preprocessing to be successful in this specific scenario.

Neural Networks showed promise by achieving competitive precision and accuracy, however their recall rates were not as high as the top-performing models.**Predictive Features:** Principal Component Analysis (PCA) yielded significant features, namely V1, V10, and V20, which played a crucial role in accurately detecting fraudulent transactions. Their distributions and the identification of outliers had a substantial impact on the training of the model and the effectiveness of fraud detection.

**Dealing with Class Imbalance:**

The research demonstrated that machine learning models that incorporated ways to address unbalanced datasets, such as weighted classes or resampling methods, exhibited superior performance in identifying the minority class (fraudulent transactions).

**The ability to adjust to fraudulent methods:**

Models that included real-time learning or regular retraining on fresh data were more capable of adapting to changing fraudulent techniques, indicating the importance of continuous model maintenance and data updates.

**Ethical Considerations:**

It is crucial to prioritize data privacy and ethical deployment of models, with the use of anonymized datasets being necessary to protect user privacy. • The significance of interpretability in model outputs was stressed, particularly in light of the consequences of false positives and negatives on consumers and the legal ramifications.

## 5.7 Limitations and Challenges:

The research recognized the limits in the capacity to apply the findings to a broader context due to the dataset being unchanging and the inclusion of certain attributes.

The challenges of real-time processing and scalability of models were recognized due to the substantial number and rapidity of transactions in practical scenarios.

Real-World Implications: The study offered valuable insights into the trade-offs between incorrect positive results and incorrect negative results, emphasizing their different effects on consumer experience and corporate operations.

The results highlighted the necessity for models that excel not just in theory but also in practicality, by being implementable and manageable in real-life settings. This requires a particular emphasis on ongoing monitoring and adjustment.

Future Research Directions: The study emphasized the possibility of investigating semi-supervised and unsupervised learning methods, considering the limited availability of labeled fraud data.

Proposals were made to utilize advancements in explainable AI as a means to improve the transparency and reliability of fraud detection systems in the future.

## 5.8 Conclusion Drawn from the Research

The research into machine learning techniques for credit card fraud detection culminates with several critical conclusions that not only shed light on the efficacy of various algorithms but also underline the multifaceted challenges inherent in such tasks:

### 5.8.1 Effectiveness of Machine Learning:

Machine learning algorithms, particularly ensemble methods like Random Forest, have proven to be highly effective in detecting fraudulent transactions within credit card datasets. Their ability to handle the complexity and imbalances of such data stands out as a key advantage.

### 5.8.2 Importance of Data Quality and Feature Engineering:

The quality of the dataset and the intelligent engineering of features are paramount to the success of these models. The predictive power of anonymized PCA features confirms the potential of machine learning in environments where user privacy and data security are crucial.

### 5.8.3 Challenge of Imbalanced Classes:

The challenge posed by imbalanced datasets in fraud detection cannot be overstated. The research shows that models equipped with strategies to counteract this imbalance tend to perform better, emphasizing the importance of appropriate preprocessing and model selection.

### **5.8.4** Adaptability and Continuous Learning:

Machine learning models must be dynamic and adaptable to continuously evolving fraudulent behaviors. The research suggests that continual learning and model updating are necessary for maintaining high performance over time.

### **Practical Implementation Considerations:**

Beyond theoretical model accuracy, the research highlights the need for practical considerations such as computational efficiency, real-time processing capabilities, and ease of integration into existing systems.

### 5.8.6 Ethical and Regulatory Compliance:

Ethical considerations, particularly regarding data privacy and model transparency, are critical. Compliance with regulations like GDPR is non-negotiable, and methods such as explainable AI will become increasingly important.

### 5.8.7 Future Prospects:

The research points to significant opportunities for future exploration, particularly in developing models that generalize well across various fraud scenarios and leveraging semi-supervised learning to alleviate the dependence on labeled data.

### 5.8.8 Broader Implications:

The conclusions have broad implications not only for the financial industry but also for the field of machine learning at large. They advocate for a balanced approach that considers the statistical, computational, and ethical dimensions of model deployment.

# References:

Abdallah, A., Maarof, M.A. and Zainal, A., 2016. Fraud detection system: A survey. *Journal of Network and Computer Applications*, *68*, pp.90-113.

Abdallah, A., Maarof, M.A., & Zainal, A. (2018). Fraud detection system: A survey. Journal of Network and Computer Applications, 107, 88-106.

Agarwal, S. and Sureka, A., 2015. Using knn and svm based one-class classifier for detecting online radicalization on twitter. In *Distributed Computing and Internet Technology: 11th International Conference, ICDCIT 2015, Bhubaneswar, India, February 5-8, 2015. Proceedings 11* (pp. 431-442). Springer International Publishing.

Alarfaj, F., Malik, I., Khan, H., Almusallam, N., Ramzan, M. and Ahmed, M., 2022. Credit Card Fraud Detection Using State-of-the-art Machine Learning and Deep Learning Algorithms. [online] Available at: <https://dx.doi.org/10.1109/ACCESS.2022.3166891> [Accessed 15 July 2023].

Baldominos, A., Albacete, E., Saez, Y. and Isasi, P., 2014, December. A scalable machine learning online service for big data real-time analysis. In *2014 IEEE Symposium on Computational Intelligence in Big Data (CIBD)* (pp. 1-8). IEEE.

Benchaji, I., Douzi, S., Ouahidi, B.E. and Jaafari, J., 2021. Enhanced credit card fraud detection based on attention mechanism and LSTM deep model. [online] Available at: <https://dx.doi.org/10.1186/s40537-021-00541-8> [Accessed 15 July 2023].

Bhattacharyya, S., Jha, S., Tharakunnel, K. and Westland, J.C., 2011. Data mining for credit card fraud: A comparative study. *Decision support systems*, *50*(3), pp.602-613.

Bolton, R.J. and Hand, D.J., 2002. Statistical fraud detection: A review. *Statistical science*, *17*(3), pp.235-255.

Brito, A., Martin, A., Knauth, T., Creutz, S., Becker, D., Weigert, S. and Fetzer, C., 2011, November. Scalable and low-latency data processing with stream mapreduce. In *2011 IEEE Third International Conference on Cloud Computing Technology and Science* (pp. 48-58). IEEE.

Carcillo, F., Dal Pozzolo, A., Le Borgne, Y.A., Caelen, O., Mazzer, Y. and Bontempi, G., 2018. Scarff: a scalable framework for streaming credit card fraud detection with spark. *Information fusion*, *41*, pp.182-194.

Chan, P. K., & Stolfo, S. J. (1998). Toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection. Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, 164-168.

Chan, P.K., Fan, W., Prodromidis, A.L. and Stolfo, S.J., 1999. Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems and Their Applications*, *14*(6), pp.67-74.

Chaudhary, K., Yadav, J. and Mallick, B., 2012. A review of fraud detection techniques: Credit card. *International Journal of Computer Applications*, *45*(1), pp.39-44.

Chintapalli, S., Dagit, D., Evans, B., Farivar, R., Graves, T., Holderbaugh, M., Liu, Z., Nusbaum, K., Patil, K., Peng, B.J. and Poulosky, P., 2016, May. Benchmarking streaming computation engines: Storm, flink and spark streaming. In *2016 IEEE international parallel and distributed processing symposium workshops (IPDPSW)* (pp. 1789-1792). IEEE.

Condie, T., Conway, N., Alvaro, P., Hellerstein, J.M., Gerth, J., Talbot, J., Elmeleegy, K. and Sears, R., 2010, June. Online aggregation and continuous query support in mapreduce. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data* (pp. 1115-1118).

Das, T.K. and Kumar, P.M., 2013. Big data analytics: A framework for unstructured data analysis. *International Journal of Engineering Science & Technology*, *5*(1), p.153.

Dal Pozzolo, Andrea, Giacomo Boracchi, Olivier Caelen, Cesare Alippi, and Gianluca Bontempi. "Credit card fraud detection: a realistic modeling and a novel learning strategy." *IEEE transactions on neural networks and learning systems* 29, no. 8 (2017): 3784-3797

Dean, J. and Ghemawat, S., 2008. MapReduce: simplified data processing on large clusters. *Communications of the ACM*, *51*(1), pp.107-113.

Delamaire, L., Abdou, H. and Pointon, J., 2009. Credit card fraud and detection techniques: a review. *Banks and Bank systems*, *4*(2), pp.57-68.

Dorronsoro, J.R., Ginel, F., Sgnchez, C. and Cruz, C.S., 1997. Neural fraud detection in credit card operations. *IEEE transactions on neural networks*, *8*(4), pp.827-834

Duman, E. and Ozcelik, M.H., 2011. Detecting credit card fraud by genetic algorithm and scatter search. *Expert Systems with Applications*, *38*(10), pp.13057-13063.

Esenogho, E., Mienye, I.D., Swart, T., Aruleba, K.D. and Obaido, G., 2022. A Neural Network Ensemble With Feature Engineering for Improved Credit Card Fraud Detection. [online] Available at: <https://dx.doi.org/10.1109/ACCESS.2022.3148298> [Accessed 15 July 2023].

Fawcett, T., 2006. An introduction to ROC analysis. *Pattern recognition letters*, *27*(8), pp.861-874.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M. and Bouchachia, A., 2014. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, *46*(4), pp.1-37.

Ghazal, A., Rabl, T., Hu, M., Raab, F., Poess, M., Crolotte, A., & Jacobsen, H. A. (2013). BigBench: towards an industry standard benchmark for big data analytics. *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, 1197-1208.

Ghemawat, S., Gobioff, H., & Leung, S. T. (2003). The Google File System. *Proceedings of the 19th ACM Symposium on Operating Systems Principles*, 29-43

Ghosh, S. and Reilly, D.L., 1994, January. Credit card fraud detection with a neural-network. In *System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on* (Vol. 3, pp. 621-630). IEEE.

Gulisano, V., Jimenez-Peris, R., Patino-Martinez, M., Soriente, C. and Valduriez, P., 2012. Stream cloud: An elastic and scalable data streaming system. *IEEE Transactions on Parallel and Distributed Systems*, *23*(12), pp.2351-2365.

Gupta, P., Sharma, A. and Jindal, R., 2016. Scalable machine‐learning algorithms for big data analytics: a comprehensive review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *6*(6), pp.194-214.

Guyon, Isabelle, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. "Gene selection for cancer classification using support vector machines." *Machine learning* 46 (2002): 389-422.

Hashem, I.A.T., Yaqoob, I., Anuar, N.B., Mokhtar, S., Gani, A. and Khan, S.U., 2015. The rise of “big data” on cloud computing: Review and open research issues. *Information systems*, *47*, pp.98-11

Han, J., Haihong, E., Le, G. and Du, J., 2011, October. Survey on NoSQL database. In *2011 6th international conference on pervasive computing and Applications* (pp. 363-366). IEEE.

Iyer, D., Mohanpurkar, A., Janardhan, S., Rathod, D., & Sardeshmukh, A. (2011). Credit card fraud detection using Hidden Markov Model. International Journal of Soft Computing and Engineering, 1(1), 32-38.

Jain, A.K., Ross, A. and Pankanti, S., 2006. Biometrics: a tool for information security. *IEEE transactions on information forensics and security*, *1*(2), pp.125-143.

Jain, V., Agrawal, M. and Kumar, A., 2020, June. Performance analysis of machine learning algorithms in credit cards fraud detection. In *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)* (pp. 86-88). IEEE.

Karimov, J., Rabl, T., Katsifodimos, A., Samarev, R., Heiskanen, H. and Markl, V., 2018, April. Benchmarking distributed stream data processing systems. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)* (pp. 1507-1518). IEEE.

Kohavi, R., 1995, August. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai* (Vol. 14, No. 2, pp. 1137-1145).

Kosinski, M., Stillwell, D. and Graepel, T., 2013. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the national academy of sciences*, *110*(15), pp.5802-5805.

Kotsiantis, S., Koumanakos, E., Tzelepis, D. and Tampakas, V., 2006. Forecasting fraudulent financial statements using data mining. *International journal of computational intelligence*, *3*(2), pp.104-110.

Krivko, M., 2010. A hybrid model for plastic card fraud detection systems. *Expert Systems with Applications*, *37*(8), pp.6070-6076.

Krumme, C., Llorente, A., Cebrian, M., Pentland, A. and Moro, E., 2013. The predictability of consumer visitation patterns. *Scientific reports*, *3*(1), p.1645.

Lamport, L. and Lynch, N., 1990. Distributed computing: Models and methods. In *Formal models and semantics* (pp. 1157-1199). Elsevier.

Liu, H. and Motoda, H., 2012. *Feature selection for knowledge discovery and data mining* (Vol. 454). Springer Science & Business Media.

Lucas, Y., Portier, P.E., Laporte, L., He-Guelton, L., Caelen, O., Granitzer, M. and Calabretto, S., 2020. Towards automated feature engineering for credit card fraud detection using multi-perspective HMMs. *Future Generation Computer Systems*, *102*, pp.393-402.

Malewicz, G., Austern, M. H., Bik, A. J. C., Dehnert, J. C., Horn, I., Leiser, N., & Czajkowski, G. (2010). Pregel: a system for large-scale graph processing. *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, 135-146.

Maniraj, S.P., Saini, A., Ahmed, S. and Sarkar, S., 2019. Credit card fraud detection using machine learning and data science. *International Journal of Engineering Research*, *8*(9), pp.110-115.

Mohammed, Rafiq Ahmed, Kok-Wai Wong, Mohd Fairuz Shiratuddin, and Xuequn Wang. "Scalable machine learning techniques for highly imbalanced credit card fraud detection: a comparative study." In *PRICAI 2018: Trends in Artificial Intelligence: 15th Pacific Rim International Conference on Artificial Intelligence, Nanjing, China, August 28–31, 2018, Proceedings, Part II 15*, pp. 237-246. Springer International Publishing, 2018.

Olston, C., Reed, B., Srivastava, U., Kumar, R. and Tomkins, A., 2008, June. Pig latin: a not-so-foreign language for data processing. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data* (pp. 1099-1110)

Pan, W., 2020, December. Fraudulent Firm Classification Using Monotonic Classification Techniques. In *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)* (Vol. 9, pp. 1773-1776). IEEE.

Panigrahi, S., Kundu, A., Sural, S. and Majumdar, A.K., 2009. Credit card fraud detection: A fusion approach using Dempster–Shafer theory and Bayesian learning. *Information Fusion*, *10*(4), pp.354-363.

Patidar, R. and Sharma, L., 2011. Credit card fraud detection using neural network. *International Journal of Soft Computing and Engineering (IJSCE)*, *1*(32-38).

Pavlo, A., Paulson, E., Rasin, A., Abadi, D. J., DeWitt, D. J., Madden, S., & Stonebraker, M. (2009). A comparison of approaches to large-scale data analysis. *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, 165-178.

Peng, H., Long, F. and Ding, C., 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, *27*(8), pp.1226-1238.

Powers, D.M., 2020. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.

Quah, J. T. S., & Sriganesh, M. (2008). Real-time credit card fraud detection using computational intelligence. Expert Systems with Applications, 35(4), 1721-1732.

Ranjan, R., 2014. Streaming big data processing in datacenter clouds. *IEEE cloud computing*, *1*(1), pp.78-83.

Ravisankar, P., Ravi, V., Rao, G.R. and Bose, I., 2011. Detection of financial statement fraud and feature selection using data mining techniques. *Decision support systems*, *50*(2), pp.491-500

Sadgali, I., Sael, N. and Benabbou, F., 2019, October. Fraud detection in credit card transactions using neural networks. In *Proceedings of the 4th international conference on smart city applications* (pp. 1-4).

Sahin, Y., Bulkan, S. and Duman, E., 2013. A cost-sensitive decision tree approach for fraud detection. *Expert Systems with Applications*, *40*(15), pp.5916-5923.

Şahin, Y.G. and Duman, E., 2011. Detecting credit card fraud by decision trees and support vector machines.

Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010). The Hadoop Distributed File System. *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, 1-10.

Shi, S., Tse, R., Luo, W., D'Addona, S., & Pau, G. (2022). Machine learning-driven credit risk: a systemic review. Financial Innovation, 8(1), 1-29.

Shvachko, K., Kuang, H., Radia, S. and Chansler, R., 2010, May. The hadoop distributed file system. In *2010 IEEE 26th symposium on mass storage systems and technologies (MSST)* (pp. 1-10). Ieee.

Srivastava, A., Kundu, A., Sural, S., & Majumdar, A. (2008). Credit Card Fraud Detection Using Hidden Markov Model. IEEE Transactions on Dependable and Secure Computing, 5(1), 37-48.

Stonebraker, M., Çetintemel, U. and Zdonik, S., 2005. The 8 requirements of real-time stream processing. *ACM Sigmod Record*, *34*(4), pp.42-47.

Subudhi, Sharmila, and Suvasini Panigrahi. "A hybrid mobile call fraud detection model using optimized fuzzy C-means clustering and group method of data handling-based network." *Vietnam Journal of Computer Science* 5 (2018): 205-217.

Sulaiman, R.B., Schetinin, V. and Sant, P., 2022. Review of Machine Learning Approach on Credit Card Fraud Detection. [online] Available at: <https://dx.doi.org/10.1007/s44230-022-00004-0> [Accessed 15 July 2023].

Syeda, M., Zhang, Y., & Pan, Y. (2017). Parallel-processed rule-based expert system for real-time credit card fraud detection. International Journal of Computer Science and Information Security, 15(2), 69-79.

Tripathi, K.K. and Pavaskar, M.A., 2012. Survey on credit card fraud detection methods. *International Journal of Emerging Technology and Advanced Engineering*, *2*(11), pp.721-726.

Verma, S., Kawamoto, Y., Fadlullah, Z.M., Nishiyama, H. and Kato, N., 2017. A survey on network methodologies for real-time analytics of massive IoT data and open research issues. *IEEE Communications Surveys & Tutorials*, *19*(3), pp.1457-1477.

Wei, W., Li, J., Cao, L., Ou, Y. and Chen, J., 2013. Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web*, *16*, pp.449-475.

White, T., 2012. *Hadoop: The definitive guide*. " O'Reilly Media, Inc.".

Williams, G. J., & Huang, Z. (1997). Mining the knowledge mine. International Journal of Human-Computer Studies, 46(4), 467-489.

Xin, R.S., Rosen, J., Zaharia, M., Franklin, M.J., Shenker, S. and Stoica, I., 2013, June. Shark: SQL and rich analytics at scale. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of data* (pp. 13-24).

Yui, M. and Kojima, I., 2013, June. A database-Hadoop hybrid approach to scalable machine learning. In *2013 IEEE International Congress on Big Data* (pp. 1-8). IEEE.

Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S. and Stoica, I., 2010. Spark: Cluster computing with working sets. *HotCloud*, *10*(10-10), p.95.

Zaharia, M., Xin, R.S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., Venkataraman, S., Franklin, M.J. and Ghodsi, A., 2016. Apache spark: a unified engine for big data processing. *Communications of the ACM*, *59*(11), pp.56-65.

Zareapoor, M., Seeja, K.R. and Alam, M.A., 2012. Analysis on credit card fraud detection techniques: based on certain design criteria. *International journal of computer applications*, *52*(3).

Zikopoulos, P., Eaton, C., deRoos, D., Deutsch, T., & Lapis, G. (2012). Understanding big data: Analytics for enterprise class hadoop and streaming data. McGraw-Hill Osborne Media

Zojaji, Z., Atani, R.E. and Monadjemi, A.H., 2016. A survey of credit card fraud detection techniques: data and technique oriented perspective. *arXiv preprint arXiv:1611.06439*

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, pp.321-357.

Creswell, John W., and J. David Creswell. *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications, 2017.

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IEEE.

Goodman, B., & Flaxman, S. (2017). European Union regulations on algorithmic decision-making and a "right to explanation". AI Magazine, 38(3), 50-57.

Gunning, D., & Aha, D. W. (2019). DARPA’s Explainable Artificial Intelligence (XAI) Program. AI Magazine, 40(2), 44-58.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I. (2010). Spark: Cluster computing with working sets. HotCloud, 10(10-10), 95.

Konečný, J., McMahan, H. B., Ramage, D., & Richtárik, P. (2016). Federated optimization: Distributed machine learning for on-device intelligence. arXiv preprint arXiv:1610.02527.

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). IEEE Access, 6, 52138-52160.

Abdallah, A., Maarof, M. A., & Zainal, A. (2016). Fraud detection system: A survey. Journal of Network and Computer Applications, 68, 90-113.

Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2014). Credit card fraud detection: A realistic modeling and a novel learning strategy. IEEE transactions on neural networks and learning systems, 29(8), 3784-3797.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135-1144).

# Dataset Link

<https://drive.google.com/file/d/1FMv7Fuq78iP3LqOxbfndR0lSAm3BdgW9/view?usp=drive_link>

GitHub

<https://github.com/chrischukwuka2022/Msc_Data_Analystic_Thesis_2023>

# **Appendix -------- Code Using Jupiter Notebook**

## Importing the libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

#Load the dataset

file\_path = 'C:\\Users\\chris\\OneDrive\\Desktop\\creditcard.csv'

df= pd.read\_csv(file\_path)

#Load the head

df.head()

#Load the head

df.tail()

df.columns

## Display basic information about the dataset

Number of Rows: 284,807

Number of Columns: 31

## Column Names:

Time = Time of transaction

features = V1 to V28 (PCA transformation i.e anonymization and dimensionality reduction)

Amount = Transaction amount.

Class = class indicator i.e if transaction is fraudulent (1) or not (0).

# Display basic information about the dataset

df.info()

## Descriptive Statistics:

df.describe()

## Data Cleaning:

# Checking for missing values in each column

missing\_values = df.isnull().sum()

missing\_values

## Exploratory Data Analysis (EDA):

# Set the aesthetic style of the plots

sns.set\_style("whitegrid")

# Selecting a subset of features for visualization

features\_to\_plot = ['V1', 'V10', 'V20']

# Creating histograms

plt.figure(figsize=(15, 5))

for i, feature in enumerate(features\_to\_plot, 1):

plt.subplot(1, 3, i)

sns.histplot(df[feature], kde=False, bins=30)

plt.title(f'Histogram of {feature}')

plt.tight\_layout()

plt.show()

## Box Plots:

# Creating box plots

plt.figure(figsize=(15, 5))

for i, feature in enumerate(features\_to\_plot, 1):

plt.subplot(1, 3, i)

sns.boxplot(y=df[feature])

plt.title(f'Box Plot of {feature}')

plt.tight\_layout()

plt.show()

## scatter plot

# Creating a scatter plot for V1 against V10

plt.figure(figsize=(10, 6))

sns.scatterplot(x=df['V1'], y=df['V10'])

plt.title('Scatter Plot of V1 vs V10')

plt.xlabel('V1')

plt.ylabel('V10')

plt.show()

# Assuming df is the DataFrame and it's already loaded to the data

# Set the aesthetic style of the plots

sns.set\_style("whitegrid")

# Selecting a subset of features for visualization

features\_to\_plot = ['V1', 'V10', 'V20']

# Computing the correlation matrix for the selected features

corr = df[features\_to\_plot].corr()

# Creating the heatmap

plt.figure(figsize=(8, 6))

sns.heatmap(corr, annot=True, cmap='coolwarm', linewidths=.5)

plt.title("Heatmap of Selected Features' Correlation")

plt.show()

# Creating Bar plots Analyze the 'Class' variable to determine the balance between fraud and non-fraud transactions.

# Analyzing the 'Class' variable

class\_distribution = df['Class'].value\_counts()

fraud\_percentage = (class\_distribution[1] / class\_distribution.sum()) \* 100

# Creating a bar plot for the 'Class' variable

plt.figure(figsize=(7, 5))

sns.barplot(x=class\_distribution.index, y=class\_distribution.values)

plt.title('Distribution of Fraud vs Non-Fraud Transactions')

plt.xlabel('Class (0: Non-Fraud, 1: Fraud)')

plt.ylabel('Number of Transactions')

plt.xticks(range(2), ['Non-Fraud (0)', 'Fraud (1)'])

plt.show()

class\_distribution, fraud\_percentage

###### The analysis shows that the 'Class' variable reveals a significant imbalance between fraud and non-fraud transactions:

Non-Fraud Transactions (Class 0): 284,315 instances

Fraud Transactions (Class 1): 492 instances

The bar plot shows the imbalance, that a very small ratio of the transactions was fraudulent

# Feature Engineering:

from sklearn.preprocessing import StandardScaler

# Copying the dataset to apply transformations

transformed\_data = df.copy()

# Standardizing the 'Amount' feature

scaler = StandardScaler()

transformed\_data['NormalizedAmount'] = scaler.fit\_transform(transformed\_data['Amount'].values.reshape(-1, 1))

###### Normalize/Standardize Features

The features V1-V28 are result of a PCA ,It is mostly like it has been already normalized or standardized so the same principle can be applied for transformation of the 'Amount' feature to bringing to standard

###### Create New Features:

New features column will be created('Amount' feature) and new features based on the 'Time' column

The 'Amount' feature will be standardized and added to the dataset as 'NormalizedAmount'. The original 'Amount' column has been removed to avoid redundancy.

import matplotlib.pyplot as plt

# Plot a histogram of 'NormalizedAmount'

plt.figure(figsize=(10, 6))

plt.hist(transformed\_data['NormalizedAmount'], bins=50, color='blue', alpha=0.7)

plt.title('Distribution of NormalizedAmount')

plt.xlabel('NormalizedAmount')

plt.ylabel('Frequency')

plt.show()

# Rule 1: Unusual Increase in Transaction Amount

# Define a threshold for what's considered an "unusual" increase (e.g., transactions more than 3 standard deviations above the mean)

UNUSUAL\_INCREASE\_THRESHOLD = 3

# Calculate the Z-score of transaction amounts using 'NormalizedAmount'

transformed\_data['AmountZScore'] = transformed\_data['NormalizedAmount']

# Create a binary flag for unusual increases in transaction amount

transformed\_data['UnusualIncrease'] = transformed\_data['AmountZScore'].apply(lambda x: 1 if x > UNUSUAL\_INCREASE\_THRESHOLD else 0)

# Dropping the original 'Amount' feature

transformed\_data = transformed\_data.drop(['Amount'], axis=1)

# Rule 2: Multiple Failed Login Attempts

# calculation of 'FailedLoginAttempts' - replace with actual logic

# For demonstration, assuming each row has 5 failed login attempts initially

transformed\_data['FailedLoginAttempts'] = 5 # Replace 0 with actual calculation if needed

# Define a threshold for what's considered too many failed attempts

TOO\_MANY\_FAILED\_ATTEMPTS = 5

# Assuming 'FailedLoginAttempts' is now a valid column in transformed\_data

transformed\_data['MultipleFailedLogins'] = transformed\_data['FailedLoginAttempts'].apply(lambda x: 1 if x >= TOO\_MANY\_FAILED\_ATTEMPTS else 0)

# Rule 3 : High Transaction Amount During Night Hours

# Example logic for defining 'IsNight' column

# Assuming 'HourOfDay' column exists and represents the hour of the transaction

# Assuming 'Time' column exists and represents time in seconds since a starting point

# Calculate 'HourOfDay' assuming 'Time' is in seconds (convert it to hours and get the hour part)

transformed\_data['HourOfDay'] = (transformed\_data['Time'] / 3600) % 24

# Now assuming 'HourOfDay' exists, define 'IsNight' based on 'HourOfDay'

# Night might be defined as hours between 22 and 6 for example

transformed\_data['IsNight'] = transformed\_data['HourOfDay'].apply(lambda x: 1 if (x >= 22 or x < 6) else 0)

HourOfDay: This feature represents the hour of the day

IsNight: This binary feature indicates whether the transaction happened during the night (from 10 PM to 6 AM). A value of 1 denotes a nighttime transaction, while 0 indicates daytime.

## Model Building:

# Import necessary libraries

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

# Assuming 'transformed\_data' DataFrame and it's already pre-processed

X = transformed\_data.drop('Class', axis=1) # Features

y = transformed\_data['Class'] # Target variable

# Split the dataset into the training set and test set

X\_train\_subset, X\_test\_subset, y\_train\_subset, y\_test\_subset = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Creating a smaller subset while maintaining class distribution using stratification

subset\_size = 0.1 # using 10% of the data

X\_subset, \_, y\_subset, \_ = train\_test\_split(X, y, test\_size=1 - subset\_size, random\_state=42, stratify=y)

# Splitting the subset into training (80%) and testing (20%) sets

X\_train\_subset, X\_test\_subset, y\_train\_subset, y\_test\_subset = train\_test\_split(

X\_subset, y\_subset, test\_size=0.2, random\_state=42, stratify=y\_subset)

# Checking the shape of the training and testing sets of the subset

X\_train\_subset.shape, X\_test\_subset.shape, y\_train\_subset.shape, y\_test\_subset.shape

logistic\_model = LogisticRegression(max\_iter=1000, random\_state=42)

logistic\_model.fit(X\_train\_subset, y\_train\_subset)

# Predicting on the test subset

logistic\_pred\_subset = logistic\_model.predict(X\_test\_subset)

# Evaluating the model on the subset

logistic\_accuracy\_subset = accuracy\_score(y\_test\_subset, logistic\_pred\_subset)

logistic\_conf\_matrix\_subset = confusion\_matrix(y\_test\_subset, logistic\_pred\_subset)

logistic\_class\_report\_subset = classification\_report(y\_test\_subset, logistic\_pred\_subset)

print("Logistic Regression Accuracy:", logistic\_accuracy\_subset)

print("Confusion Matrix for Logistic Regression:\n", logistic\_conf\_matrix\_subset)

print("Classification Report for Logistic Regression:\n", logistic\_class\_report\_subset)

### Random Forest model

from sklearn.ensemble import RandomForestClassifier

# Initializing the Random Forest model

random\_forest\_model = RandomForestClassifier(random\_state=42)

# Training the model on the subset

random\_forest\_model.fit(X\_train\_subset, y\_train\_subset)

# Predicting on the test subset

rf\_pred\_subset = random\_forest\_model.predict(X\_test\_subset)

# Evaluating the model on the subset

rf\_accuracy\_subset = accuracy\_score(y\_test\_subset, rf\_pred\_subset)

rf\_conf\_matrix\_subset = confusion\_matrix(y\_test\_subset, rf\_pred\_subset)

rf\_class\_report\_subset = classification\_report(y\_test\_subset, rf\_pred\_subset)

print("Random Forest Accuracy:", rf\_accuracy\_subset)

print("Confusion Matrix for Random Forest:\n", rf\_conf\_matrix\_subset)

print("Classification Report for Random Forest:\n", rf\_class\_report\_subset)

## Gradient Boosting

from sklearn.ensemble import GradientBoostingClassifier

# Initializing the Gradient Boosting model

gradient\_boosting\_model = GradientBoostingClassifier(random\_state=42)

# Training the model on the subset

gradient\_boosting\_model.fit(X\_train\_subset, y\_train\_subset)

# Predicting on the test subset

gb\_pred\_subset = gradient\_boosting\_model.predict(X\_test\_subset)

# Evaluating the model on the subset

gb\_accuracy\_subset = accuracy\_score(y\_test\_subset, gb\_pred\_subset)

gb\_conf\_matrix\_subset = confusion\_matrix(y\_test\_subset, gb\_pred\_subset)

gb\_class\_report\_subset = classification\_report(y\_test\_subset, gb\_pred\_subset)

print("Gradient Boosting Accuracy:", gb\_accuracy\_subset)

print("Confusion Matrix for Gradient Boosting:\n", gb\_conf\_matrix\_subset)

print("Classification Report for Gradient Boosting:\n", gb\_class\_report\_subset)

## SVM model

from sklearn.svm import SVC

# Initializing the SVM model

svm\_model = SVC(kernel='rbf', random\_state=42)

# Training the model on the subset

svm\_model.fit(X\_train\_subset, y\_train\_subset)

# Predicting on the test subset

svm\_pred\_subset = svm\_model.predict(X\_test\_subset)

# Evaluating the model on the subset

svm\_accuracy\_subset = accuracy\_score(y\_test\_subset, svm\_pred\_subset)

svm\_conf\_matrix\_subset = confusion\_matrix(y\_test\_subset, svm\_pred\_subset)

svm\_class\_report\_subset = classification\_report(y\_test\_subset, svm\_pred\_subset)

print("SVM model Accuracy:", svm\_accuracy\_subset)

print("Confusion Matrix for SVM model:\n", svm\_conf\_matrix\_subset)

print("Classification Report for SVM model:\n", svm\_class\_report\_subset)

## Decision Tree model

from sklearn.tree import DecisionTreeClassifier

# Initializing the Decision Tree model

decision\_tree\_model = DecisionTreeClassifier(random\_state=42)

# Training the model on the subset

decision\_tree\_model.fit(X\_train\_subset, y\_train\_subset)

# Predicting on the test subset

dt\_pred\_subset = decision\_tree\_model.predict(X\_test\_subset)

# Evaluating the model on the subset

dt\_accuracy\_subset = accuracy\_score(y\_test\_subset, dt\_pred\_subset)

dt\_conf\_matrix\_subset = confusion\_matrix(y\_test\_subset, dt\_pred\_subset)

dt\_class\_report\_subset = classification\_report(y\_test\_subset, dt\_pred\_subset)

print(" Decision Tree Accuracy:", dt\_accuracy\_subset)

print("Confusion Matrix for Decision Tree:\n", dt\_conf\_matrix\_subset)

print("Classification Report for Decision Tree:\n", dt\_class\_report\_subset)

# visualizations to compare the performance of these models. We will plot bar charts for accuracy, precision, recall, and F1 score for the positive class (class 1, which indicates fraud).

import matplotlib.pyplot as plt

import numpy as np

# List of models

models = ['Logistic Regression', 'Random Forest', 'Gradient Boosting', 'SVM', 'Decision Tree']

# Performance metrics for each model

accuracy\_scores = [0.9989, 0.9991, 0.9989, 0.9982, 0.9989]

precision\_scores = [0.83, 0.86, 0.64, 0.00, 0.75]

recall\_scores = [0.50, 0.60, 0.90, 0.00, 0.60]

f1\_scores = [0.62, 0.71, 0.75, 0.00, 0.67]

# Setting the positions and width for the bars

positions = np.arange(len(models))

width = 0.15

# Plotting

plt.figure(figsize=(15, 7))

plt.bar(positions - width\*2, accuracy\_scores, width, label='Accuracy')

plt.bar(positions - width, precision\_scores, width, label='Precision (Class 1)')

plt.bar(positions, recall\_scores, width, label='Recall (Class 1)')

plt.bar(positions + width, f1\_scores, width, label='F1 Score (Class 1)')

plt.xlabel('Models')

plt.ylabel('Scores')

plt.title('Comparison of Model Performance')

plt.xticks(positions, models)

plt.legend()

plt.show()

# pip install --upgrade scikit-learn

# Random\_forest\_model confusion matrix

from sklearn.metrics import confusion\_matrix

import seaborn as sns

# Assume 'model' is your trained model and 'X\_test', 'y\_test' are your test datasets

y\_pred = random\_forest\_model.predict(X\_test\_subset)

cm = confusion\_matrix(y\_test\_subset, y\_pred)

sns.heatmap(cm, annot=True, fmt='d')

plt.title('Confusion Matrix')

plt.ylabel('Actual Label')

plt.xlabel('Predicted Label')

plt.show()

# Hyperparameter : RandomizedSearchCV

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

from sklearn.ensemble import GradientBoostingClassifier

from sklearn.model\_selection import RandomizedSearchCV

from sklearn.linear\_model import LogisticRegression

from sklearn.ensemble import RandomForestClassifier

from sklearn.svm import SVC

from sklearn.tree import DecisionTreeClassifier

from scipy.stats import uniform, randint

n\_iter\_search = 5 # You can adjust this number as needed

# Logistic Regression

# Logistic Regression parameters and RandomizedSearchCV setup

logistic\_params = {

'C': uniform(0.001, 100),

'penalty': ['l1', 'l2']

}

logistic\_random\_search = RandomizedSearchCV(

LogisticRegression(solver='liblinear'),

param\_distributions=logistic\_params,

n\_iter=n\_iter\_search,

cv=5,

scoring='accuracy',

random\_state=42,

verbose=1

)

# Fit the Logistic Regression model with RandomizedSearchCV

logistic\_random\_search.fit(X\_train\_subset, y\_train\_subset)

# Get the best Logistic Regression model

best\_logistic\_model = logistic\_random\_search.best\_estimator\_

# Evaluate the best models on the test data

logistic\_pred = best\_logistic\_model.predict(X\_test\_subset)

logistic\_accuracy = accuracy\_score(y\_test\_subset, logistic\_pred)

logistic\_conf\_matrix = confusion\_matrix(y\_test\_subset, logistic\_pred)

logistic\_class\_report = classification\_report(y\_test\_subset, logistic\_pred)

print("Logistic Regression Accuracy:", logistic\_accuracy)

print("Confusion Matrix for Logistic Regression:\n", logistic\_conf\_matrix)

print("Classification Report for Logistic Regression:\n", logistic\_class\_report)

# Random Forest model

random\_forest\_params = {

'n\_estimators': [50, 100], # Reduced number of options

'max\_depth': [None] + list(randint(10, 30).rvs(3)), # Reduced range and values

'min\_samples\_split': [2, 5], # Reduced options

'min\_samples\_leaf': [1, 3] # Reduced options

}

random\_forest\_search = RandomizedSearchCV(

RandomForestClassifier(),

param\_distributions=random\_forest\_params,

n\_iter=n\_iter\_search,

cv=5,

scoring='accuracy',

random\_state=42,

verbose=1

)

# Fit the Random Forest model with RandomizedSearchCV

random\_forest\_search.fit(X\_train\_subset, y\_train\_subset)

# Get the best Random Forest model

best\_random\_forest\_model = random\_forest\_search.best\_estimator\_

random\_forest\_pred = best\_random\_forest\_model.predict(X\_test\_subset)

random\_forest\_accuracy = accuracy\_score(y\_test\_subset, random\_forest\_pred)

random\_forest\_conf\_matrix = confusion\_matrix(y\_test\_subset, random\_forest\_pred)

random\_forest\_class\_report = classification\_report(y\_test\_subset, random\_forest\_pred)

print("Random Forest Accuracy:", random\_forest\_accuracy)

print("Confusion Matrix for Random Forest:\n", random\_forest\_conf\_matrix)

print("Classification Report for Random Forest:\n", random\_forest\_class\_report)

# Decision Tree

# Decision Tree parameters and RandomizedSearchCV setup

decision\_tree\_params = {

'max\_depth': [None] + list(randint(10, 20).rvs(3)), # Reduced range and smaller values

'min\_samples\_split': randint(2, 6), # Reduced range

'min\_samples\_leaf': randint(1, 3) # Reduced range

}

decision\_tree\_search = RandomizedSearchCV(

DecisionTreeClassifier(),

param\_distributions=decision\_tree\_params,

n\_iter=n\_iter\_search,

cv=5,

scoring='accuracy',

random\_state=42,

verbose=1,

n\_jobs=-1 # Utilize all available CPU cores

)

# Fit the Decision Tree model with RandomizedSearchCV

decision\_tree\_search.fit(X\_train\_subset, y\_train\_subset)

# Get the best Decision Tree model

best\_decision\_tree\_model = decision\_tree\_search.best\_estimator\_

# Best hyperparameters

best\_params = random\_search.best\_params\_

print("Best Parameters:", best\_params)

decision\_tree\_pred = best\_decision\_tree\_model.predict(X\_test\_subset)

decision\_tree\_accuracy = accuracy\_score(y\_test\_subset, decision\_tree\_pred)

decision\_tree\_conf\_matrix = confusion\_matrix(y\_test\_subset, decision\_tree\_pred)

decision\_tree\_class\_report = classification\_report(y\_test\_subset, decision\_tree\_pred)

print("Decision Tree Accuracy:", decision\_tree\_accuracy)

print("Confusion Matrix for Decision Tree:\n", decision\_tree\_conf\_matrix)

print("Classification Report for Decision Tree:\n", decision\_tree\_class\_report)

# Gradient Boosting

param\_dist = {

'n\_estimators': [100, 200], # Fewer options

'learning\_rate': [0.1, 0.2],

'max\_depth': [3, 4],

'min\_samples\_split': [2, 3]

}

random\_search = RandomizedSearchCV(

estimator=GradientBoostingClassifier(random\_state=42),

param\_distributions=param\_dist,

n\_iter=5, # Reduced number of parameter settings that are sampled

scoring='accuracy',

cv=3,

random\_state=42

)

# Fit the random search model to find optimal hyperparameters

random\_search.fit(X\_train\_subset, y\_train\_subset)

# Best model after random search

best\_gb\_model = random\_search.best\_estimator\_

# Best hyperparameters

best\_params = random\_search.best\_params\_

print("Best Parameters:", best\_params)

# Predictions using the best Gradient Boosting model

gb\_best\_pred = best\_gb\_model.predict(X\_test\_subset)

# Evaluation

gb\_best\_accuracy = accuracy\_score(y\_test\_subset, gb\_best\_pred)

gb\_best\_conf\_matrix = confusion\_matrix(y\_test\_subset, gb\_best\_pred)

gb\_best\_class\_report = classification\_report(y\_test\_subset, gb\_best\_pred)

print(" Gradient Boosting Accuracy:", gb\_best\_accuracy)

print("Confusion Matrix for Gradient Boosting:\n",gb\_best\_conf\_matrix )

print("Classification Report:\n", gb\_best\_class\_report)

# svm model

from scipy.stats import uniform

from sklearn.svm import SVC

from sklearn.model\_selection import RandomizedSearchCV

svm\_params = {

'C': uniform(0.5, 0.5), # Simplified to a narrower range around a reasonable default

'kernel': ['rbf'] # Stick to one kernel type for speed

}

n\_iter\_search = 5 # Further reduced number of iterations

svm\_search = RandomizedSearchCV(

SVC(probability=True, max\_iter=1000), # Added max\_iter

param\_distributions=svm\_params,

n\_iter=n\_iter\_search,

cv=3, # Reduced CV folds

scoring='accuracy',

random\_state=42,

verbose=2, # Increased verbosity

n\_jobs=4 # Reduced jobs if necessary

)

# Fit the random search model to find optimal hyperparameters

svm\_model.fit(X\_train\_subset, y\_train\_subset)

# Best model after random search

best\_svm\_model = random\_search.best\_estimator\_

# Best hyperparameters

best\_params = random\_search.best\_params\_

print("Best Parameters:", best\_params)

# Predictions using the best Gradient Boosting model

svm\_best\_pred = best\_svm\_model.predict(X\_test\_subset)

# Evaluation

svm\_model\_best\_accuracy = accuracy\_score(y\_test\_subset, svm\_best\_pred)

svm\_model\_best\_conf\_matrix = confusion\_matrix(y\_test\_subset,svm\_best\_pred)

svm\_model\_best\_class\_report = classification\_report(y\_test\_subset, svm\_best\_pred)

print("svm\_model Accuracy:", svm\_model\_best\_accuracy)

print("Confusion Matrix for svm\_model:\n",svm\_model\_best\_conf\_matrix )

print("Classification Report:\n", svm\_model\_best\_class\_report)

# Neural Networks

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

# Assume 'Class' is the target variable

X = transformed\_data.drop('Class', axis=1)

y = transformed\_data['Class']

# Splitting the data into train and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Scaling features

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

from tensorflow.keras.callbacks import EarlyStopping

from tensorflow.keras.layers import Dropout

# Setting a random seed for reproducibility

tf.random.set\_seed(42)

# Defining the model

model = Sequential()

# Define early stopping callback

early\_stopping = EarlyStopping(

monitor='val\_loss', # Monitor validation loss

patience=3, # Number of epochs with no improvement after which training will be stopped

restore\_best\_weights=True # Restores model weights from the epoch with the best value of the monitored quantity

)

model = Sequential()

model.add(Dense(10, input\_dim=X\_train\_scaled.shape[1], activation='relu'))

model.add(Dropout(0.2)) # Dropout layer

model.add(Dense(8, activation='relu'))

model.add(Dropout(0.2)) # Another dropout layer

model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary\_crossentropy', optimizer='adam', metrics=['accuracy'])

history = model.fit(

X\_train\_scaled, y\_train,

validation\_split=0.2,

epochs=20, # You might increase this if you're using early stopping

batch\_size=32,

verbose=1,

callbacks=[early\_stopping] # Include early stopping here

)

import matplotlib.pyplot as plt

# Manually extracted accuracy values for training and validation sets

train\_accuracy = [0.9874, 0.9990, 0.9991, 0.9989, 0.9990, 0.9991]

val\_accuracy = [0.9989, 0.9991, 0.9991, 0.9990, 0.9991, 0.9991]

# Epochs are simply the range from 1 to the number of epochs have data for

epochs = range(1, len(train\_accuracy) + 1)

# Plotting training and validation accuracy

plt.figure(figsize=(10, 5))

plt.plot(epochs, train\_accuracy, 'bo-', label='Training accuracy')

plt.plot(epochs, val\_accuracy, 'ro-', label='Validation accuracy')

plt.title('Training and Validation Accuracy')

plt.xlabel('Epoch')

plt.ylabel('Accuracy')

plt.legend()

plt.show()

import matplotlib.pyplot as plt

# Manually extracted loss values for training and validation sets

train\_loss = [0.0452, 0.0071, 0.0058, 0.0054, 0.0050, 0.0046]

val\_loss = [0.0042, 0.0036, 0.0032, 0.0033, 0.0032, 0.0033]

# Epochs are simply the range from 1 to the number of epochs have data for

epochs = range(1, len(train\_loss) + 1)

# Plotting training and validation loss

plt.figure(figsize=(10, 5))

plt.plot(epochs, train\_loss, 'bo-', label='Training loss')

plt.plot(epochs, val\_loss, 'ro-', label='Validation loss')

plt.title('Training and Validation Loss')

plt.xlabel('Epoch')

plt.ylabel('Loss')

plt.legend()

plt.show()

# Model Evaluation:

from sklearn.metrics import roc\_auc\_score

# Calculating ROC-AUC for each model

# For models like Logistic Regression and SVM that use decision functions, we use 'decision\_function' method

# For tree-based models, we use 'predict\_proba' method

# Logistic Regression

logistic\_roc\_auc = roc\_auc\_score(y\_test\_subset, logistic\_model\_subset.decision\_function(X\_test\_subset))

# Random Forest

rf\_probs = random\_forest\_model.predict\_proba(X\_test\_subset)[:, 1] # Probability of the positive class

rf\_roc\_auc = roc\_auc\_score(y\_test\_subset, rf\_probs)

# Gradient Boosting

gb\_probs = gradient\_boosting\_model.predict\_proba(X\_test\_subset)[:, 1]

gb\_roc\_auc = roc\_auc\_score(y\_test\_subset, gb\_probs)

# SVM (using decision function)

svm\_roc\_auc = roc\_auc\_score(y\_test\_subset, svm\_model.decision\_function(X\_test\_subset))

# Decision Tree

dt\_probs = decision\_tree\_model.predict\_proba(X\_test\_subset)[:, 1]

dt\_roc\_auc = roc\_auc\_score(y\_test\_subset, dt\_probs)

logistic\_roc\_auc, rf\_roc\_auc, gb\_roc\_auc, svm\_roc\_auc, dt\_roc\_auc

import matplotlib.pyplot as plt

# ROC-AUC scores for each model

roc\_auc\_scores = {

'Logistic Regression': 0.9997889553288779,

'Random Forest': 0.9998417164966584,

'Gradient Boosting': 0.9970189940204011,

'SVM': 0.515089693985227,

'Decision Tree': 0.7998241294407317

}

# Create lists for the plot

models = list(roc\_auc\_scores.keys())

scores = list(roc\_auc\_scores.values())

# Create the bar plot

plt.figure(figsize=(10, 5))

plt.bar(models, scores, color=['blue', 'green', 'red', 'purple', 'orange'])

plt.xlabel('Models')

plt.ylabel('ROC-AUC Score')

plt.title('Comparison of Model ROC-AUC Scores')

plt.ylim([0, 1.1]) # Extend y-axis to fit labels

plt.xticks(rotation=45) # Rotate model names for better readability

# Adding the text labels on the bars

for i in range(len(models)):

plt.text(i, scores[i] + 0.02, f'{scores[i]:.4f}', ha = 'center')

plt.tight\_layout()

plt.show()