Bachelor of Science in Computer Science & Engineering

Developing A Framework for Credit Card Fraud Detection

by

Yeasin Arafath

ID: 1504038

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Chattogram-4349, Bangladesh.

April, 2021

Developing A Framework for Credit Card Fraud Detection

Submitted in partial fulfilment of the requirements for
Degree of Bachelor of Science
in Computer Science & Engineering

by

Yeasin Arafath
ID: 1504038

Supervised by

Prof. Dr. Mohammad Shamsul Arefin
Professor
Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)
Chattogram-4349, Bangladesh.

The thesis titled '**Developing A Framework for Credit Card Fraud Detection'** submitted by ID: 1504038, Session 2019-2020 has been accepted as satisfactory in fulfilment of the requirement for the degree of Bachelor of Science in Computer Science & Engineering to be awarded by the Chittagong University of Engineering & Technology (CUET).

Board of Examiners

Chairman

Prof. Dr. Mohammad Shamsul Arefin
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Member (Ex-Officio)

Dr. Asaduzzaman
Professor & Head
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Member (External)

Dr. Mohammed Moshiul Hoque
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Declaration of Originality

This is to certify that I am the sole author of this thesis and that neither any part of this thesis nor the whole of the thesis has been submitted for a degree to any other institution.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. I am also aware that if any infringement of anyone's copyright is found, whether intentional or otherwise, I may be subject to legal and disciplinary action determined by Dept. of CSE, CUET.

I hereby assign every rights in the copyright of this thesis work to Dept. of CSE, CUET, who shall be the owner of the copyright of this work and any reproduction or use in any form or by any means whatsoever is prohibited without the consent of Dept. of CSE, CUET.

____

Signature of the candidate

Date:

Acknowledgements

The success and final result of this thesis involved a great deal of support and assistance from many people, and I consider myself incredibly fortunate to have received it all the way through my thesis. Professionally and socially, it has been a rewarding experience. Much of what I've accomplished has been possible only because of such oversight and assistance.Following that, I'd like to express my heartfelt gratitude to Prof. Dr. Mohammad Shamsul Arefin, Department of Computer Science Engineering, Chittagong University of Engineering Technology (CUET), for his guidance, motivation, and unwavering support during my thesis work. I appreciate his many probing questions, which force me to consider things from various angles throughout the entire time.

I owe my gratitude to Dr. Asaduzzaman, Head Professor, Department of Computer Science Engineering, Chittagong University of Engineering Technology (CUET) who took a keen interest in my work and guided me by providing the necessary information's. I am thankful for the constant encouragement, support, and guidance from all Teaching staffs of the department.

Finally, I want to thank my father and mother for their unconditional love, support, encouragement, and contribution all throughout my life and academic career in every aspect along the years.

Abstract

Credit card use has expanded as new developments emerge. Credit card fraud is on the rise as credit cards become the most used online and manual payment method. Credit card theft must be stopped because it involves huge financial damage. Several modern approaches [1] focused on artificial intelligence, data processing, deep learning, sequence matching, genetic programming, and other technologies can be used to detect fraudulent transactions. We employed deep learning algorithms. Machine Learning is a branch of Artificial Intelligence. It is a technical examination of the algorithms and mathematical models used by computer systems to accomplish a specific role efficiently without the use of explicit commands, but focusing on patterns and inferences. Machine learning is a technique that allows us to train computers using various algorithms to make them capable of making our own decisions. Many algorithms, such as Logistic Regression, K-Neighbors Classifier, Random Forest Classifier, LinearSVC, GaussianNB, Decision Tree Classifier, and others [2], can be used to detect defect fraud in machine learning. We have conducted a quantitative study of these algorithms.

Keywords: Credit Card Fraud Detection, Logistic Regression, K-Neighbors Classifier, Random Forest Classifier, LinearSVC, GaussianNB, Decision Tree Classifier.

Table of Contents

Acknowledgements **iii**

Abstract **iv**

List of Figures **vii**

1 Introduction **1**

1.1 Introduction 1

1.2 Credit Card Fraud Detection Framework 1

1.3 Difficulties 2

1.4 Applications 3

1.5 Motivation 3

1.6 Contribution of the thesis 4

1.7 Thesis Organization 4

1.8 Conclusion 5

2 Literature Review **6**

2.1 Introduction 6

2.2 Related Literature Review on Credit Card Fraud Detection 6

2.3 Conclusion 8

2.3.1 Implementation Challenges 8

3 Methodology **10**

3.1 Introduction 10

3.2 Steps of Proposed Credit Card Fraud Detection Framework 10

3.3 Data Pre-processing 11

3.4 Model Training And Testing 12

3.5 Model Hyperparameter Tuning with RandomizedSerachCV 12

3.6 Conclusion 13

4 Results and Discussions **14**

4.1 Introduction 14

4.2 Dataset Description 14

4.3 Impact Analysis 15

4.3.1 Social and Environmental Impact 15

4.3.2 Ethical Impact 16
4.4 Evaluation of Credit Card Fraud Detection Framework 17
4.5 Evaluation of Performance 20
4.6 Conclusion 23

5 Conclusion 29
5.1 Conclusion 29
5.2 Future Work 30

List of Figures

3.1 Splitting of Dataset 10
3.2 Steps of the proposed framework. 11

4.1 Exploratory Data Analysis and Plotting Libraries 17
4.2 Models from Scikit-Learn 18
4.3 Model Selection And Evaluation 18
4.4 Load Data 19
4.5 Data Frame 19
4.6 Value Counts for Each Class 20
4.7 Value Counts for Classes 20
4.8 Data Info 21
4.9 Comparison Class Column with Amount Column 21
4.10 Credit Card Fraud Detection in Function of Amount And Time . 22
4.11 Correlation Matrix 23
4.12 Splitting Data into X and y 23
4.13 Split Data into Train And Test Sets 24
4.14 List of Models 24
4.15 Function to Fit And Score Models 24
4.16 Baseline Models Scores 24
4.17 Model Comparisons 25
4.18 Hyperparameter Grid for Random Forest Classifier 25
4.19 Random Hyperparameter Search for RandomForestClassifier . . . 25
4.20 Best Params for Random Hyperparameter Search Model for RandomForestClassifier 26
4.21 ROC Curve 26
4.22 Confusion Matrix 27
4.23 Classification Report 27
4.24 Cross-Validated Classifications Metrics 28

Chapter 1

Introduction

1.1 Introduction

Credit card theft is the fraudulent use of credit card details without the consent of the card's holders. The credit card can be used in person or online. Cardholders use their keys at the end of the dealership in the event of physical use. The fraudster must procure the card in physical form by deception and then use it to commit fraud.

To commit fraud, information such as Card Verification Value (CVV code), expiry date, card number, and pin code are required for internet card transactions. Fraudsters collect card data by intercepting e-mails, phishing, and skimming victims' online transactions.

This chapter will include an outline of the Credit card fraud detection framework, as well as an explanation of the inherent complexities of this issue. This chapter would also discuss the motivation and contribution of this particular study.

1.2 Credit Card Fraud Detection Framework

Classification algorithms in machine learning are widely regarded as a very prominent and successful research area in the field of fraud detection [3]. Classification is the method of categorizing a given collection of data into groups. It can be done with both structured and unstructured data. Predicting the class of provided data points is the first step in the process. The groups are also known as the objective, mark, or divisions. Classification algorithms include linear classifiers, support vector machines, decision trees, and others.

In this application, a hybrid feature extraction approach that blends various features is used to detect credit card fraud. The key stages are as follows:

1. To obtain the generalized format of the dataset, preprocessing steps are carried out.

2. Baseline models are trained and score using the training and testing dataset.
3. Among the baseline models, best accurate models is setupr for random hyperparameter tuning with randomizedsearchcv.
4. Best params of tuned model are evaluated.

1.3 Difficulties

One of the most studied fields of fraud detection is credit card fraud detection, which relies on automatic analysis of recorded transactions to determine fraudulent activity. Fraud detection systems are vulnerable to a range of issues and obstacles, which are described below. An successful fraud detecting method should be capable of dealing with these issues in order to provide the best results.

1. **Imbalanced Data:** The evidence on credit card fraud is distorted, meaning that only a small percentage of all credit card transactions are fraudulent. Because of this, identifying illegal transactions is complex and imprecise.

2. **Overlapping Data:** Many transactions may be deemed fraudulent while still appearing to be normal (fake positive), and a fraudulent transaction may appear to be real (fake negative). As a result, achieving a low false positive and false negative rate is a significant challenge for fraud detection systems.

3. **Lack of Adaptability:** The dilemma of adaptability often confronts classification algorithms.Both controlled and unsupervised fraud detection mechanisms are inefficient at identifying new patterns of ordinary and fraudulent conduct.

4. **Fraud Detection Cost:** The method should consider both the importance of the detected fraudulent transaction and the expense of stopping it. Stopping a dishonest sale of a few dollars, for example, yields little money.

5. **Lack of Standard Metrics:** There is no common assessment criteria for evaluating and comparing the outcomes of fraud detection systems in order to determine which is the most efficient.

1.4 Applications

The Credit Card Fraud Detection can be used in a variety of situations. Fraud detection from this process may be used for a wide range of applications,

1. Online banking system.

2. Online payment system using credit card.

3. Online transactions.

1.5 Motivation

The vast majority of transactions are now conducted electronically, necessitating the use of credit cards and other online payment services. Both the business and the customer prosper from this approach. Consumers save time by not needing to go to the supermarket to place their orders, and businesses save money by not having to buy physical stores to save costly rent costs.

The new era seems to have incorporated some very useful features that have changed how companies and consumers interact with one another, but at a cost. Businesses must hire trained software engineers and penetration testers to ensure that all transactions are genuine and not fake.

These individuals are building the company's servers in such a manner that the customer has no influence over vital transaction components such as payment number. Most (if not all) of the issues can be avoided with proper design, but even the architecture used to construct the server is not flawless.

1.6 Contribution of the thesis

Thesis or research work is done to accomplish a particular set of targets, such as defining a new approach or improving existing ones. The primary goal of this work was to increase the identification accuracy of credit card fraud detection. The primary contribution of this thesis is the following:

1. Compare between different machine learning models and evaluate the best models based on accuracy.
2. Setup random hyperparameter models with randomizedsearchcv.
3. Evaluate the best params of the tuned model.
4. Score the hyperparameter tuned model with best params.

1.7 Thesis Organization

The rest of this thesis report is organized as follows:

- Chapter 2 gives a brief summary of previous research works in the field of Credit Card Fraud Detection.
- Chapter 3 describes the proposed methodology for the detection of Credit Card Fraud. In the proposed framework, after data pre-processing, data is splitted into training and testing data. Six different machine learning models are trained using the training data, and evaluate the models using the testing data. This models are considered as baseline models. From all the baseline models, best score models is setup for hyperparamter tuning with randomizedsearchcv and evaluate the best params of the tuned model with score.
- Chapter 4 provides the description of the working dataset and analysis of the performance measure for the proposed framework.
- Chapter 5 contains the overall summary of this thesis work and provides some future recommendations as well.

1.8 Conclusion

This chapter provides a description of credit card transactions and their significance in our lives. This chapter describes the Credit Card Fraud Detection Framework, as well as the complexities. This section also discusses the inspiration for this work and contributions. The history and current state of the issue will be discussed in the following chapter.

Chapter 2

Literature Review

2.1 Introduction

Several approaches to bringing strategies to detect fraud have been proposed in previous research, ranging from regulated approaches to unsupervised approaches to hybrid approaches, making it possible to review the technologies associated with credit card fraud detection and to get a clearer understanding of the forms of credit card fraud. As time progressed, fraud patterns evolved, introducing new forms of fraud, making it a popular research subject. The rest of this section goes into individual machine learning algorithms, machine learning models, and fraud detection systems that are used in fraud detection. The issues found during the study have been researched in order to later implement an efficient machine learning model.

2.2 Related Literature Review on Credit Card Fraud Detection

Researchers suggested a vast range of credit card fraud prevention algorithms, the majority of which were focused on neural network and data mining approaches. Niu at al. [4] discussed on a comparison study of credit card fraud detection: supervised versus unsupervised. S. Benson Edwin Raj and A. Annie Portia [5] suggested a paradigm that operates in two stages: preparation and detection. The credit card holder's shopping behaviour is evaluated using the k-means clustering algorithm during the training process, and the sequence is assembled during the detection phase. If the present transaction fits the chain, it is considered legitimate; otherwise, it is considered fraudulent. Aihua Shen et al.[6] suggested a two-stage method for detecting credit card fraud. In the first step, using sequence alignment, a successful score is determined based on true cardholder transaction history and transaction behavioral changes. In the second point, the bad score is calculated by using the fraudulent transaction signature provided by the previous fraudulent transaction. If the difference between the good and poor scores exceeds a predetermined threshold, the transaction is illegal; otherwise, it is legal.

Khyati Chaudhary et al. [7] suggested a BLAHFDS hybridization of BLASTA – SSAHA for detecting credit card fraud. This model has two stages. The first step is the profile analyzer, which compares the time and sequence of the current transaction to the transactional record. The second stage is the deviation analyzer, which compares the deviated time-amount series to the fraud history index. It computes the cumulative variance between the profile score and the deviation score. The overall discrepancy is used to spot fraud. AlejandroCorrea Bahnsen et al.[8] suggested GASS, a combination of two common algorithms, Genetic Algorithm (GA) and Scatter Search (SS). GASS incorporates certain SS components into the GA steps. The steps are as follows: number of parent options, number of offspring, succession, mutation operator, recombination and mutation probability, fitness function, selection and termination criteria, fitness function, selection and termination criteria. The proposed method aims to reduce classification costs. Delamaire et al.[9] proposed a model for identifying fraud that classifies the signs and features of fraudulent transactions using Bayesian classification and an interaction law. The rules and patterns that are produced are used to detect fraud. Save et al.[10] discussed on the idea for credit card fraud detection using decision tree.

Awoyemi et al.[11] proposed a paradigm based on transaction aggregation. They aggregated the transaction and developed the customer's buying behaviour in this model. These behaviors are used to detect fraudulent credit card transactions. Delamaire et al.[12] extended the purchase aggregation approach to observe customers' periodic purchasing actions. They improved fraud prevention by using feature processing and cost awareness. Xuan et al.[13] used an interaction rule to create natural behaviour patterns from a false transactional database. This trends are used to spot fraud. For fraud prevention, Dal Pozzolo et al.[14] used the self-organizing map (SOM). SOM is used for previous transactional data classification and clustering, deriving secret patterns from previous data, and as a filtering tool. Patidar et al.[15] suggested an innovative solution that combines network-based and inherent functions. The intrinsic function determines how the latest requested transaction differs from previous transactions in terms of the card's Regency–Frequency–Monetary parameters (RFM). The merchant-card relationship is a network-based mechanism that produces a time-dependent suspicious score for each merchant. Ekrem et al.[16] proposed a cost-sensitive decision tree method for identifying fraudulent credit card transactions and reducing the cost of misclassification.

2.3 Conclusion

A thorough literature review is explored in this chapter. This discussion has been split into simple Credit Card Fraud Detection sections for convenience. The researchers' feature extraction techniques and classifiers are listed here. The following chapter provides a thorough description of the suggested technique for detecting credit card fraud.

2.3.1 Implementation Challenges

As several authors [8] [12] [14] [17] have stated, one of the most significant problems associated with credit card fraud detection is the lack of datasets from which researchers may conduct study. The explanation for the lack of real-world data is that banks and financial institutions are unable to disclose confidential consumer activity data for privacy purposes.

Credit card fraud databases contain highly distorted results, with far more legitimate transactions than fraudulent transactions, and the lawful and fraudulent transactions differ by at least a hundred times. In practice, 98 percent of transactions are legitimate, while just 2 percent are fraudulent.

According to Quah et al in[18] millions of credit card transactions are processed every day. Analyzing such large numbers of transactions necessitates highly qualified methods that scale well, as well as a considerable amount of computational power. It places some restrictions on the researchers.

Fraudsters with complex behavior [19] alter their behavior over time in order to avoid detection by new detection systems and adapt fraud types. As a result, fraud is becoming more difficult and advanced, to the point that human experts are unable to forecast it.

Chapter 3

Methodology

3.1 Introduction

Credit card misuse is the illegal use of a credit card number without the permission of the cardholder. Credit cards can be used both physical and online. Anomaly detection strategies for credit card fraud are categorized as regulated or unsupervised. The use of supervised methods has several drawbacks. If a suspicious transaction occurs and is not conformed to the database, these transactions are considered normal, while anomaly events are found by new transactions and happened reports of unsupervised approaches. There are 492 illegal transactions in the sample of 2,84,807 transactions. Since the amount of fraudulent transactions is very limited, once machine learning models are trained using data, models become overfit [20]. In our research work, we have minimized overfitting and increased the model's accuracy based on hyperparameter tuning best parameters.

3.2 Steps of Proposed Credit Card Fraud Detection Framework

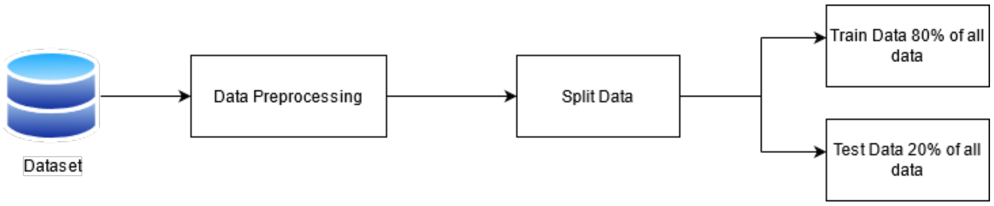


Figure 3.1: Splitting of Dataset

The basic steps of the suggested protocol for Credit card fraud identification process are depicted in Figures 3.1 and 3.2. Figure 3.1 shows data preprocessing

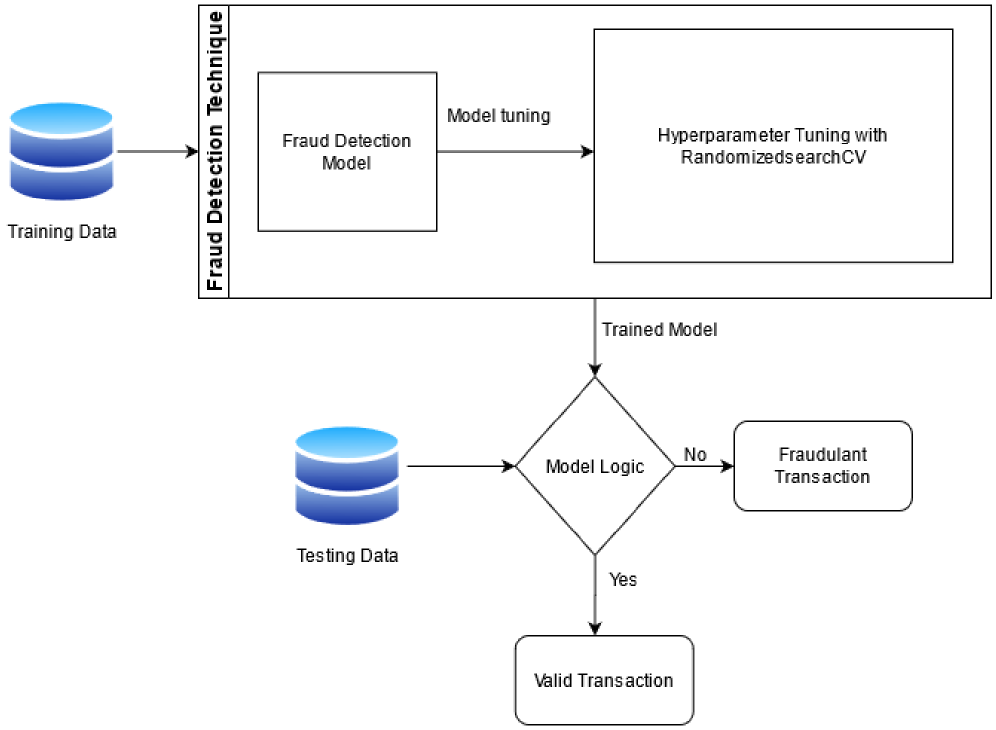


Figure 3.2: Steps of the proposed framework.

such as evaluating the data and correcting missing data after loading the dataset from CSV format to dataframe. Following data preprocessing, the entire dataset is divided into two collections, with 80 percent of the data processed as training data and the other 20 percent stored as testing data.

Figure 3.2 shows how, after training the machine learning algorithm with training data, the dependent model improves by hyperparameter tuning with randomizedsearchcv, and model logic is built up. After refining the simple model with the right parameters, the model is checked by analyzing data and determining whether the transaction is true or fraudulent. If the model rationale is yes, it implies that the transaction is valid; otherwise, it indicates that the transaction is fraudulent.

3.3 Data Pre-processing

The credit card fraud identification dataset was collected from Kaggle in CSV format. The CSV dataset is first loaded into the module, which tests the entire dataset as a data frame. After reviewing the entire dataset, searching for missing data, and restoring any missing data, the data can be divided into training and analysis data. After the initial data preprocessing, split the whole data into input variables X and output variables y. And using the train test split method, data of input variable and output variable, X and y, is splitted into train and test data, where test size is assumed as 20 percent. The credit card fraud monitoring system considers 80 percent of all data to be training data, while the other 20 percent is believed to be study data.

3.4 Model Training And Testing

We considered six separate machine learning models in our research work: Logistic Regression, K-Neighbors Classifier, Random Forest Classifier, LinearSVC, GaussianNB, and Decision Tree Classifier.

The fit and score system is used to train the models. As model parameters, models, train data, and test data are passed.The NumPy random seed value of 42 is used to set the random shown in NumPy. Get the name and model of each model by looping through the model objects. The input training data and output training data are used to match models. Models are scored on the research results after they have been developed using the training data. Inside the suit and score process, testing data of input variables and output variables are used to score the model's results. The model scores for each model are saved in a list, and the model scores list is eventually returned by the suit and score process.

3.5 Model Hyperparameter Tuning with RandomizedSerachCV

Hyperparameter Tuning with RansomizedSearchCV is used for tuning to boost the baseline models. Build a hyperparameter grid for each model for Hyperparameter Tuning with RandomizedSerachCV. A hyperparameter grid with the values of several estimators, such as max width, min samples break, min sample leaf, and so on. After building the baseline model's hyperparameter grids, the models are ready for tuning with Randomized SerachCV. Numpy random seed is initially set to 42. RandomizedSearchCV employs the RandomizedSearchCV approach from sklearn model collection. The RandomizedSearchCV system parameters passed are models, parameter distributions as hyperparameter grid, cv, number of iters, and verbose. This creates a setup for random hyperparameter quest for models. For models with input and output training results, fit the random hyperparameter search model. The best params are evaluated from the models grid system using the best params attribute after fitting the random hyperparameter check for models. The best params attributes indicate the best valued parameters for the random hyperparameter search model. Rate the hyperparameter tuning algorithm by evaluating the RandomsearchCV model score using input and output testing data and the RandomsearchCV grid process.

3.6 Conclusion

This chapter discusses a technique for the Credit card fraud identification framework. To evaluate and improve the models, hyperparameter tuning with randomized search cv is used for model tuning, and after tuning the model, this chapter compares the models based on their accuracy. The next chapter is about the experimental result analysis of the proposed framework.

Chapter 4

Results and Discussions

4.1 Introduction

A thorough description of the proposed system for credit card fraud detection was given in the previous chapter. The efficiency of the proposed structure is examined in this chapter.

This architecture was developed in Python 3.7 with an Intel Core i7 processor and 8GB RAM. The dataset 'Credit Card Fraud Detection' [19] was used for this thesis work. This data was obtained from Kaggle.

4.2 Dataset Description

The chosen dataset contains information on cardholders who used a credit card in September 2013. In the survey of 2,84,807 transactions, there are 492 illegal transactions. The dataset in question is in comma-separated variables (CSV) format. CSV files are used to store tabular data. As a result of the PCA (Principal Component Analysis) translation of input values, dataset values are in numerical form. This conversion is performed to ensure that the user's sensitive information remains hidden and that the user's protection is protected.

Columns with heads ranging from V1 to V28 display PCA transformed numeric values, while time, number, and class features display their true values. When working with large datasets, it is not always possible to perform detailed observations on each value; hence, graphical representation of data facilitates observation. Time, number, class, and columns V1 to V28 are represented in the form of a Histogram in this dataset, for a total of 31 features. The distribution of numerical data is correctly represented by a histogram. The time feature shows the amount of time that has passed in transactions, while the number feature displays the actual transaction amount. Class is a consequence vector that returns values in the form of 0 and 1, with 1 indicating illegal transactions and 0 indicating legitimate transactions.

4.3 Impact Analysis

The aim of impact analysis is to discover the underlying causes of specific events. It is focused on cutting-edge techniques such as machine learning simulation, Deep Learning technology, and so on. At the moment, the number of credit card use cases is enormous, and credit cards are used for purchases nearly everywhere. And the large number of credit card users creates a large number of risks for both card customers and business owners such as banks and others. The explanation for this is that, when money is transacted online, there is an ongoing party attempting to make it illegal to take money from credit card customers.

When the number of people who use credit cards grows, so do the risks. Hackers or third-party middlemen are often attempting to hack down the machine and discover a bug in the system in order to conduct illegal practices. To discourage this operation, a technology for detecting credit card fraud must be created. To deter this operation, this fraud detection system will recognise the hacker. The credit card fraud detection system can use the user's daily activities, such as time, number, and other facts, to identify the cardholder's real user. If the transaction details is odd, the machine will check to see if the customer is genuine. Otherwise, it would not cause a transaction to be completed. By creating this type of structure, it would be possible to ensure that the transaction will earn the users' interest. Users would be able to keep their account and money away from the hacker while also making a legitimate transaction.

4.3.1 Social and Environmental Impact

When the number of credit card users grows, implementing a mechanism or scheme to deter invalid credit card transactions becomes an increasingly essential activity. Prior to the idea of this, a large amount of money was robbed by a hacker who detected theft in the machine and took the users' money without their knowledge. As a result, credit card users are socially dissatisfied due to the unavailability of their safe. Often, as it occurs on a large scale, hackers steal a large sum of money, which has a significant social effect.

A individual, such as a service provider, who makes an effort to keep his or her money in a bank account. And attempts to borrow capital from the bank in the simplest way possible. As a result, people choose to use credit cards for simple purchases, as they are available on all online business channels. As a result, validating their purchases is a critical problem. Otherwise, the transaction would not get positive feedback from the customer. Through creating a system capable of detecting illegitimate transactions, it would be possible to validate the user's transaction. In addition, customers would have more credit card options available to them.

People who have a bank account nowadays try to use credit cards to transact their money. As a result, developing a method for detecting credit card fraud becomes increasingly necessary. When a user attempts to transact a certain amount of money with this type of identification scheme, the system will attempt to identify the user based on the user's previous knowledge. If they have some problems, the machine will reject the transaction or check to see if the customer is legitimate.

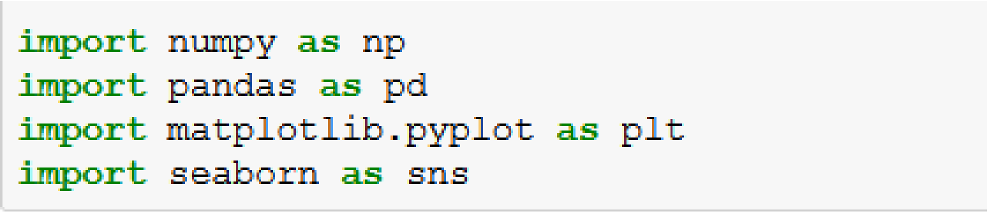
4.3.2 Ethical Impact

The credit card fraud monitoring framework would be able to deter invalid charges from being made from the accounts of the users. Since a bank or other online processing firms accept credit card transactions, the online transaction system must verify the transaction before conducting the process. However, only an online processing mechanism is responsible for the transaction's results, and the system is unaware of the users' validation. Creating a credit card fraud identification mechanism would enable the online transaction system to determine whether or not the user's purchases are legitimate.

Generally, a normal user of a credit card, having some common information from which framework is able to detect the user is valid or not. As the hacker is using the process of detecting fraud in a system is not an ethical way, but a developing system like fraud detection system will provide a way to the companies so that the transactions are getting more and more valid.The ethical method of detecting frauds in the system would ensure the safety of not only consumers but also the business. Aside from that, business or online transaction vendors would be aware of their system's protection. They will also be able to offer the right protection to account holders or customers.

4.4 Evaluation of Credit Card Fraud Detection Framework

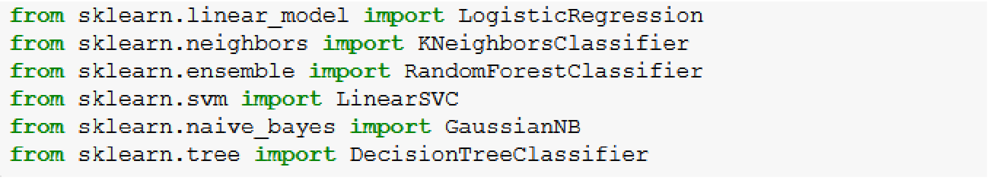
Numpy, Pandas, Matplotlib, and Seaborn are used for routine exploratory data processing and plotting in this thesis work. Numpy is the foundational Python package for scientific computation. Pandas is an open-source data analysis and manipulation framework that is quick, efficient, scalable, and simple to use. It is designed on top of the Python programming language. Matplotlib is a Python library that allows you to create static, animated, and immersive visualizations. Seaborn is a matplotlib-based Python data visualization library.



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Figure 4.1: Exploratory Data Analysis and Plotting Libraries

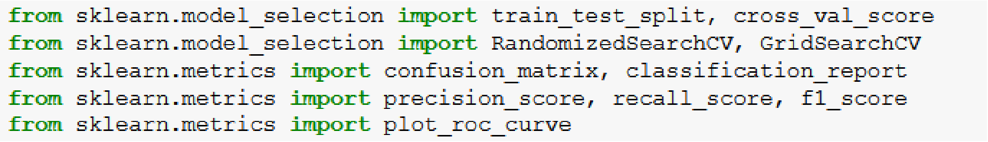
Scikit-learn models such as Logistic Regression, K-Neighbors Classifier, Randomforest Classifier, LinearSVC, GaussianNB, and Decision Tree Classifier are used in this thesis work. Logistic regression is a data processing technique for describing and explaining the relationship between one dependent binary variable and one or more independent nominal, ordinal, interval, or ratio-level variables. KNeighborsClassifier implements classification by voting by the target point's closest k-neighbors, while RadiusNeighborsClassifier implements classification by voting by all neighborhood points within a set radius, r, of the target point, t. Random forest is a versatile, user-friendly machine learning algorithm that delivers excellent results much of the time even without hyper-parameter tuning. Because of its simplicity and variety, it is perhaps one of the most widely used algorithms (it can be used for both classification and regression tasks). A Linear SVC (Support Vector Classifiergoal)'s is to match the data you have by returning a "best fit" hyperplane that separates or categorizes the data. After obtaining the hyperplane, you can then feed some features to your classifier to determine the "predicted" class. Implementation of Gaussian Naive Bayes We built a GaussianNB classifier. To make it simpler to grasp, the decision tree acquires information in the form of a tree, which can also be rewritten as a series of distinct laws.



```
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import LinearSVC
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
```

Figure 4.2: Models from Scikit-Learn

Model assessment is a critical phase in the model development process. For model evaluation in this thesis work, the train test break process, cross val score method, RandomizedSearchCV, Confusion matrix, Classification report, precision score, recall score, f1 score, and plot roc curve are used.



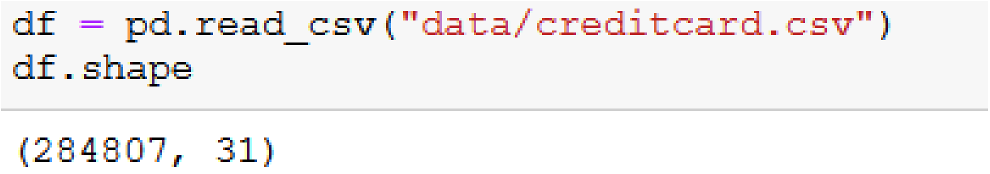
```
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.metrics import precision_score, recall_score, f1_score
from sklearn.metrics import plot_roc_curve
```

Figure 4.3: Model Selection And Evaluation

The very first step in this thesis work is to load the dataset into the data frame. Using the Pandas library, a dataset in CSV format is loaded as a data frame, and the dataset volume is calculated to be 2,84,807. And there are 31 attributes in the dataset.

There are two kinds of data in the dataset. Class 1 denotes legitimate data, while Class 0 denotes fraudulent data. There are 2,84,315 fraud data and 492 valid data from the 2,84,807 data.

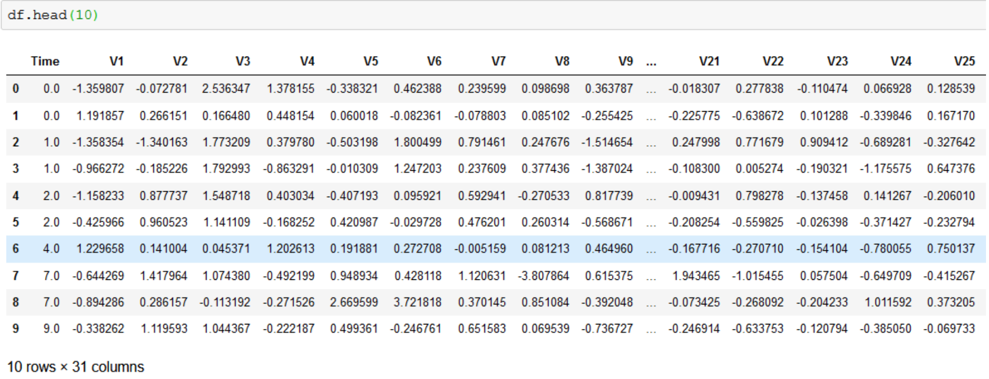
Dataset consists of 31 column, Time, Amount, Class and V1 to V28 in PCA



```
df = pd.read_csv("data/creditcard.csv")
df.shape
```

```
(284807, 31)
```

Figure 4.4: Load Data



```
df.head(10)
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.647376
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.206010
5	2.0	-0.425966	0.960523	1.141109	-0.168252	0.420987	-0.029728	0.476201	0.260314	-0.568671	...	-0.208254	-0.559825	-0.026398	-0.371427	-0.232794
6	4.0	1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.005159	0.081213	0.464960	...	-0.167716	-0.270710	-0.154104	-0.780055	0.750137
7	7.0	-0.644269	1.417964	1.074380	-0.492199	0.948934	0.428118	1.120631	-3.807864	0.615375	...	1.943465	-1.015455	0.057504	-0.649709	-0.415267
8	7.0	-0.894286	0.286157	-0.113192	-0.271526	2.669599	3.721818	0.370145	0.851084	-0.392048	...	-0.073425	-0.268092	-0.204233	1.011592	0.373205
9	9.0	-0.338262	1.119593	1.044367	-0.222187	0.499361	-0.246761	0.651583	0.069539	-0.736727	...	-0.246914	-0.633753	-0.120794	-0.385050	-0.069733

10 rows × 31 columns

Figure 4.5: Data Frame

format. Class data is in int64, and others data are in float64 format. Amount of each data is 2,84,807. Datatypes in float64 is 30 and int64 is 1.

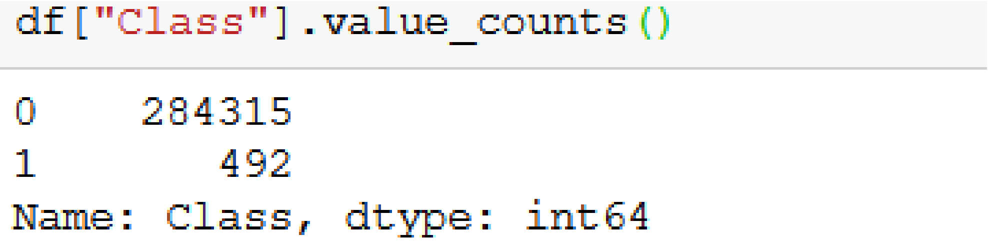
A correlation matrix is nothing more than a table that shows the correlation coefficients for various variables. The matrix illustrates the relationship between all potential pairs of values in a table. It is a valuable method for summarizing a broad dataset as well as identifying and visualizing trends in the data. A correlation matrix is made up of rows and columns that represent the variables. The correlation coefficient is contained in each cell of a table. Furthermore, the correlation matrix is used in conjunction with other mathematical research approaches.

It may be helpful, for example, in the analysis of multiple linear regression models.

Remember that the equations have a lot of independent variables.

Spliting data into input variables X and output varibales y. Output variables y, consists of Class attributes and other attributes are consists of input variables X.

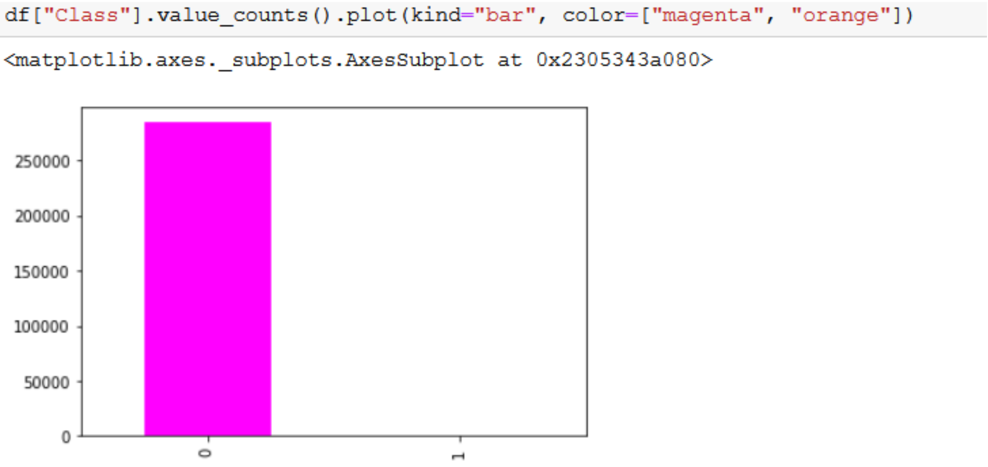
Set Numpy random seet as 42. Split the whole dataset of input variables X and



```
df["Class"].value_counts()
```

```
0    284315
1       492
Name: Class, dtype: int64
```

Figure 4.6: Value Counts for Each Class



```
df["Class"].value_counts().plot(kind="bar", color=["magenta", "orange"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2305343a080>
```

Figure 4.7: Value Counts for Classes

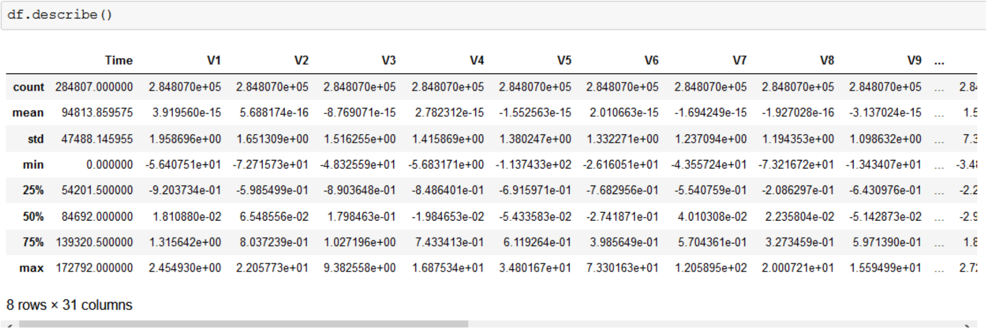
output variables y into X train, X test, y train and y test sets, where test size is assumed as 20 percent of all data.

We considered six separate scikit learn machine learning models in this study work: Logistic Regression, K-Neighbors Classifier, Random Forest Classifier, LinearSVC, GaussianNB, and Decision Tree Classifier. Models can be entered as a sequence into a dictionary called models.

4.5 Evaluation of Performance

To measure the performance of the models, a fit and score approach is built in this thesis work, with parameters including a list of models, X train data, X test data, y train data, and y test data. Set the NumPy random seed to 42 in the fit and score process, and model scores are saved in a list.

The number of genuine transactions in the credit card dataset is 2,84,315 and the number of fraudulent transactions is 492. As a result, the cumulative number of legitimate transactions is very high, and baseline models become overfitted.

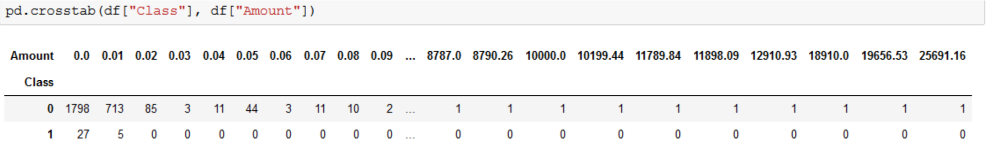


```
df.describe()
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	...	2.8
mean	94813.859575	3.919560e-15	5.688174e-16	-8.769071e-15	2.782312e-15	-1.552563e-15	2.010663e-15	-1.694249e-15	-1.927028e-16	-3.137024e-15	...	1.5
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00	1.415869e+00	1.380247e+00	1.332271e+00	1.237094e+00	1.194353e+00	1.098632e+00	...	7.3
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.683171e+00	-1.137433e+02	-2.616051e+01	-4.355724e+01	-7.321672e+01	-1.343407e+01	...	-3.4
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-6.915971e-01	-7.682956e-01	-5.540759e-01	-2.086297e-01	-6.430976e-01	...	-2.2
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.433583e-02	-2.741871e-01	4.010308e-02	2.235804e-02	-5.142873e-02	...	-2.9
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.119264e-01	3.985649e-01	5.704361e-01	3.273459e-01	5.971390e-01	...	1.8
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.480167e+01	7.330163e+01	1.205895e+02	2.000721e+01	1.559499e+01	...	2.7

8 rows × 31 columns

Figure 4.8: Data Info



```
pd.crosstab(df["Class"], df["Amount"])
```

Amount	0.0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	...	8787.0	8790.26	10000.0	10199.44	11789.84	11898.09	12910.93	18910.0	19656.53	25691.16
Class																					
0	1798	713	85	3	11	44	3	11	10	2	...	1	1	1	1	1	1	1	1	1	1
1	27	5	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

Figure 4.9: Comparison Class Column with Amount Column

To address this problem, we look at Hyperparameter Tuning with RandomizedSearchCv. Random Forest Classifier outperforms all baseline models in terms of precision, and for this purpose, hyperparameter tuning with randomizedsearchcv is considered for Random Forest Classifier.

For hyperparameter tuning with randomizedsearchcv, create a hyperparameter grid for Random Forest Classifier.

For tuning the Random Forest Classifier, set the numpy random seed as 42. Setup hyperparameter search for Random Forest Classifier using RandomizedSearchCV with the parameter of RandomForestClassifier method, param distribution as rendor forest grid, cv as 5, number of iter is 20 and verbose as True.

After setting up the random hyperparameter search model for RandomForestClassifier, fit the model with the training data. After fit the model, best params of the fit model of random hyperparameter search model for RandomForestClassifier is evaluated and the evaluated value for fit model is, number of estimators is 510, min samples split is 14, min samples leaf is 1, and max depth is None.

A receiver operating characteristic curve (ROC curve) is a graph that depicts the contribution of a classification model over all classification thresholds.

Two parameters are depicted in this graph:

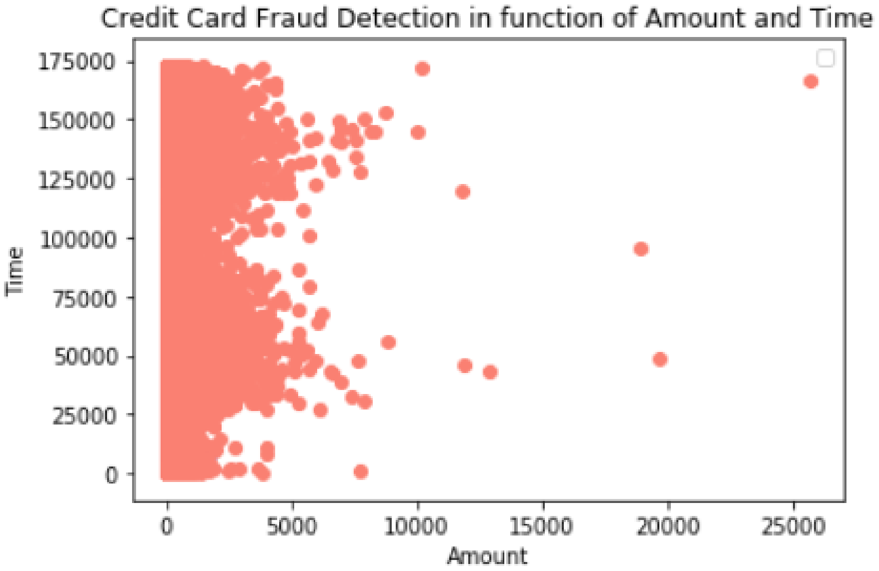


Figure 4.10: Credit Card Fraud Detection in Function of Amount And Time

True Positives Level.

A table that is often used to represent the results of a classification model (or "classifier") on a set of test data whose true values are known is known as an uncertainty matrix.

The complexity matrix itself is simple to understand, but the terms synonymous with it can be confusing.

A classification report is used to evaluate the predictive performance of a classification algorithm.

How many predictions are right and how many are wrong.

As seen below, True Positives, False Positives, True Negatives, and False Negatives are used to predict the metrics of a classification report.

Cross validation is a machine learning algorithm evaluation technique that entails training the model on a subset of the available data and then testing it on the remaining input data.

Simply put, we set aside a portion of the data and then train the model on the rest.

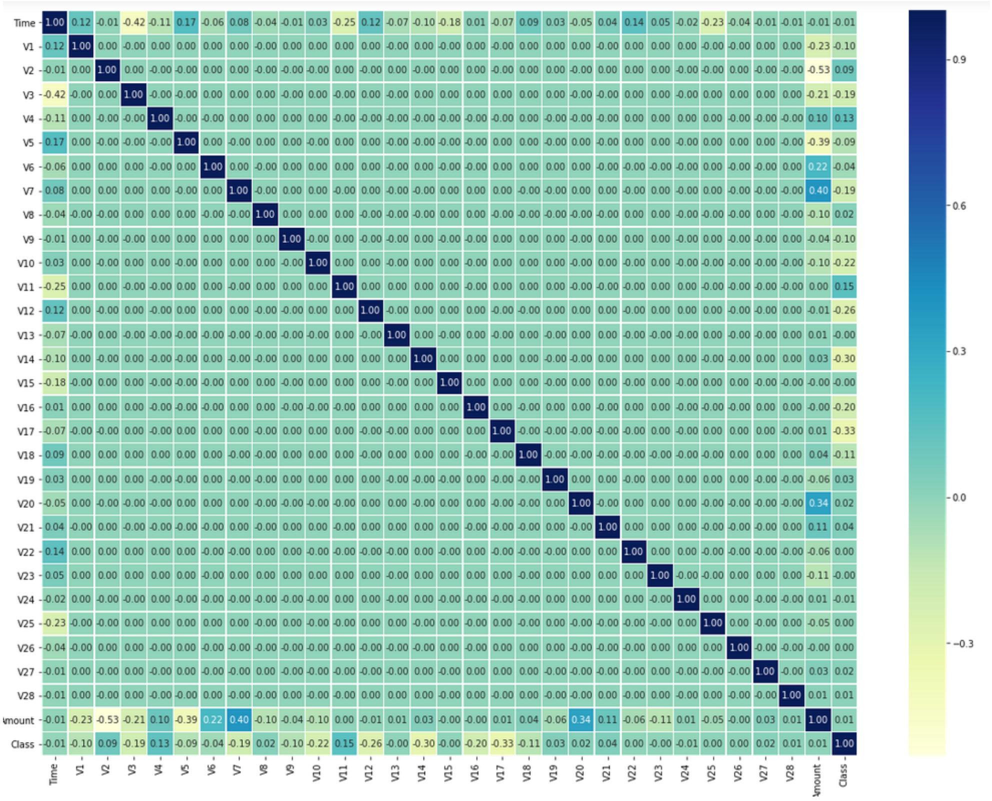
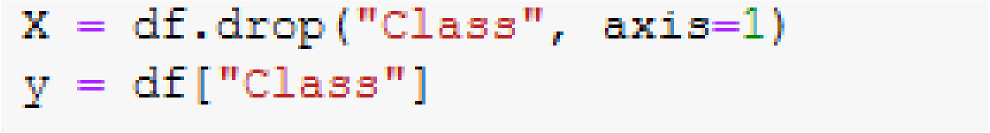


Figure 4.11: Correlation Matrix



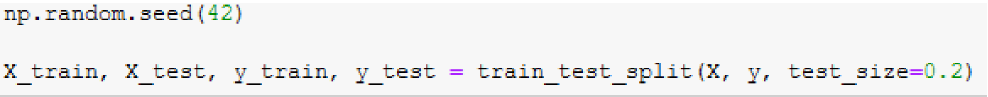
```
X = df.drop("Class", axis=1)
y = df["Class"]
```

Figure 4.12: Splitting Data into X and y

4.6 Conclusion

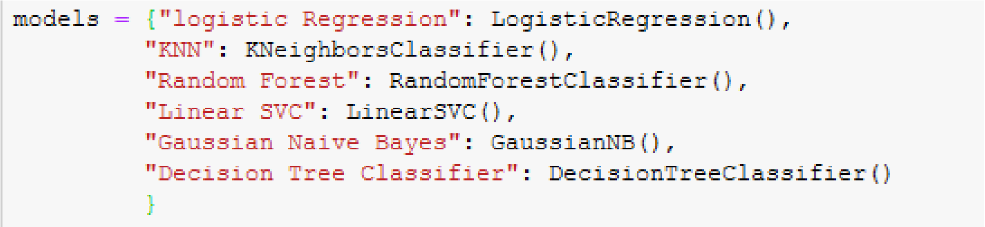
This chapter presents the findings of the Credit Card Fraud Detection Framework.

The proposed framework's performance is also discussed here. The suggested random hyperparameter tuning approach offers greater precision, as shown by the results. The conclusion to this research work is drawn in the following sections.



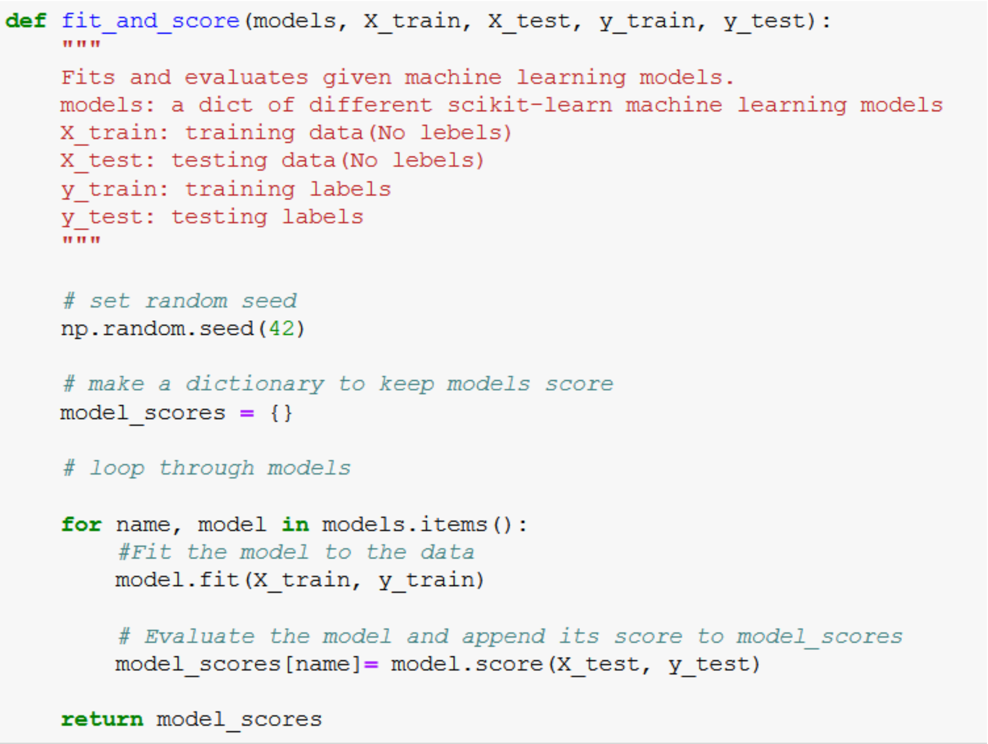
```
np.random.seed(42)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

Figure 4.13: Split Data into Train And Test Sets



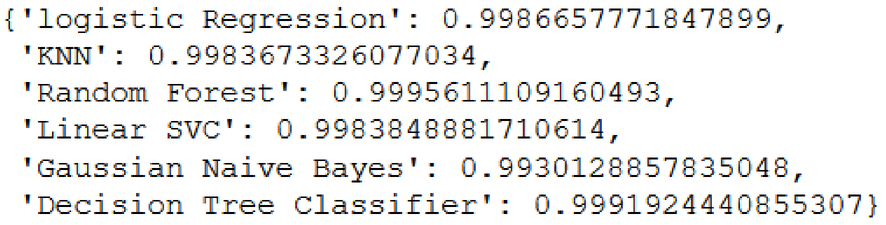
```
models = {"logistic Regression": LogisticRegression(),
          "KNN": KNeighborsClassifier(),
          "Random Forest": RandomForestClassifier(),
          "Linear SVC": LinearSVC(),
          "Gaussian Naive Bayes": GaussianNB(),
          "Decision Tree Classifier": DecisionTreeClassifier()
          }
```

Figure 4.14: List of Models



```
def fit_and_score(models, X_train, X_test, y_train, y_test):
    """
    Fits and evaluates given machine learning models.
    models: a dict of different scikit-learn machine learning models
    X_train: training data(No lebels)
    X_test: testing data(No lebels)
    y_train: training labels
    y_test: testing labels
    """

    # set random seed
    np.random.seed(42)

    # make a dictionary to keep models score
    model_scores = {}

    # loop through models

    for name, model in models.items():
        #Fit the model to the data
        model.fit(X_train, y_train)

        # Evaluate the model and append its score to model_scores
        model_scores[name]= model.score(X_test, y_test)

    return model_scores
```

Figure 4.15: Function to Fit And Score Models



```
{'logistic Regression': 0.9986657771847899,
 'KNN': 0.9983673326077034,
 'Random Forest': 0.9995611109160493,
 'Linear SVC': 0.998384881710614,
 'Gaussian Naive Bayes': 0.9930128857835048,
 'Decision Tree Classifier': 0.9991924440855307}
```

Figure 4.16: Baseline Models Scores

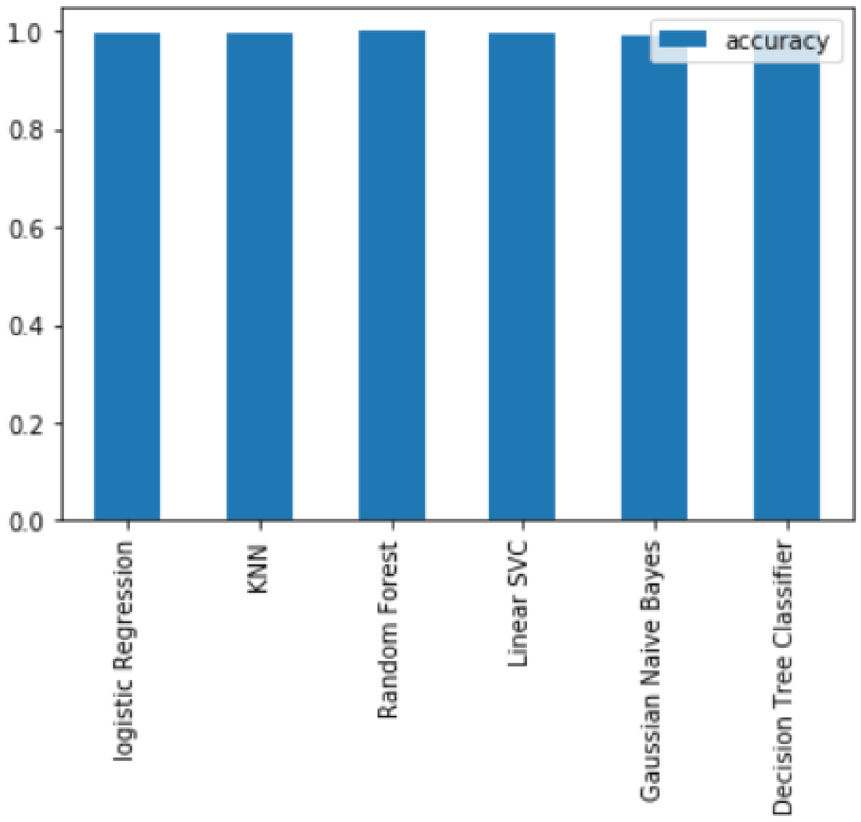
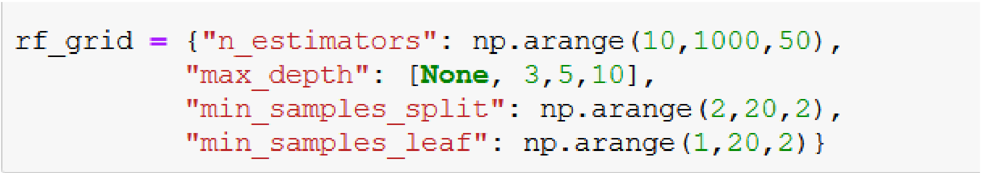
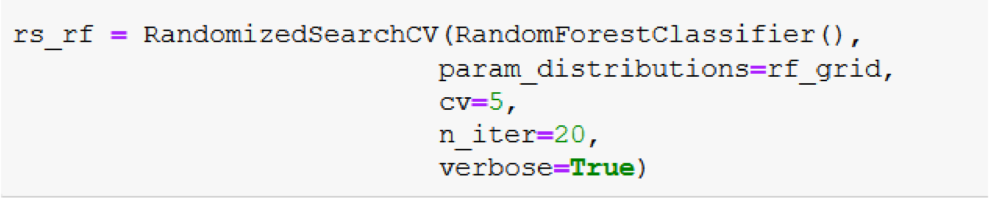


Figure 4.17: Model Comparisons



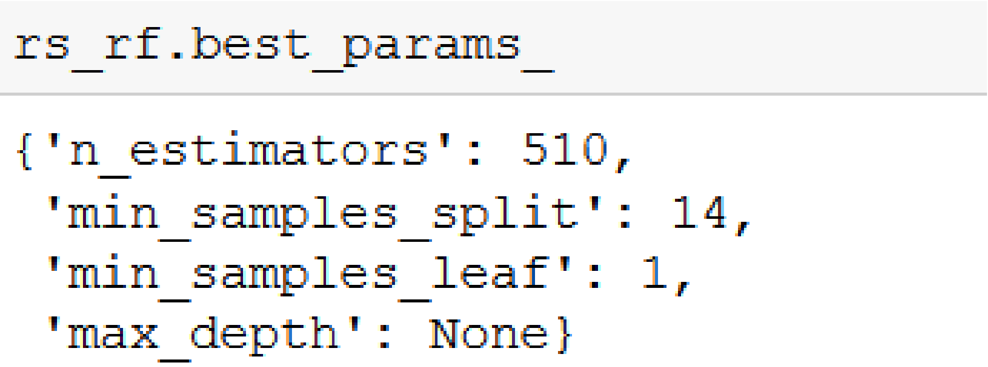
```
rf_grid = {"n_estimators": np.arange(10,1000,50),
           "max_depth": [None, 3,5,10],
           "min_samples_split": np.arange(2,20,2),
           "min_samples_leaf": np.arange(1,20,2)}
```

Figure 4.18: Hyperparameter Grid for Random Forest Classifier



```
rs_rf = RandomizedSearchCV(RandomForestClassifier(),
                           param_distributions=rf_grid,
                           cv=5,
                           n_iter=20,
                           verbose=True)
```

Figure 4.19: Random Hyperparameter Search for RandomForestClassifier



```
rs_rf.best_params_
```

```
{'n_estimators': 510,
 'min_samples_split': 14,
 'min_samples_leaf': 1,
 'max_depth': None}
```

Figure 4.20: Best Params for Random Hyperparameter Search Model for RandomForestClassifier

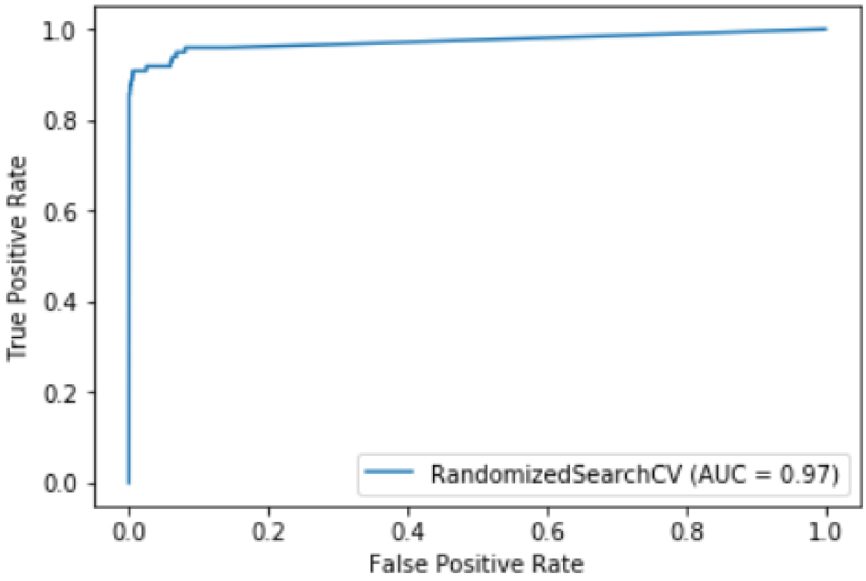


Figure 4.21: ROC Curve

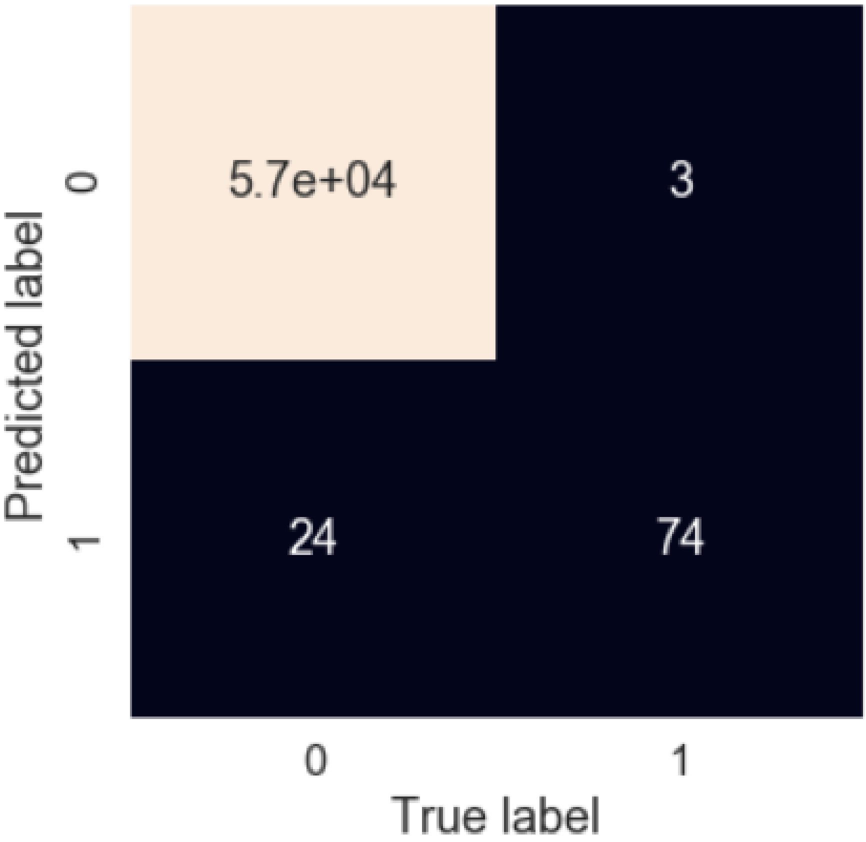
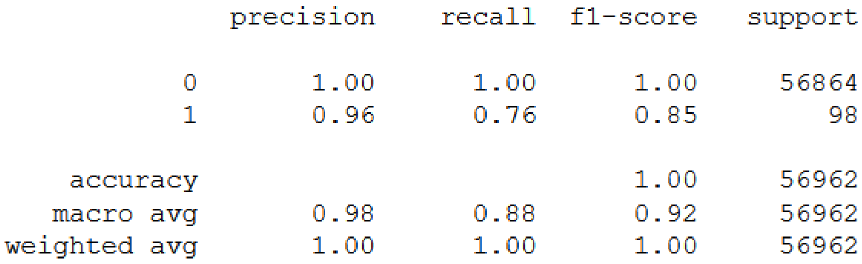


Figure 4.22: Confusion Matrix



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56864
           1       0.96      0.76      0.85        98

    accuracy                           1.00     56962
   macro avg       0.98      0.88      0.92     56962
weighted avg       1.00      1.00      1.00     56962
```

Figure 4.23: Classification Report

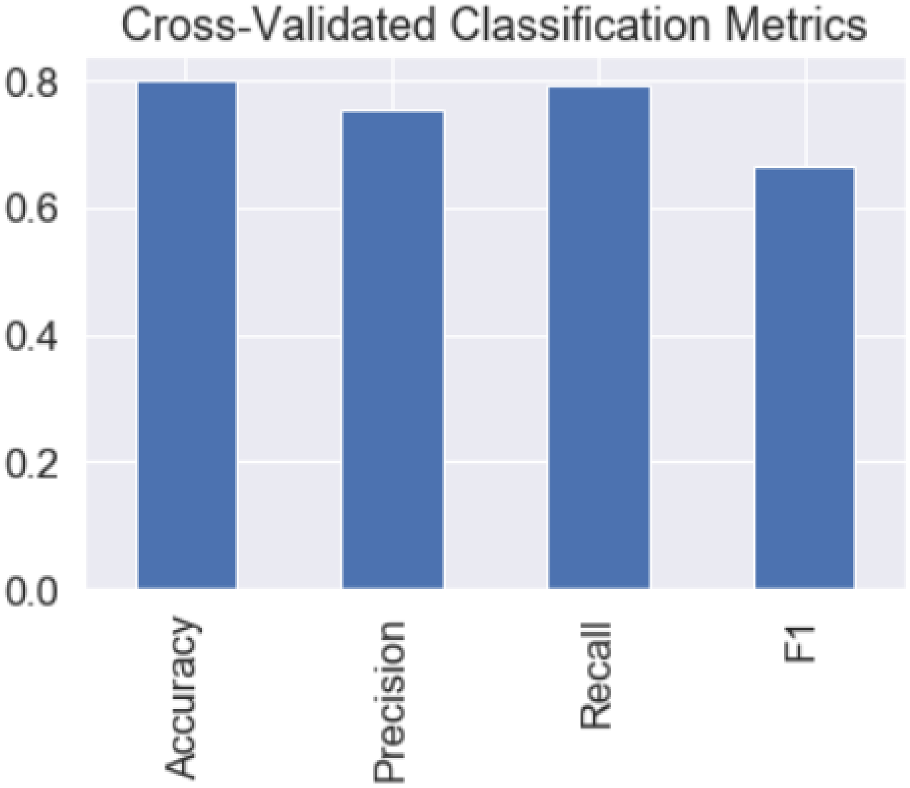


Figure 4.24: Cross-Validated Classifications Metrics

Chapter 5

Conclusion

5.1 Conclusion

As long as detecting credit card fraud is a work in progress, there is still room for change. The credit card dataset is used in this thesis work. Since user purchases contain user information, it is extremely difficult to use raw user data. PCA transformation of user input, on the other hand, is used for model development and ranking.

We used 2,84,807 transaction data points from Kaggle in our project. In our dataset, there are 2,84,315 legitimate transactions and 492 fraudulent transactions. The percentage of fraud data over a significant number of legitimate transactions is 0.172 percent. The dataset contains 31 properties, the last of which is Class value, which means 1 or 0. A legitimate transaction has a value of 1, whereas a fraudulent transaction has a value of 0.

Divide the total data into train data and test data, with a test scale of 20Sckikit learn models are educated using 80 percent of all results. In our study, we looked at six different model training algorithms. Logistic Regression, K-Neighbors Classifier, Random Forest Classifier, LinearSVC, GaussianNB, and Decision Tree Classifier are among them. After training all models, rate them using the testing results, which represents the remaining 20 percent of all data.

Based on the 2,84,807 transactions, baseline models estimate the score. Random Forest Classifier does comparatively well than the other baseline models. So, in our project work, we tuned our Random Forest Classifier Model, and for this, we used randomizedsearchcv to set up a random hyperparameter tuning model.

We trained our new hyperparameter tuning model for RandomForestClassifier once more, and this time we evaluated the model's score, which was higher than before.

5.2 Future Work

As long as detecting credit card fraud is a work in progress, there is still room for change. The dataset from Kaggle is used in this thesis work. A number of data attributes are fixed in the Kaggle dataset, and models are trained using this information. However, the features of consumer transaction records are changing on a daily basis. As a result, developing the model based on consumer transaction data necessitates a continuous operation.

This thesis work does not make use of any of the user's material. Collecting more and more characteristics from consumer purchases would have greater precision in detecting fraud. Aside from that, the amount of fraud data in the Kaggle dataset is very limited, which is a major issue for model testing. Models are constructed using a vast volume of legitimate data, so models do well for this particular forecast, but increasing the number of fraud data would yield more reliable results. By resolving these constraints, a solid architecture for detecting Credit Card Fraud can be developed.

References

[1] Y. Kou, C.-T. Lu, S. Sirwongwattana and Y.-P. Huang, ‘Survey of fraud detection techniques,’ in *IEEE International Conference on Networking, Sensing and Control, 2004*, IEEE, vol. 2, 2004, pp. 749–754 (cit. on p. iv).

[2] Y. Jain, S. NamrataTiwari and S. Jain, ‘A comparative analysis of various credit card fraud detection techniques,’ *International Journal of Recent Technology and Engineering*, vol. 7, no. 5, pp. 402–407, 2019 (cit. on p. iv).

[3] F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé and G. Bontempi, ‘Combining unsupervised and supervised learning in credit card fraud detection,’ *Information Sciences*, 2019 (cit. on p. 1).

[4] X. Niu, L. Wang and X. Yang, ‘A comparison study of credit card fraud detection: Supervised versus unsupervised,’ *arXiv preprint arXiv:1904.10604*, 2019 (cit. on p. 6).

[5] S. B. E. Raj and A. A. Portia, ‘Analysis on credit card fraud detection methods,’ in *2011 International Conference on Computer, Communication and Electrical Technology (ICCCET)*, IEEE, 2011, pp. 152–156 (cit. on p. 6).

[6] A. Shen, R. Tong and Y. Deng, ‘Application of classification models on credit card fraud detection,’ in *2007 International conference on service systems and service management*, IEEE, 2007, pp. 1–4 (cit. on p. 7).

[7] K. Chaudhary, J. Yadav and B. Mallick, ‘A review of fraud detection techniques: Credit card,’ *International Journal of Computer Applications*, vol. 45, no. 1, pp. 39–44, 2012 (cit. on p. 7).

[8] A. C. Bahnsen, D. Aouada, A. Stojanovic and B. Ottersten, ‘Feature engineering strategies for credit card fraud detection,’ *Expert Systems with Applications*, vol. 51, pp. 134–142, 2016 (cit. on pp. 7, 8).

[9] L. Delamaire, H. Abdou and J. Pointon, ‘Credit card fraud and detection techniques: A review,’ *Banks and Bank systems*, vol. 4, no. 2, pp. 57–68, 2009 (cit. on p. 7).

[10] P. Save, P. Tiwarekar, K. N. Jain and N. Mahyavanshi, ‘A novel idea for credit card fraud detection using decision tree,’ *International Journal of Computer Applications*, vol. 161, no. 13, 2017 (cit. on p. 7).

[11] J. O. Awoyemi, A. O. Adetunmbi and S. A. Oluwadare, ‘Credit card fraud detection using machine learning techniques: A comparative analysis,’ in *2017 International Conference on Computing Networking and Informatics (ICCNI)*, IEEE, 2017, pp. 1–9 (cit. on p. 7).

[12] L. Delamaire, H. Abdou and J. Pointon, ‘Credit card fraud and detection techniques: A review,’ *Banks and Bank systems*, vol. 4, no. 2, pp. 57–68, 2009 (cit. on pp. 7, 8).

[13] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang and C. Jiang, ‘Random forest for credit card fraud detection,’ in *2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)*, IEEE, 2018, pp. 1–6 (cit. on p. 7).

[14] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi and G. Bontempi, ‘Credit card fraud detection: A realistic modeling and a novel learning strategy,’ *IEEE transactions on neural networks and learning systems*, vol. 29, no. 8, pp. 3784–3797, 2017 (cit. on p. 8).

[15] R. Patidar, L. Sharma *et al.*, ‘Credit card fraud detection using neural network,’ *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 1, no. 32-38, 2011 (cit. on p. 8).

[16] Y. G. Şahin and E. Duman, ‘Detecting credit card fraud by decision trees and support vector machines,’ 2011 (cit. on p. 8).

[17] A. G. de Sá, A. C. Pereira and G. L. Pappa, ‘A customized classification algorithm for credit card fraud detection,’ *Engineering Applications of Artificial Intelligence*, vol. 72, pp. 21–29, 2018 (cit. on p. 8).

[18] J. T. Quah and M. Sriganesh, ‘Real-time credit card fraud detection using computational intelligence,’ *Expert systems with applications*, vol. 35, no. 4, pp. 1721–1732, 2008 (cit. on p. 9).

[19] D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic and A. Anderla, ‘Credit card fraud detection-machine learning methods,’ in *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, IEEE, 2019, pp. 1–5 (cit. on p. 9).

[20] Z. Zojaji, R. E. Atani, A. H. Monadjemi *et al.*, ‘A survey of credit card fraud detection techniques: Data and technique oriented perspective,’ *arXiv preprint arXiv:1611.06439*, 2016 (cit. on p. 10).