University of Essex
Department of Mathematical Sciences

MA981: DISSERTATION

Enhancing Credit Card Fraud Detection: Using Data-Driven Approaches

Shashank Rawat
2204433

Supervisor: Dr. Zoe Bartlett

August 25, 2023
Colchester

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Abstract

Credit card fraud presents a significant challenge in digital payment systems, leading to significant financial losses for both financial institutions and customers. With an increased global push for digital payment, the rise in fraudulent activities in the sector is also expected. Credit cards frauds are constantly changing and evolving,and ever adapting fraudsters, using advanced data-driven technology can offer a perfect solution to this problem by learning patterns in fraudulent transactions and stopping them at the source.

Good quality and quantity of data is key for successful data-driven techniques, and getting accurate and relevant data can be difficult due to privacy concerns and imbalances in the real-life dataset, where the majority of the transactions are non-fraudulent.

The study utilises multiple machine learning and neural network algorithms on european credit card transaction data, which is highly imbalanced, with only 0.172% of the transaction being fraud. The aim of the study is not only to detect the correct fraudulent transaction in an upcoming transaction but also to prevent it from classifying a genuine transaction as fraud, which can lead to a bad consumer experience. In order to achieve this, all algorithms are compared across multiple evaluation metrics and the best algorithm with the best balance score is selected. Resampling methods were utilised to mitigate the effect of the imbalance in the data.

After extensive experimentation and analysis, the study selects a random forest classifier, an ensemble method which combines multiple decision tree classifiers to predict a class, combined with random oversampling method as most effective classifier for predicting fraudulent transactions and avoiding genuine transactions being classified as fraud. The use of selected data-driven method coupled with a rule-based system can enhance the performance a credit card fraud detection system.

1.2 Background

Digital payments have become a key pillar of our Financial Ecosystem, transforming how we use and interact with money. Digital payments refer to the process of conducting electronic transactions involving making payments for goods and services over the internet or through digital platforms like point of sale (POS) systems and other mediums. It has made our life as a consumer easy as this method eliminates the need for physical cash and traditional payment methods such as checks and bank drafts. According to the analysis by Statista[1], a top provider of market and consumer data statistics, it is projected that the total market value for digital payments/Transactions will be worth US $ 9.46 trillion by 2023, and it is anticipated to grow at a rate of 11.80% annually, reaching US $ 14.78 trillion by 2027, (Statista,2023)[1]. This indicates that digital payments are not only the present but also the future of the global payment system.

Many consumers rely on credit cards as a financial tool for making purchases on credit. It enables users to borrow money from the issuer and repay it in instalments or in full. Bank of America introduced The first credit card in 1958 in the form of BankAmericard in Fresno, California, starting with 60000 cards with $300 credit for the city residents. Compared to the current global picture, there are 2.8 billion credit card users worldwide(Money.co.uk,2023)[4]. Credit Card payments have become a predominant method in digital payments. It plays a significant role in the overall payment environment by providing a convenient, globally accepted and secure method for consumers to make online and electronic transactions. Ac-

cording to a report by UK Finance, in the year 2021, 57% of all payments made in the UK were conducted using cards, while over 66% of adults in the UK possess credit cards (UK Payment Markets Summary 2022)[2].

Given the vast number and volume of daily transactions, payment systems are consistently targeted by fraudulent activities, including card fraud, identity theft, account takeovers, phishing, and various other techniques. These fraudulent acts pose challenges for both consumers and financial institutions, as they can lead to financial losses and compromised personal data. Therefore, it is crucial to detect fraudulent activities by financial institutions to prevent financial losses, protect data, improve user experience, prevent financial crimes, and promote the growth of the digital payment ecosystem.

1.3 Research Problem

Credit card fraud involves the unauthorized use of someone else's credit card or credit card details to make transactions without the knowledge or consent of the cardholder. Such fraudulent actions can encompass purchasing goods or services or transferring funds to another account. As we become more reliant on credit cards, instances of its fraud have increased in recent years, especially with the advancement of technology.

When we provide our card details to service providers for online transactions, this sensitive information is stored in databases, making it susceptible to leaks if not handled carefully. Fraudsters can obtain card information through these data breaches and exploit the stolen data to carry out unauthorized transactions. These fraudulent activities will be responsible for a remarkable $408.50 billion in global losses for the financial industry in the upcoming Decade (Nilson Report)[3].

Financial institutions employ various methods to thwart these fraudulent activities, like rule-based transaction monitoring systems, real-time alerts, CVV verification, two-factor authentication, geolocation techniques, and machine learning classifications. Although these

measures are beneficial, they may come with certain shortcomings, such as cost and resource requirements, delayed detection, impacts on user experience, and the occurrence of false positive cases, which can result in potential business losses. Researchers are continuously trying to develop new technology to mitigate these issues. Moreover, fraudsters continuously adapt and evolve their techniques to evade detection over time. Hence, more research for a robust fraud detection system is essential to identify potential fraudulent credit card transactions in real time, safeguarding consumers and financial institutions from financial losses. Such a system would ensure a secure and trustworthy digital payment ecosystem.

1.4 Objectives

The main objective of this study is to utilise these data-driven techniques to build an adaptive fraud detection system that achieves high detection accuracy in credit card fraud. In order to attain this goal, the following objectives have been taken into account:

1. Examining existing systems and research in place for credit card fraud detection, gaining knowledge from them and evaluating their limitations.
2. Analysing how data quality and quantity impact the system's performance and exploring strategies(data preprocessing, features Selection) to overcome these challenges.
3. Implementing multiple supervised machine learning Classification algorithms, comparing their results from various aspects and choosing the best algorithm that detects fraudulent transactions effectively.
4. Applying Neural network algorithms to the data to determine whether they can improve fraud detection accuracy compared to traditional machine learning approaches.
5. Exploring multiple sampling techniques to balance the data and its effect on the efficiency of a robust data-driven fraud detection system.
6. Examining the limits and requirements necessary to improve a proposed fraud detection system.

CHAPTER 2

Literature Review

In this section, we learn about credit card fraud and the fraud detection system. The main focus of this section is to achieve the first objective of the study by learning about the existing systems, research in the field and their limitations in detecting credit card fraud.

2.1 Overview of Credit Card fraud

There are two main categories of credit card fraud: application fraud and behavioural fraud [26, 27]. Application fraud occurs when someone obtains a credit card using false information and quickly reaches the credit limit. On the other hand, behavioural fraud is when a person obtains legitimate card details through fraudulent means and then engages in sales without the cardholder being physically present, such as in online or telephone transactions.

Behavioural fraud can be broken down into three types: stolen or lost card, counterfeit card, and card not present (CNP). Stolen or lost card fraud occurs when someone steals or the user loses the card, which is used to commit fraudulent transactions. This type of fraud can be prevented through user awareness and quick response. Counterfeit credit card fraud is committed by a perpetrator who creates a fake card using the victim's card details. The most frequent type of credit card fraud that accounts for approximately 75% of all card frauds in the world[5] is card not present(CNP), where no physical card is needed, and

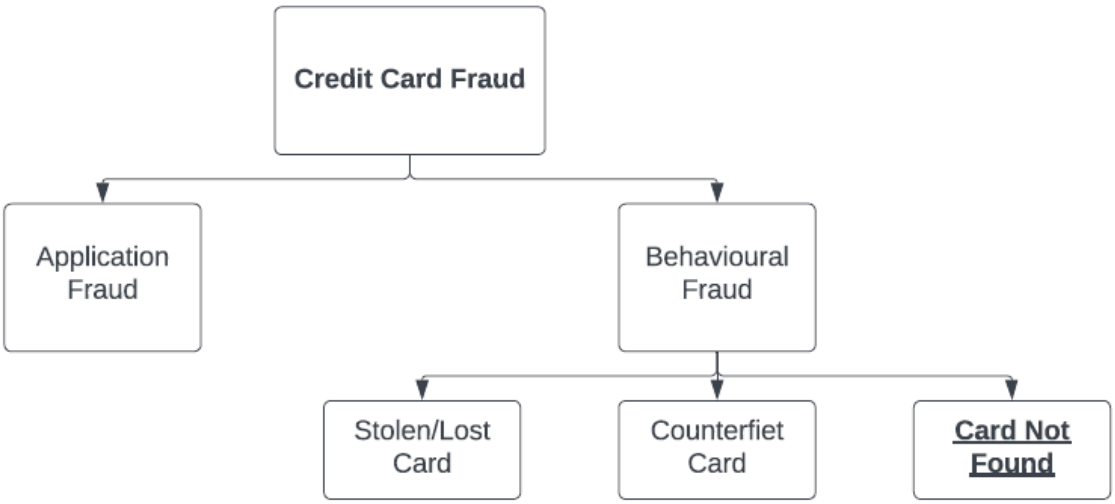

Figure 2.1: Types of Credit Card Fraud

fraudulent transactions are made online using the victim's card details. These types of fraud are particularly alarming due to their anonymous and lightning-fast nature of transactions. Moreover, their reach is virtually limitless, adding to the potential damage they can cause[**?**].

2.2 Fraud Detection System

The actions taken against fraud are of two main types. Firstly, Preventive actions aim to proactively minimise fraud by implementing various measures and strategies at the source [25]. Secondly, Reactive actions, also known as Fraud Detection, are taken in response to identified or suspected fraudulent activities.

A credit card fraud detection system is designed to determine whether a new credit card transaction is legitimate or fraudulent [25]. Its main goal is to quickly and precisely identify fraud cases, which then alerts the financial system to review high-risk transactions promptly. It relies on analyzing past transaction data, including various transaction details features (transaction amount, geolocation, frequency, user demographic information, etc.)

Two primary methods used in fraud detection systems are rule-based and data-driven.

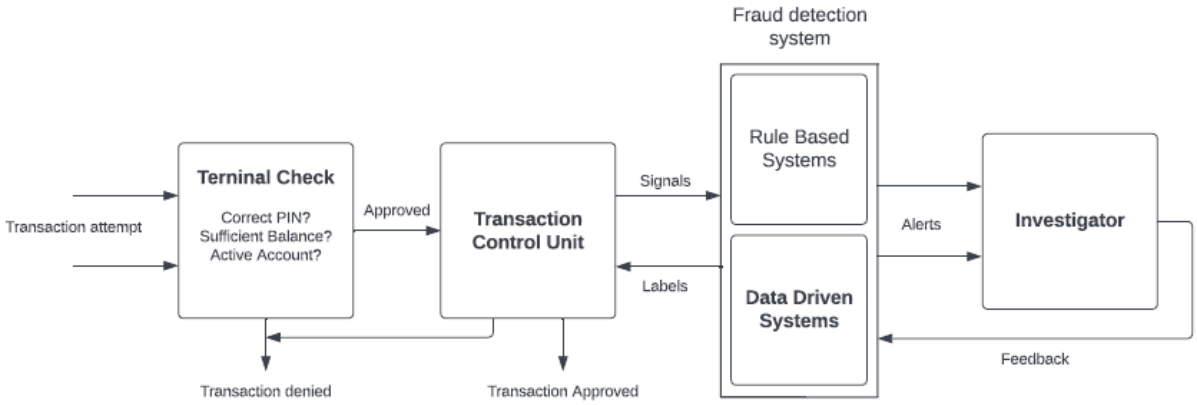

Figure 2.2: An Illustration of Real-time Fraud detection system. this study focus on the Data driven Systems in FDS.

Rule-based detection, also called expert-based detection, involves identifying abnormal activities based on predetermined rules or criteria set by specialists in the field. The drawbacks of this technique are its rigidity, Its ability to generate high false positives and negatives and its lack of scalability as it relies on human supervision to update rules.

The second and most advanced and researched approach is data-driven, where the fraud detection system employs both Machine Learning algorithms (such as Logistic Regression, Random Forest, SVM, etc.) and Deep Learning methods (including LSTM, CNN, RNN) to learn from historical transaction data. By doing so, the system derives the patterns associated with fraudulent transactions [25]. Although this method lacks interpretability, its main advantage lies in its capacity to find patterns from a Large volume of data.

Fraudsters continuously adjust to emerging technologies and devise fresh methods to evade security systems. Therefore, it is crucial to update the fraud detection system constantly, and to thwart this the financial institutions invest significant amounts of money each year to develop and update their FDS. The progress in creating new fraud detection systems is highly challenging due to the limited exchange of ideas on this topic. This is because the availability of the elaborative fraud detection system in the public domain would provide fraudsters with the necessary resources to come up with ways to outsmart these systems. Researchers

are continually working to overcome the associated challenges. data-driven techniques like machine learning and artificial intelligence have enabled researchers to implement robust fraud detection systems. It enables us to identify patterns, anomalies, and trends that may indicate fraudulent activities from the large volume of historical transaction data that human analysts cannot.

2.3 Related work

Researchers are harnessing data-driven technologies to advance credit card fraud detection and prevention, employing innovative approaches that leverage vast datasets, sophisticated algorithms, and machine learning techniques and deep learning techniques. These advancements are transforming the fraud detection landscape by enhancing accuracy, reducing false positives, and adapting to evolving fraudulent tactics. In recent times, extensive exploration of machine learning and deep learning for identifying credit card fraud has resulted in the creation of various approaches, and these approaches include supervised, unsupervised, and semi-supervised techniques.

2.3.1 Supervised Learning

Supervised learning is a method in which the algorithm learns from well labelled data. It means that in each input transaction data, the output is labelled as either fraudulent or genuine [6]. The model then uses this data to identify patterns and relationships to make accurate predictions on whether new, unseen transactions are fraudulent or not.

In a 2020 study[7] , the author used several supervised machine learning classification algorithms (Decision Tree, kNN, Logistic Regression and Random forest, Naive Bayes) to build a credit card fraud classification model on the imbalanced dataset and present the comparison of their performance with different matrices sensitivity, precision and time. They

concluded that although the better sensitivity score of KNN, the best algorithm is chosen to be was Decision tree, kNN being a non-parametric algorithm, took a long time to test the data, the author also recommended implementing additional scoring measures and utilizing multiple datasets to enhance the effectiveness of the proposed fraud detection system. In a recent 2019 study by Heta Naik et al. [8], various supervised classification algorithms were researched and analyzed. Naive Bayesian, Logistic Regression(LR), Adaboost, and J48 were among the selected algorithms, with J48 being a widely used algorithm in the diagnosis of coronary heart disease and classifying e-governance data, it produces decision trees based on information theory [9]. Adaboost and LR were found to have the highest accuracy, with Adaboost having a processing time approximately one-fourth of that of Logistic Regression. However, J48 took around twenty times the time taken by Adaboost due to its production and processing of multiple decision trees. While the study did not use evaluation matrices minimally and did not handle the data's imbalance, it still provided valuable insights into the performance of different algorithms. Yashvi Jain et al. conducted a study in 2019 [10] that evaluated various techniques, including Artificial Neural Network(ANN), Support Vector Machine, Naive Bayes, kNN, Hidden Markov Model, and Fuzzy Logic-based model. The study compared these techniques using evaluation matrices such as accuracy, precision, false alarm rate, and cost. However, the study found that these techniques are not guaranteed to produce the same results in all environments. While ANN is highly accurate, it is expensive to train. KNN and SVM have excellent results with small data sets but are unsuitable to a large datasets. Furthermore, some techniques, like decision tree and support vector machine, perform better on pre-processed data, while other algorithms, like logistic regression and fuzzy systems, give better accuracy on unsampled data. To address these issues, the authors suggested creating a hybrid model that pairs a technique that takes a long time to train but provides accurate results with optimization to reduce the training time computational cost.

Ensemble learning can be used to improve the machine learning models, in this process multiple weak models are created to predict an outcome, either by using many different modelling algorithms or using different training data sets. The ensemble model then combines each base model's prediction and results in the final prediction for the unseen input data [11]. In the paper [12] author used the mix of adaboost ensemble technique that learns

from mistakes by increasing the weight of misclassified data points [14]. and long short term memory(LSTM) a special type of recurrent neural network(RNN) and Based on their findings, it was concluded that this model outperformed the traditional classification models in every evaluation metric. The author discussed their approach's limitations and potential drawbacks, which are its need for large amounts of data and the potential for overfitting. The paper also highlights the necessity for further investigation into sampling and feature engineering methods. Joy Iong-Zong Chen, in his study [14], suggests a supervised deep convolutional neural network approach to classify fraudulent activity. The author found that this approach is better than traditional fraud detection systems because of effectiveness in handle large amounts of data and its speed and accuracy in detecting fraud.

In a supervised classification environment, class imbalance presents a significant challenge. This issue is particularly prevalent in credit card fraud detection environments where the number of fraud cases is typically less than 1% among the historical transaction data, resulting in an algorithm to classify the majority class correctly while ignoring the minority class, undermining the aim of building a fraud detection system.

In a 2019 study, S. Makki et al. [15] performed an analysis in which the author tested eight classification models and selected the top three based on accuracy, sensitivity, and area under the precision-recall curve. The author also tried three different sampling techniques on the dataset: Random Oversampling, One-Class Classification, and Cost Sensitive. After comparing the performance of the algorithms, the study concluded that the chosen algorithms worked better in imbalanced data. However, the author suggested using more sampling techniques, such as SMOOTE in future works.

2.3.2 Unsupervised Learning

Unsupervised learning, also known as outlier detection or anomaly detection, is a data mining process for credit card fraud detection systems [16] in this approach, the model is trained on an unlabeled dataset to identify unusual patterns and anomalies and group transactions

similar data points into respective classes/clusters. It is widely used in FDS systems for its ability to work well even in imbalanced data.

In a study conducted by Hariteja Bodepudi in 2021 [17], three unsupervised algorithms, namely Isolation Forest, Local Outlier Factor, and one class SVM were used for anomaly detection. The author argued that unsupervised learning is better than traditional data mining techniques because labelled data is not readily available, and constant model training is required to keep it up-to-date. The study found that Isolation Forest emerged as the superior algorithm compared to the other methods. In 2020 A. K. Rai et al. [18] proposed a novel technique for detecting fraud in credit card data using a Neural Network (NN) based unsupervised learning technique study compared the models such as Auto Encoder, Isolation Forest, Local Outlier Factor, and K Means clustering on a credit card dataset. The author also suggests the need for a large pool of credit card data to make the model more robust. In [19], S. Jiang et al. propose a new credit card fraud detection framework based on an unsupervised attentional anomaly detection network (UAAD-FDNet) which uses the generator to create a real transaction data sample and a discrimnator which uses a hybrid weighted loss function to encode the transaction data, and this approach outperforms traditional ML and Dl techniques used in the realm of FDS.

2.3.3 Semi-supervised Learning

Semi-supervised approach utilises both supervised and unsupervised methods to build a robust model. It is first trained on labelled data to construct a classifier, which is subsequently applied to assign labels to unlabeled data.

In the research paper [6], the author proposes a hybrid technique combining supervised and unsupervised methods to enhance fraud detection accuracy. The author recommends using supervised techniques to learn from previous fraudulent activities and unsupervised approaches to detect new types of fraud for optimal results. In another study[20], The researchers created a technique that combines Laplacian RLS and Laplacian SVM with manifold regularisation to enhance the precision of identifying fraudulent transactions in

a semi-supervised model. The study also repeated the challenge model faced due to data imbalance and emphasised the importance of shuffling of data to improve the accuracy and resiliency of the system. Dzakiyullah Nur. et al. in the study [21] used t-distributed stochastic neighbour embedding T-SNE and autoencoder to classify the fraudulent activities and performed extraordinarily with precision rate of 0.98 and 1.00 for non-fraudulent and fraudulent transactions, respectively. As for their future work, the study recommended exploring other methods, such as Gaussian Mixture and Isolation Forest, to enhance the classification process further.

2.4 Challenges and Limitations

Although Data-Driven Predictive models are dependable and precise in predicting credit card it is important to note that there are still some limitations to their effectiveness. Based on previous research, the primary limitations include:

- **Availability of Relevant Data:** Financial institutions are sometimes hesitant to share data because they worry about the possibility of data leakage and compromising customer privacy. This can make it challenging to access relevant data.
- **Adaptive Fraud Patterns Over Time:** Fraudsters are always coming up with new ways to trick Fraud Detection Systems (FDS), which means that the models need frequent updates to identify these patterns[22].
- **Class Imbalance:** In large datasets of credit card transactions, credit card fraud occurrences are relatively rare, around 1% or less of the total transactions[23, 24]. This significant class imbalance poses challenges for Data-Driven Classification Models to effectively identify and predict both fraudulent and genuine classes with precission.
- **Feature Selections:** There are numerous features associated with any Transaction Data, many of which offer limited insights. Using those features can decrease the predictive power of the classifier.

2.5 Approach

Taking into consideration the limitations discussed in the preceding section, this study intends to conduct a comparative analysis of different supervised Classification machine learning and neural network algorithms, as well as ensemble and hybrid resampling approaches, to address the issue of class imbalance in the data. The following are the approaches that will be utilized in the study.

- Feature selection is essential in performance improvement and interpretability of the model. In this study, feature selection will be conducted using correlation coefficient analysis and Recursive Feature Elimination (RFE) to enhance the feature set.
- Various techniques for supervised classification will be utilized in the study, machine learning algorathims Logistic Regression, Decision Trees, Naive Bayes, Support Vector Machines (SVM), and ensemble methods such as Bagging (Random Forest) and Boosting (AdaBoost). Additionally, neural networks Artificial Neural Networks (ANN) will be implemented. After comparitive study based on diffferent evauation matrix top three algorathims will be selected for further evaluation.
- Class imbalance is a significant hurdle in building an efficient model for FDS. The study will use undersampling, oversampling and Synthetic Minority Oversampling Technique (SMOTE) to balance the data.
- Comparing the best model selected after implementing different sampling techniques.
- Examine the limitation of the system proposed.

CHAPTER 3

Preliminaries

In this section, we establish the foundation for our study by presenting an overview of the essential concepts, terminologies and methodologies that constitute the foundation of this study. This chapter will help the reader to understand the key principles behind the supervised data-driven methods used in this study, aiming to develop a robust credit card fraud detection system.

3.1 Machine Learning

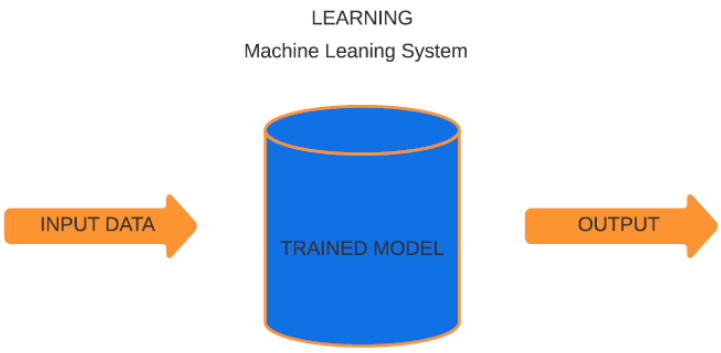

Figure 3.1: Machine Learning Flow

Machine learning is a sub-division of artificial intelligence that involves the use of algorithms to enable computer systems to learn from the data, detect patterns and make precise

future predictions or decisions without human intervention. Machine learning algorithms are fed with input historical data to train the model, a model learns the pattern in the data and uses this knowledge to predict future results. It is extensively used in several vital real-world applications like in diagnosing diseases, drug discovery in healthcare, recommender systems, language translation, facial recognition, supply chain optimization and many more. Machine learning has three categories: supervised, unsupervised and reinforcement learning. This study deals with utilizing supervised learning techniques to learn the pattern in the historical transaction data and predict fraud in future transactions.

3.2 Neural Networks

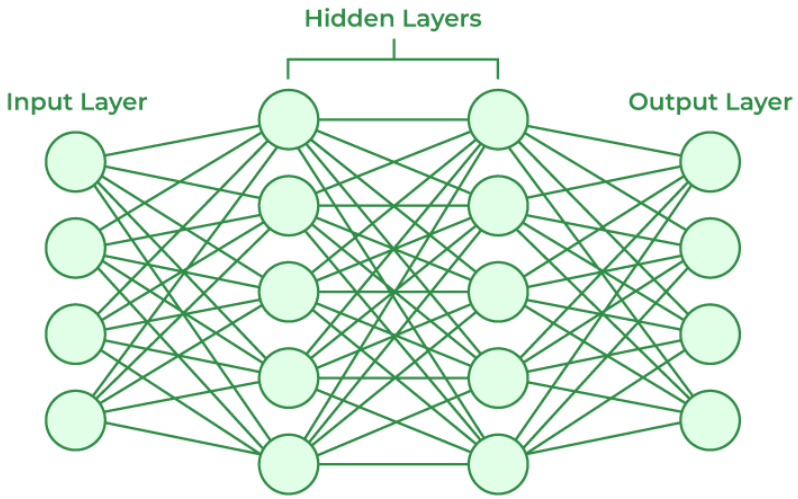

Figure 3.2: A Neural Network Architecture [28]

A neural network also is a method in artificial intelligence that trains models to process data in a way similar to the human brain; it mimics biological neurons signalling to one another. Deep learning is a type of machine learning that utilizes layered structures of interconnected nodes or neurons to develop system that learns from errors and enhances precision gradually.

A Neural Network consists of layers of nodes, and it has an input layer, one or more concealed layers, and an output layer; the final outcome of data processing of an artificial neural network is determined by the output layer. For instance, in our study of fraud detection classification problem, the output layer will have one output node, which will

give the result as 1 or 0. Each neuron connects to another and has an associated weight and threshold value.

3.3 Supervised Learning

Supervised learning is an approach in which a model is trained on well-labelled training data to find the pattern and make predictions or decisions. In this approach, the algorithm is provided with input-outputs, where the input signifies the features or attributes of the data, and the output signifies the corresponding desired result or label, after the model is trained it is provided with the new unseen set of input data and the algorithm will predict the output correctly.

A general machine learning algorithm equation is as follows:

$$\mathbf{Y} = \mathbf{f}(\mathbf{x})$$

Here, Y is the target or output, x is the input features, and f(x) is the target function, the target function is trained to predict the output Y correctly for the unseen input x. supervised learning can be divided into two subcategories: Classification and Regression. In classification problems, the output variable is a Class or group, for example, fraud or non-fraud, email spam or not spam,in regression, the output is in the form of continuous numerical values like house price, revenue and many more. In this study, we are using classification problems for fraud detection.

3.4 Classification

Classification is a supervised learning method that involves training a model to correctly identify input data's category or class label. Using input training data, the model predicts the probability that future data will fall into one of the predetermined categories. One example of this is email spam detection where the algorithm predicts the probability of an incoming email being spam and categorizes it as either spam or not spam. In this study, we will utilize binary classification, which involves categorizing incoming transactions as either fraudulent or genuine.

3.5 Ensemble Learning

Ensemble learning is a technique that merges two or more algorithms to achieve better results than compared to when the algorithms are used individually[29]. Rather than relying on just one model, we aggregate the outcomes of multiple models through methods like voting, weighted voting, or taking an average to predict the outcome. The models involved in ensemble learning can be either heterogeneous (different types of algorithms) or homogeneous (the same kind of algorithm).

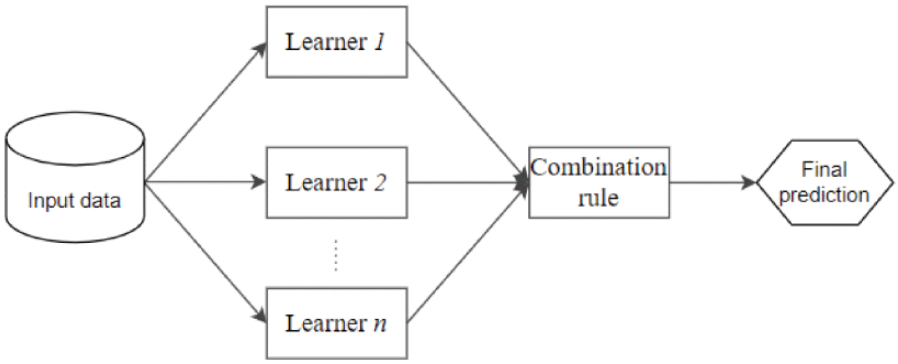

Figure 3.3: An Ensemble Learning Technique. [29]

3.5.1 Bagging

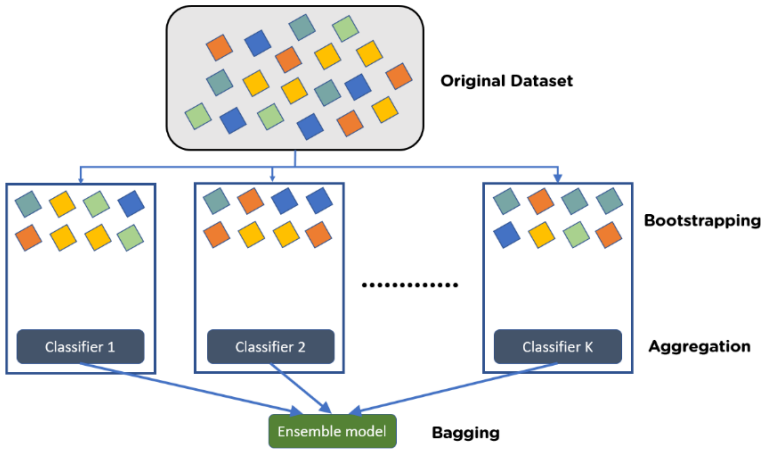

Figure 3.4: Bagging [30]

Bagging stands for bootstrap aggregating is an ensemble learning method that combines multiple models to make a prediction. It uses bootstraping on training datasets by randomly

selecting examples from the original training data with replacement. Each of these subsets is then trained on an individual model to make a prediction. The final prediction is made by aggregating the results of all the models. Decision Tree is commonly used as the base algorithm for bagging. In this study, we will be using Random Forest Algorithm, which is based on bagging ensemble learning method.

3.5.2 Boosting

Boosting methods are one of data science's most popular and extensively used algorithms. It is an ensemble technique that converts weak learners into strong learners, thus providing more accurate results. While the Bagging technique involves training weak learners in parallel, in this process a random sample of data is selected and fitted with a model and then trained sequentially with other models, with each model aiming to compensate for the misclassification of its previous learner by increasing the weight of misclassified data enabling the future learner model to focus on those instances and improve the overall performance of the model. This study uses Adaboost algorithm to classify incoming transactions into fraudulent or non-fraudulent categories.

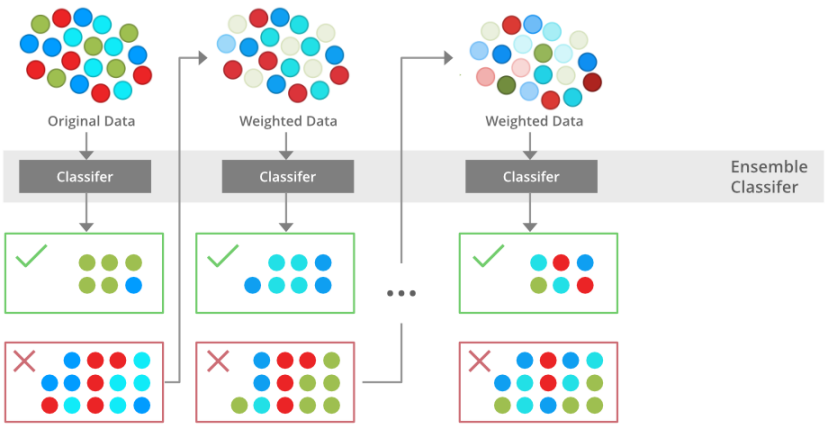

Figure 3.5: Boosting [31]

3.6 Resampling Methods

The real-world problems like fraud detection classification are highly influenced by class imbalance, which means that in the data provided, one class is much more prevalent than another, and this leads to poor performance from classifiers algorithms, as they tend to prioritize the larger class and overlook the smaller ones. There are several ways to keep check of the class imbalance in data, and this study uses different resampling techniques to mitigate the impact of class imbalance on a classifier. Resampling methods is a data level method which aims to balance the dataset without considering the attributes of a class label, this study utilized these three resampling techniques.

3.6.1 Random Undersampling

This method randomly reduces the majority calss by removing observations to create a balanced dataset. It is particularly effective for large datasets with many redundant majority class observations. However, the disadvantage of this method is that relevant observations may also be eliminated due to the random nature of the downsizing process.

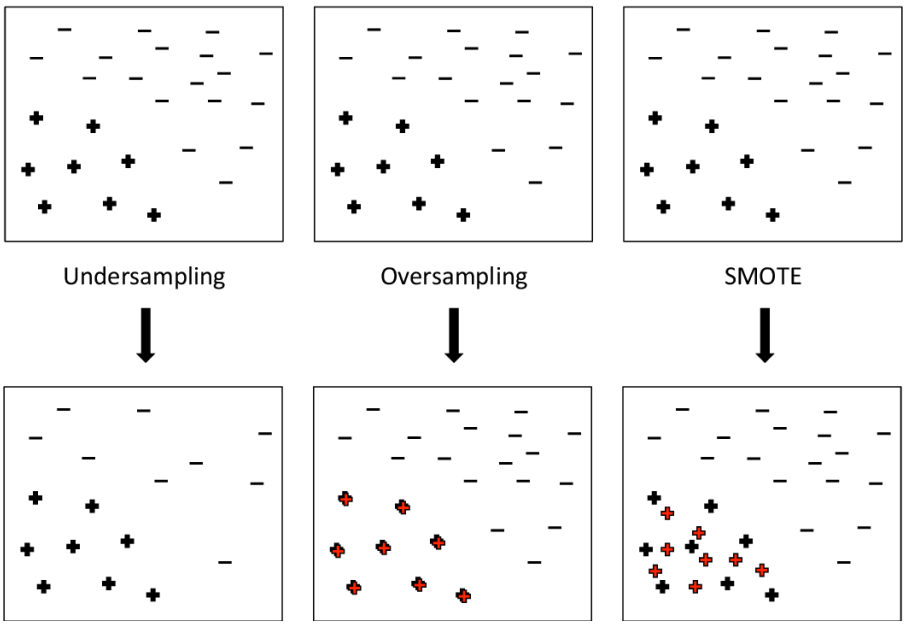

Figure 3.6: Resampling Methods [25]

3.6.2 Random Oversampling

This method focuses on minority class by duplicating them randomly to reduce class imbalance. However, unlike undersampling, the downside of oversampling is that it poses a risk of increasing overfitting and prolonging the model's training time.

3.6.3 Synthetic Minority Oversampling Technique (SMOTE)

SMOTE is a method to balance the dataset by oversampling the minority class by generating synthetic observation in the training dataset before fitting the model, it is highly beneficial in mitigating the risk of overfitting through oversampling.

3.7 Selected Algorithms

3.7.1 Logistic Regression

Logistic regression is one of the most popular machine learning classification algorithms. It is based on linear regression, it is a regression technique with categorical outcomes, and the prediction are in terms of the probability of outcome belonging to each class. It uses the sigmoid function to map the prediction values to probabilities between 0 and 1.

$$Sigmoid(z) = \frac{1}{1 + e^{-z}}$$

In a simple binary classification scenario, if there are input $x_1, x_2, ..., x_n$ features as and the target features as y which is 0 or 1.
For making a prediction, the logistic regression model takes into account the input features and calculates a weighted sum. This sum is then put through the logistic (sigmoid) function so the predictions fall between 0 and 1. The formula for logistic regression can be represented as follows:

$$P(y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n)}}$$

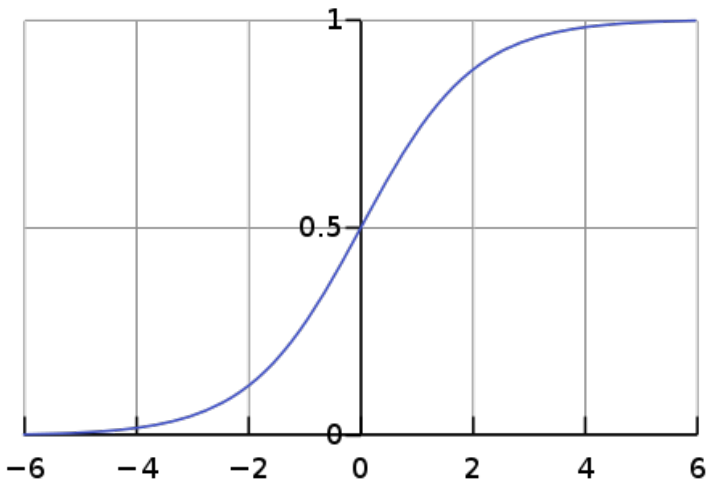

Figure 3.7: Sigmoid Function [32]

where$\beta_0, \beta_1, \beta_2, ..., \beta_n$ are the coefficient that model learn during training.
To make binary classification from the predicted probability, by default cut-off probability is taken as 0.5, which means if the probability of an observation is above 0.5 it will be classified as 1 and vice-versa.

3.7.2 Decision Tree

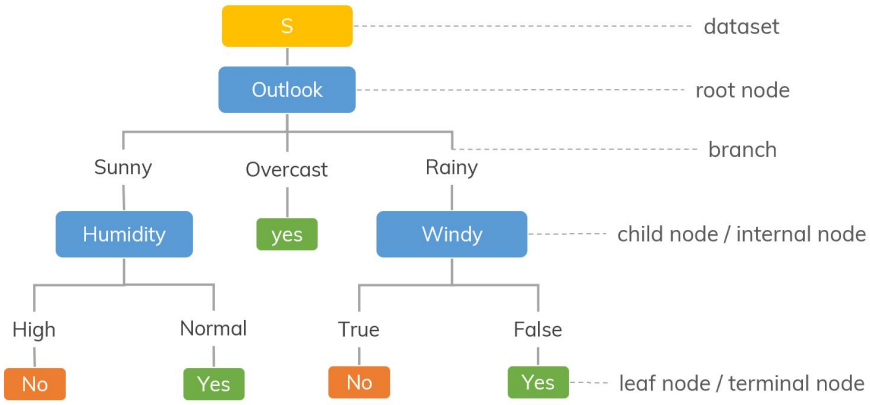

Figure 3.8: A Decision Tree Classifier [33]

The decision tree is a type of supervised learning algorithm that is often utilized for classification problems. It has the ability to handle both categorical and continuous variables, and its primary objective is to predicts the target variable by acquiring basic decision rules from the data features. A decision tree is a hierarchical structure algorithm that breaks down

complex decision processes into a series of simpler choices. It consists of nodes, branches, and leaves, with each internal node corresponding to a feature and each branch representing a decision based on that feature. Finally, each leaf represents a class label.

Being a hierarchical process, the algorithm recursively divides the data based on the most informative feature, ultimately leading to classification. The selection of the most important feature is determined by metrics such as the gini index or entropy, which measure the impurity in the node, although the algorithms may be susceptible to overfitting, their interpretability and emphasis on key features represent significant advantages.

3.7.3 K-Nearest Neighbors

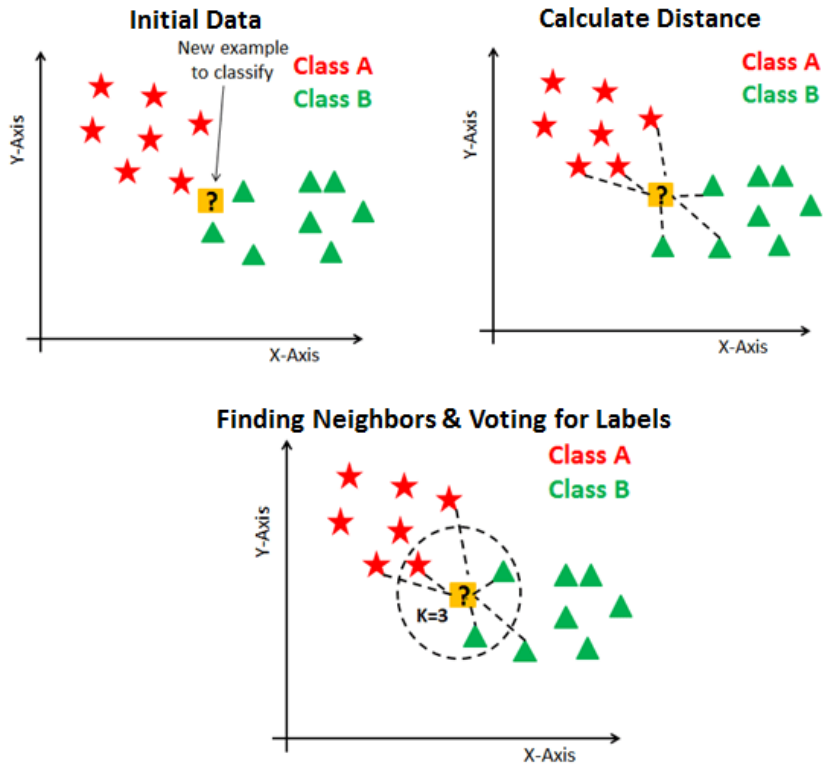

Figure 3.9: kNN working [34]

The k-nearest neighbours (KNN) algorithm is a non-parametric supervised machine learning technique that is straightforward and simple to implement. It uses the concept of proximity to classify or make predictions about a data point's grouping. It functions on the premise that data points with similar characteristics are likely to have similar labels or values.

The fundamental concept of the kNN classifier is to assign a classification label to a data point by considering the classification labels of its *K* closest neighbours in the feature space. The algorithm selects the *K* data points in the datasset that are closest to the data point being classified, using a distance metric like Euclidean distance, and then assigns the label that is most common among those neighbours.

In the training phase, the KNN algorithm stores the complete training dataset as a reference point. To make predictions, it uses a selected distance metric to determine the distance between the input data point and all the training examples. The algorithm identifies the *K* closest neighbours after determining the distances between the input data point and all the training examples. It then evaluates the class labels of these neighbours and selects the most frequently occurring one to assign as the predicted class for the new data point. kNN is a widely used algorithm for tasks such as image recognition, recommending products or services, and detecting anomalies.

3.7.4 Naive Bayes

The Naive Bayes classifier is widely used in machine learning and is known for its efficiency in classifying data, especially in natural language processing, text classification, and spam detection. It is a probabilistic algorithm that applies Bayes' theorem and assumes that the features are conditionally independent given the class label. This means features do not influences each others presence. Although it may seem like an unrealistic assumption, the Naive Bayes classifier performs impressively in various practical scenarios, and the main advantage is its Efficiency and fast training and testing time.

The Naive Bayes classifier computes the posterior probability of a class based on a set of features, which is achieved through Bayes' theorem:

$$P(Y|X) = \frac{P(X|Y).P(Y)}{P(X)}$$

here,

$P(Y|X)$ is the posterior probability of class given the feature.

$P(X|Y)$ is the probability of a feature given the class.

$P(X)$ prior probability of a class.

$P(Y)$ feature probability.

3.7.5 Random Forest

Random forest is an ensemble learning algorithm. It works on the principle of bagging as stated in section 3.5.1, which involves making a model working with the combination of multiple weak learners helping in accuracy improvement. Random forest uses a decision tree as the weak learner. Decision tree model performance can be hijacked with the presence of few correlated features; random forest helps to decorrelate those features and improve the model's overall performance.

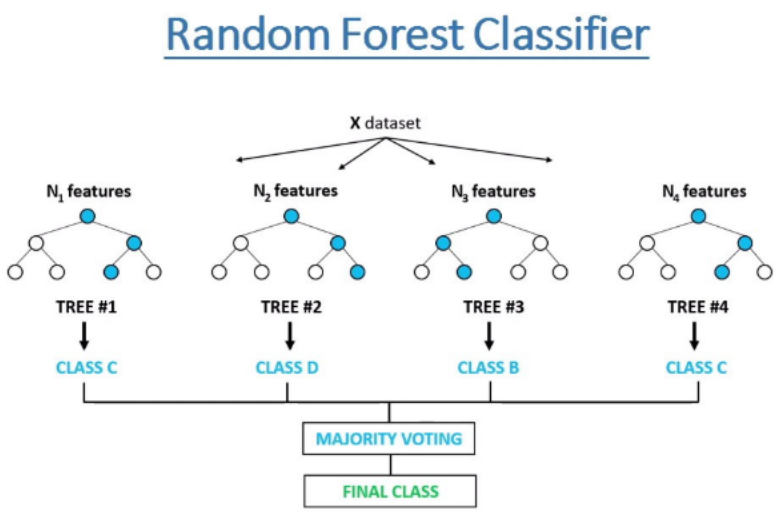

Figure 3.10: Random Forest [35]

In addition to bootstrapping samples are randomly selected with replacement from the training dataset, it also randomly select subset features for training the different decision tree model and aggregates the outcomes of weak learners into a better classifier. Random forest is a powerful and widely used algorithm due to its ability to reduce the possibility of overfitting

a model. It can handle noise, non-linear, and high-dimensional data by providing valuable insights into feature importance. Thus, minimizing the negative effect of multicollinearity in the model.

3.7.6 Adaptive Boosting

Adaptive boosting, also known as Adaboost, is an effective technique for ensemble learning that utilizes the boosting method to enhance the performance of weak learners. The algorithm combines the predictions of the weak learners by iteratively adjusting the emphasis on misclassified instances. Its objective is to generate a robust classifier by iteratively learning from the errors of weak learners.

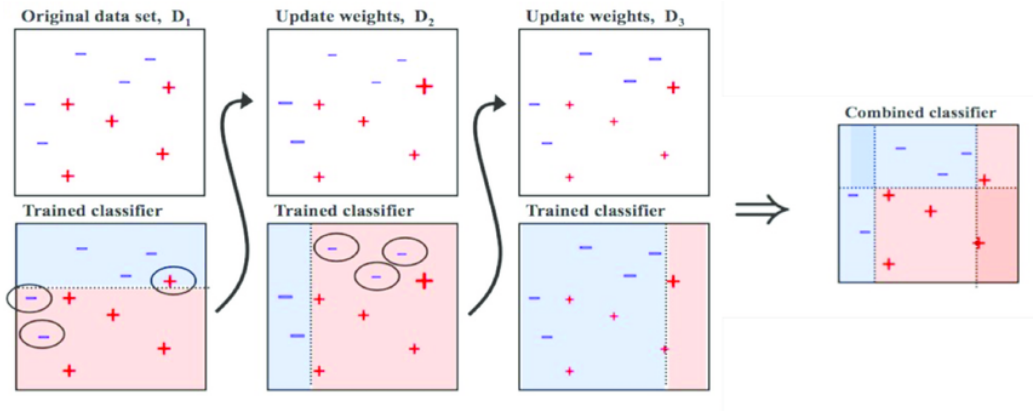

Figure 3.11: Adaptive Boosting [36]

In this algorithm, the model begins by initializing equal weights to all data points. During each iteration, it assigns higher weights to data points that are wrongly classified. The model then places more emphasis on the points with higher weights in the next iteration. This process continues until a lower error in classification is achieved. Adaboost is helpful in improving accuracy and minimizing overfitting of the model to a minimal level.

3.7.7 Support vector machine

Supppot vector machine is a robust classification algorithm for two-class classification problem. Its primary objective is to identify the optimal hyperplane that separates the classes in the feature space with the maximum margin. It works very well for both linear and non-linear

data.

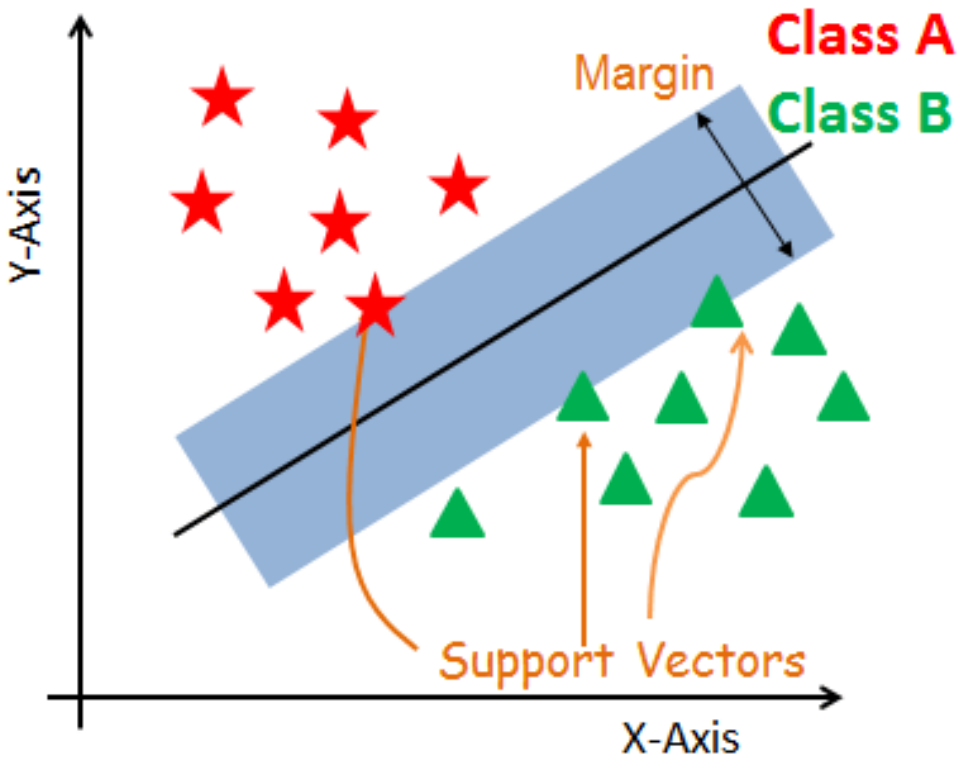

Figure 3.12: Support vector machine [37]

As mentioned above SVM algorithm aims to locate a hyperplane that can effectively separate the two classes. It is accomplished by identifying the support vectors, which are data points that are closest to the hyperplane. In the real-world problem, the data points are not always linear separable, svm often utilizes a kernel to transform the original feature space into a higher-dimensional space where the data points are more likely to be linearly separable. It is essential for a hyperplane to have a maximum margin because if the margin is low, there will be a high chance of misclassification in testing or unseen data. The key advantage of this algorithm is its high accuracy and its ability to handle high dimensional data with large number of features.

3.8 Evaluation Metrics

Evaluation metrics serve as tools to assess the performance of classification models by comparing their predictions with the actual outcomes. These metrics play a crucial role in assessing the efficiency of an algorithm on a dataset by analyzing their predictions against actual outcomes. They provide valuable insights into the model's performance and are helpful in comparing various models or adjusting parameters. This study analyzes the evaluation metrics used for classification problems.

3.8.1 Confusion matrix

A confusion matrix is a tabular representation that provides a thorough understanding of a Classification algorithm's performance on a dataset. It is used to evaluate the accuracy of the model's predictions by comparing them with the actual outcomes. In the case of binary classification, the confusion matrix is a 2×2 matrix that includes True positive, False positive, True negative, and False negative predictions.

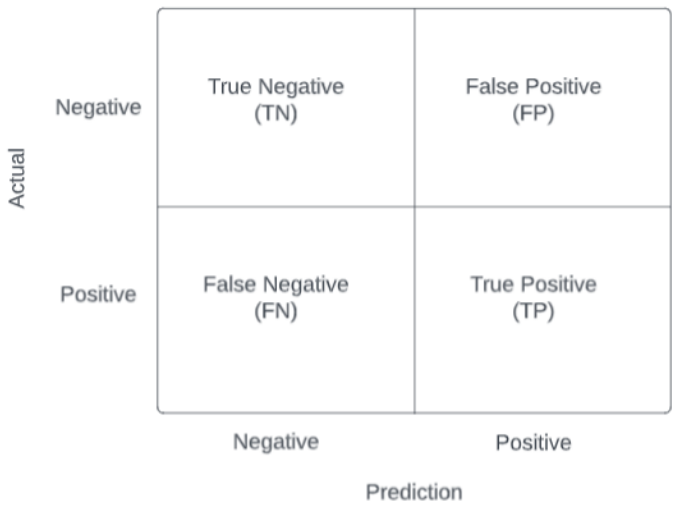

Figure 3.13: Confusion matrix

True Positive (TP) is when the algorithm predicts positive, and the actual value is also positive.

False Positive (FP) is when the algorithm predicted positive, but the actual value was nega-

tive.

True Negative (TN) is when both the predicted and actual value is negative.

False Negative (FN) is when the prediction is negative, but the actual value is positive.

3.8.2 Accuracy

It is the proportion of all the correct predictions compared to all the predictions. In the imbalance classification problem, accuracy can be misleading as accuracy can be artificially increased by correctly classifying the majority class without discriminating between the classes.

$$Accuracy = \frac{TP + TN}{ToatlObservation}$$

3.8.3 Precision

It is the proportion of true positive values and all positive predictions. It focuses on the proportion of true positive predictions among all positive predictions. A high precision value means the model is making accurate positive predictions. It's a measure of the model's ability to avoid false positives.

$$Precision = \frac{TP}{TP + FP}$$

3.8.4 Recall

It is also known as sensitivity and True positive rate, it is the proportion of true positive prediction values and all the actual positive instances. A higher recall value means the model identifies most positive instances well.

$$Recall = \frac{TP}{TP + FN}$$

3.8.5 Specificity

It is the proportion of true negative and all actual negative instances. A high specificity indicates that the model is good at identifying most of the negative instances.

$$Specificity = \frac{TN}{TN + FP}$$

False positivity rate = 1 - Specificity

3.8.6 F1-Score

F1-score is the harmonic mean of precision and recall. It offers a balance between precision and recall, which is particularly beneficial for the imbalanced dataset. Its value ranges from 0 to 1 with a high F1 score indicates that the model has good balance between precision and recall.

$$F1Score = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

3.8.7 Area Under Receiver Operating Characteristic curve

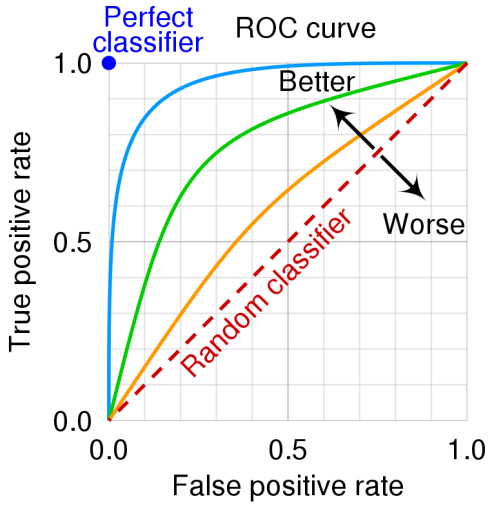

Figure 3.14: AUC-ROC Curve [38]

ROC curves show the comparison between the true positive rate and false positive rate, and the area under the curve (AUC-ROC) measures how well the model can differentiate

between different classes. The AUC-ROC value ranges between 0 and 1, where higher values indicate better model performance. A model with a 0.5 value means a random classifier and 1 value represents a model with strong classifying power between classes.

CHAPTER 4

Methodology

In this section of the study, the method applied to achieve objective two, three, four and five are discussed. It tells about the flow of the method used in the study.

4.1 Dataset

Dataset plays a crucial role in any analysis, collection of data is a vital and costly process, especially in the financial domain, as it involves people's finances and personal confidential information. At the same time, the quality and relevance of the dataset can significantly impact the accuracy and applicability of the research findings. The dataset used in this study is taken from the Kaggle [39], which was collected and analysed in a research [40] by Worldline and the Machine Learning Group of ULB (Universitè Libre de Bruxelles). It consists of credit card transaction data from European cardholders for two days in September 2013. The dataset includes a total of 284,807 transactions, with only 492 of them, or 0.172%, being fraudulent transactions, which makes this dataset highly unbalanced.

The dataset consists of 31 variables, with a total of 30 independent numerical features. Due to confidentiality reasons, 28 of these features have been generated using principal component analysis and are denoted as V1, V2, ..., V28. The other two independent features are 'Amount'and "Time'. 'Class' is the dependent or target variable which is classified as

Feature	Description
Amount	Transaction Amount
Time	Time elapsed between each transaction and the First transaction
V1	First Principal Component
V2	Second Principal Component
–	–
V8	Last Principal Component
Amount	Class Label (1 = Fraud and 0 = Genuine)

Table 4.1: Description of Dataset

either 1 for fraud or 0 for 'non-fraud'.

4.1.1 Descriptive analysis

As discussed above the dataset is imbalanced, which is true for nearly all of the real-world fraud situations, majority of transactions are legitimate, but our focus is on identifying the minority class that represents fraudulent activity. Figure 4.1 displays the bar plot distribution of the class distribution in the dataset.

Multicollinearity is often referred to as a necessary evil in predictive modelling. It arises when two or more independent variables are correlated with each other, which leads to unstable and unreliable model results. Figure 4.2 presents a correlation plot that explains the pairwise correlation between variables. The plot reveals that the 'Amount' and V2 have a moderate correlation coefficient value of -0.53. However, the other features do not have a high correlation. It is likely because the dataset has already undergone PCA, which is also used for reducing multicollinearity.

Figure 4.3 represent the average amount of fraudulent transaction and non fraud transactions, whereas Figure 4.4 shows that the highest numberof fradulent transaction happens at 0300 hours and 1200 hours.

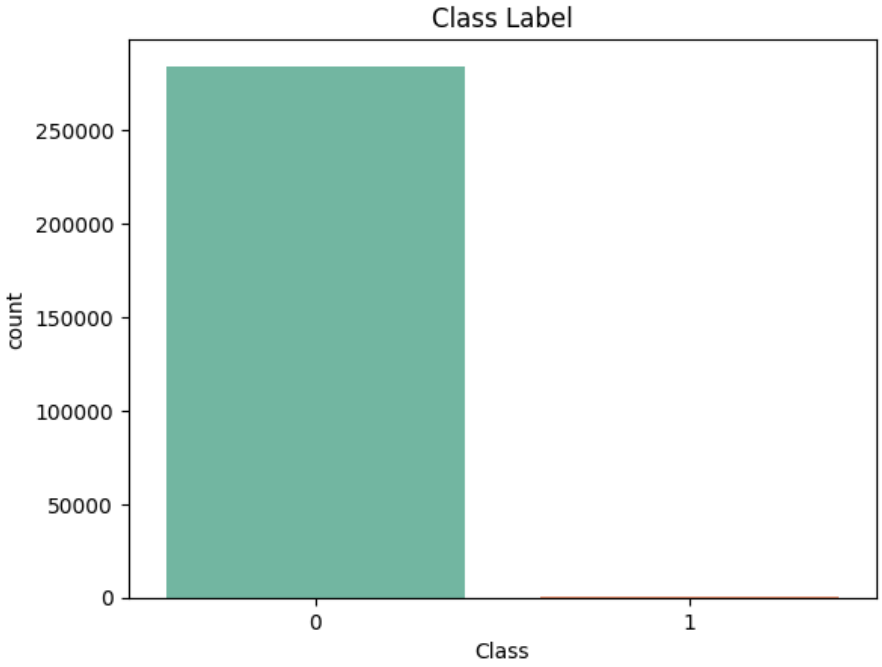

Figure 4.1: Class distribution in the dataset

4.1.2 Data Preprocessing

Data preprocessing is done to enhance the quality of the data by cleaning it and correcting its formats. It helps in reducing noise and enables better training of the model, dataset used for this study does not have null values and no outliers because of the PCA transformation. 'Time' feature, which represents the duration between the transaction and the first transaction, is converted into hours to enhance the analysis of the time variable in the study.

Transforming features into a consistent scale through data standardization is a crucial process in machine learning to ensure each feature has a comparable range and prevent certain features from dominating others. This process ultimately improves the stability and performance of machine learning models. mathematically it is represented as:

$$z = \frac{x - \mu}{\sigma}$$

where μ is the mean of the variable values and σ is the standard deviation of the same feature. Feature 'Amount' and 'Time' is standardized by using RobustScaler from scikit-learn library.

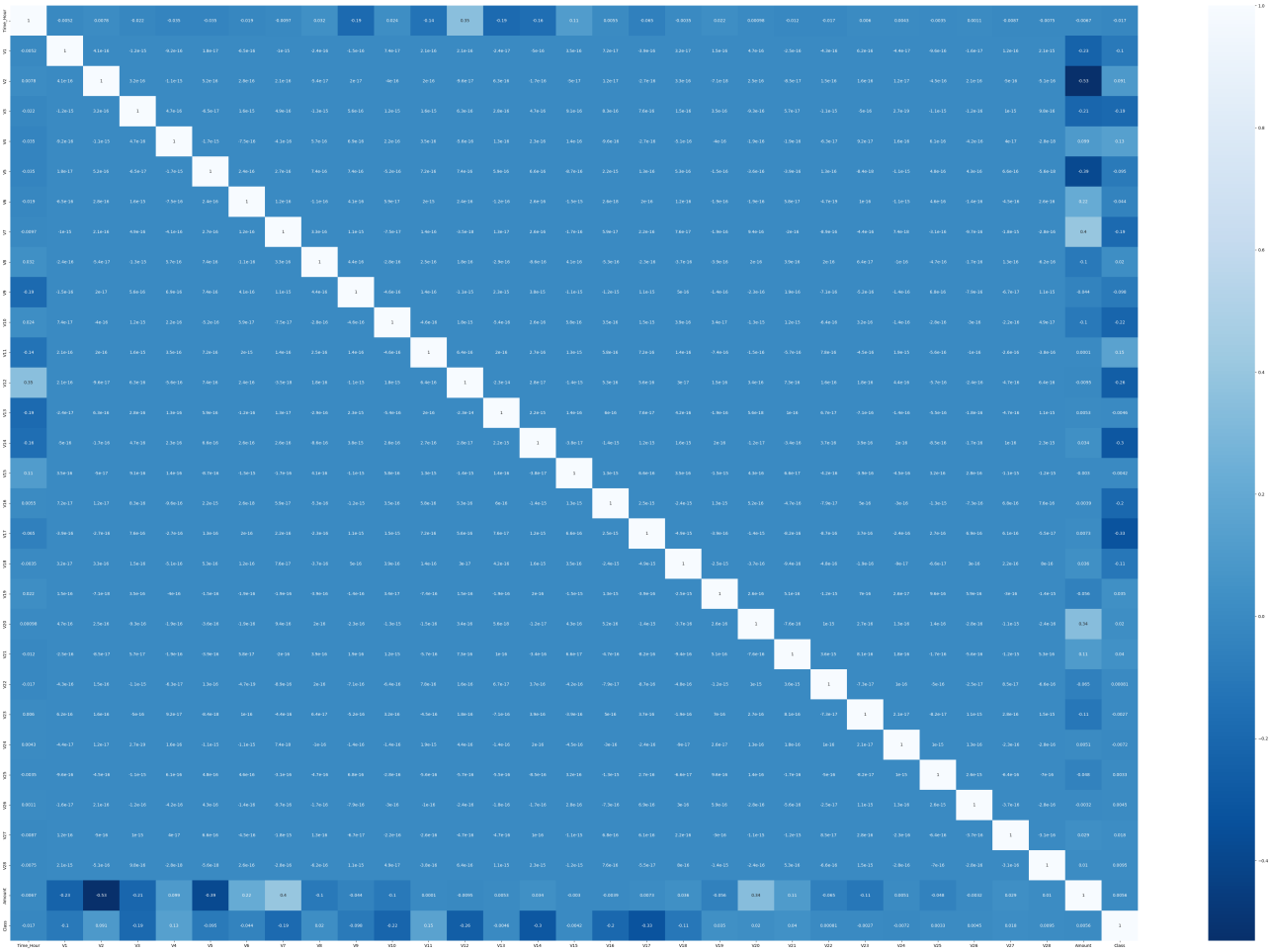

Figure 4.2: Correlation plot

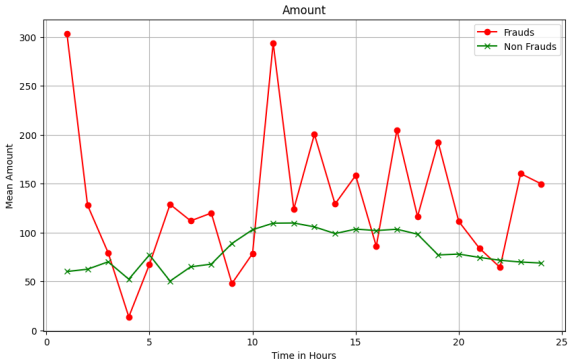

Figure 4.3: Average amount of class vs. time

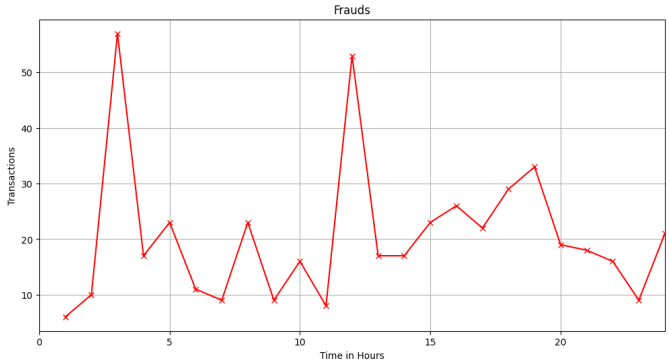

Figure 4.4: Fraud count vs. time

4.1.3 Train-test split

The dataset used in the study has a total of 284,807 observations with 492 fraudulent transactions, data is split into training and testing stratified samples of 75% and 25% each of the com-

plete dataset for each algorithm using the scikit-learn library train_test_split model_selection's train_test_split method.

Once the data was split, the training dataset had 213605 observations and 31 features, while the testing dataset had 71202 transactions with the same number of features. The proportion of fraudulent transactions in both datasets is the same, at 0.172%. The training dataset has 369 fraudulent transactions, and the training dataset contains 123 observations. Hyperparameter tuning, feature selection and training of the model is done on the training sample, and the test sample is used to measure the performance of the model.

4.2 Hyperparameter Tunning

Hyperparameters are specific parameters that the user sets to regulate the learning process, these are external to the model and can be changed to get optimized results from the model. But, before starting with the model building and hyperparameter tuning, lets understand the cross-validation. It is a technique used to evaluate the performance of a model. It involves splitting the available dataset into multiple subsets for training and validation helping to overcome overfitting and providing an insight into the model's stability. We used k-Fold and stratified k-fold cross-validation where k represents the number of subsets of the dataset.

Every model has there own set of hyperparameters, and its tunning is necessary to get optimal results. Hyperparameter tuning is a process of finding the best set of hyperparameters to achieve those results. It involves trying different combinations and evaluating their impact on the model's performance. GridSearchCV and RandomSearchCV with 5-fold cross-validation is used on multiple algorithms to select the best combination of the hyperparameter in the study.
ROC AUC is taken as a scoring parameter in hyperparameter tunning for fraud detection classification problems because it offers a thorough evaluation of a model's capacity to differentiate between positive and negative classes at different threshold levels. It considers both sensitivity (true positive rate) and false positive rate, which makes it ideal for imbalanced datasets and balances the need to identify true positives while minimizing false positives correctly.

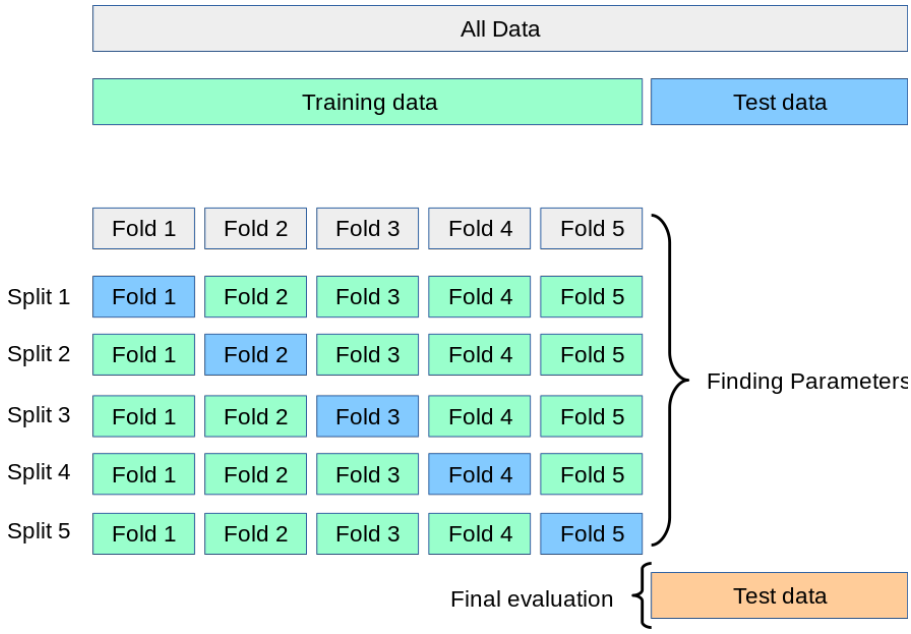

Figure 4.5: Cross validation [41]

4.3 Feature Selection Using Recursive Feature Elimination

Feature selection is crucial in model training as it enhances model performance and reduces the chance of overfitting. By focusing on the most relevant attributes, feature selection improves model accuracy, interpretability, and model generalisation. Additionally, feature selection can lead to faster training times, optimise resource utilisation, and increase interpretability.

In this study, Recursive Feature Elimination (RFE) is utilised for selecting essential features. It is the process of iteratively eliminating the least important features from a dataset based on their effect on model performance. It starts with training a model on the complete feature set, assessing the significance of each feature, and gradually eliminating the least important ones until a desired number of features is obtained.

4.4 Training and testing

After tuning the hyperparameters for each predictive model, A total of eight supervised classification models are constructed with the selected hyperparameters, the models are trained on the training dataset, and tested on the test sample before evaluating their performance.

4.5 Performance Evaluation

Once a model has been constructed, it is essential to assess its performance and compare it to other models to determine which one performs most accurately for the given scenario and dataset. However, in cases where the dataset is imbalanced, and the majority class accounts for approximately 99% of the observations, accuracy may not be the most reliable measure. For instance, if out of 100 transactions, only one observation is fraudulent and a model overgeneralised the dataset and classifies all observations as non-fraudulent, it may have an accuracy of 99%, but it fails to identify the minority class, which is the main objective of the model.

To avoid this situation, this study utilises precision, recall, F1-score, and roc_auc score to compare models, for detecting the maximum fraudulent transaction correctly a higher recall value is preferred. Apart from classifying frauds correctly, it is also essential to avoid false positives, which are the transactions that are incorrectly classified as fraud. This can lead to inconvenience for credit card users and loss of business for financial institutions in the long run, this can be avoided by the higher value of precision. F1-score, as a harmonic mean of precision and recall, is a reliable measure to compare model performance.

Area under ROC Curve is also an efficient measure of the model performance by indicating how well it can differentiate between positive and negative cases. A higher value indicates a better performing model.

This study chooses the top three algorithms based on performance evaluation measures, applies resampling techniques to enhance their performance, and selects a final model suitable for a robust credit card fraud detection system.

4.6 Resampling and comparision

Classification in imbalanced data can be a difficult task. However, this can be mitigated by training the model on resampled data with equal class distribution. Resampling is a technique used to address unequal class distributions in classification problems, as mentioned in section 3.6.

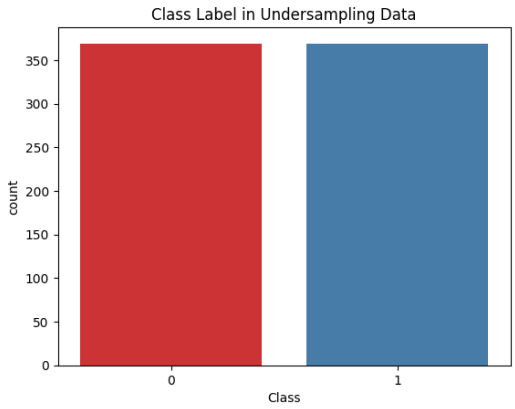

Figure 4.6: Undersampled data

Figure 4.7: Oversampled data

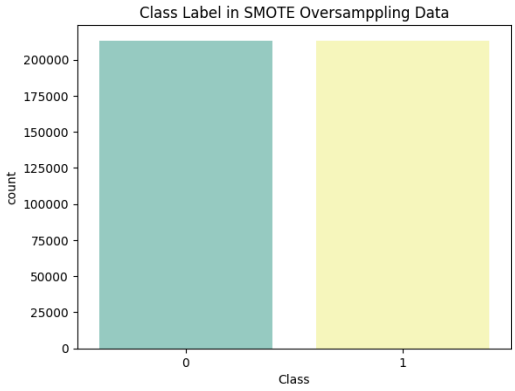

Figure 4.8: SMOTE sampled data

When dealing with the classification of imbalanced classes, the open-source library Imbalance Learn from Scikit Learn provides useful tools. This library allows for dataset resampling in various ways, including Undersampling, Oversampling, and SMOTE Oversampling. The dataset is resampled using these technique Fig 4.6, Fig 4.7 and Fig 4.8 represent the class distribution of training dataset after resampling.

The top three models selected will be trained on resampled data using undersampled, oversampled, and SMOTE sampled data. Finally, the algorithms' performance will be

analyzed, and the best sampling technique will be chosen. This study aims to identify the best-performing algorithm coupled with the correct sampling technique based on various evaluation metrics and recommend it for building a robust credit card fraud detection system.

CHAPTER 5

Results and Analysis

5.1 Logistic Regression

The algorithm is applied in two stages. Firstly, with all the features included, and secondly, with the features selected using the Recursive feature elimination method. Hyperparameter tuning is performed on both models.

Base Model

Best Parameters: C = 0.1, solver = 'liblinear'

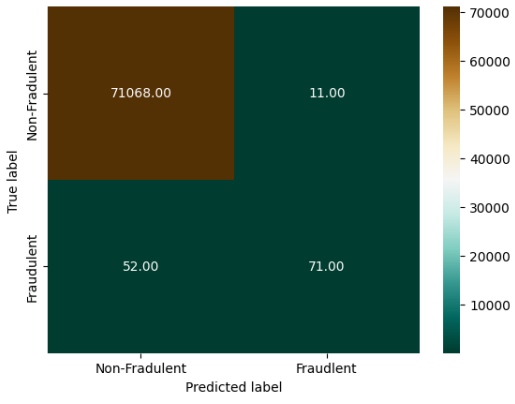

Figure 5.1: Confusion matrix of Base LR

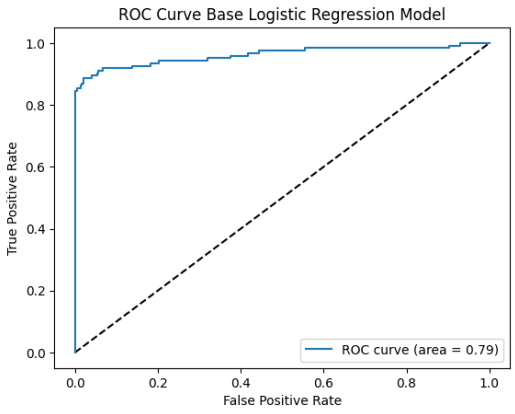

Figure 5.2: ROC_AUC curve of Base LR

LR showed decent performance with a precision score of 0.8658, meaning accurately identifying 86.58% of predicted fraud cases, a recall score of 0.5772, indicating the model was

able to capture 57.72% of actual fraud cases that could be considered as low, an F1 score of 0.6926 on the test dataset. The accuracy score was 0.9991, which was expected due to the imbalanced dataset. The ROC AUC score of 0.7885 was satisfactory in correctly classifying positive and negative classes. Fig 5.1 and 5.2 shows the confusion matrix and ROC_AUC curve for the model.

Reduced Model

Best Parameters: C=0.1, solver='liblinear'

The features include 'Amount', 'V3', 'V5', 'V7', 'V8', 'V9', 'V10', 'V11', 'V12', 'V13', 'V15', 'V16', 'V20', 'V21', 'V24' were selected with the help of RFE.The algorithm performed relatively better than the base model, with a recall score increased to 0.5934 an increase of about 1.5% of correct fraud transaction detection and an f1 score of 0.7019. However, the precision measure slightly decreased to 0.8558, indicating a slight decrease in its ability to avoid false positives. Nonetheless, the ROC AUC score improved to 0.7966, making the reduced model better than the base model in differentiating between positive and negative classes.

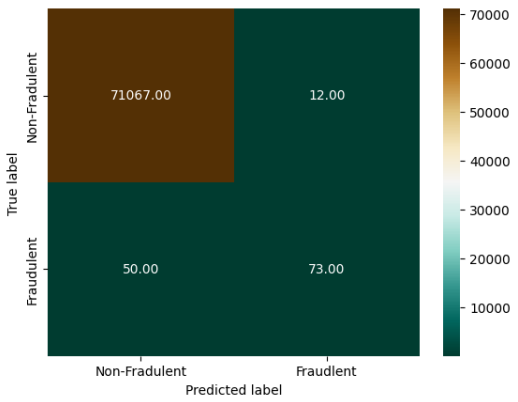

Figure 5.3: Confusion matrix of Reduced LR

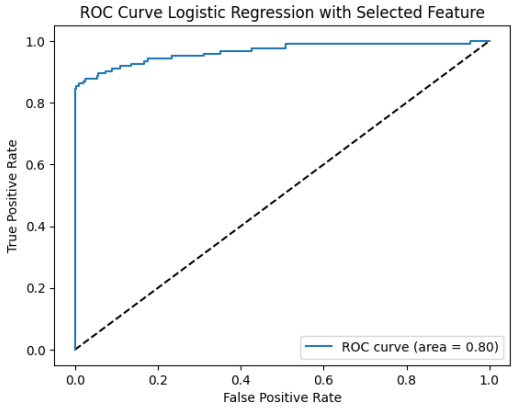

Figure 5.4: ROC_AUC curve of Reduced LR

Algorithm	Accuracy	Precision	Recall	F1 Score	ROC-AUC
Base Logistic Regression	0.9991	0.8658	0.5772	0.6926	0.7885
Reduced Logistic Regression	0.9991	0.8588	0.5934	0.7019	0.7966

Table 5.1: Evaluation Measures for Logistic Regression Algorithm

Table 5.1 shows the Evaluation score of both model. Both models used the same hy-

perparameter, which included a C regularisation value of 0.1. The model with the smaller value of C increased the strength of the regularisation, preventing overfitting. 'liblinear' optimization algorithm was used as it is well suited for binary classification. Although both models performed similarly, logistic regression with reduced features was chosen for further comparison because models with fewer features are more time-efficient.

5.2 Decison Tree

Base Model

Best Parameters: criterion='entropy', max_depth=5, min_samples_leaf=2, min_samples_split=10

With 5-fold cross-validation hyperparameter tunning, entropy is selected as the criterion, which uses information gain to measure impurity in the node and with a maximum depth of 5 of the decision tree. The decision tree classifier with all the features gives an accuracy of 0.9992. It accurately identifies 87.36% of the predicted frauds with fewer false positives, a recall score of 0.6747, which represents it predicts 67.47% of the 123 fraudulent cases in the test dataset correctly, and an F1 score of 0.76146 on the test dataset, which is quite exceptional because of the simplicity of the algorithm, an excellent ROC AUC score of 0.8373 also represent that model has been able to generalise the dataset.

Algorithm	Accuracy	Precision	Recall	F1 Score	ROC-AUC
Base Decision Tree Classifier	0.9992	0.8736	0.6747	0.7614	0.8373
Reduced Decision Tree Classifier	0.9990	0.8481	0.5447	0.6633	0.7722

Table 5.2: Evaluation Measures for Decision Tree Classifier

Reduced Model

Best Parameters: criterion='entropy', max_depth=5, min_samples_leaf=4, min_samples_split=5)

In the reduced model, the best hyperparameter was with entropy as the criterion and the maximum depth of the tree was selected as 4. The features 'V9', ' V10', ' V12', ' V13', and 'V16' were selected with the help of RFE. The algorithm's power of classification decreased in all the measures with a precision score of 0.8481, a recall of 0.5447, which fell by around 13%,

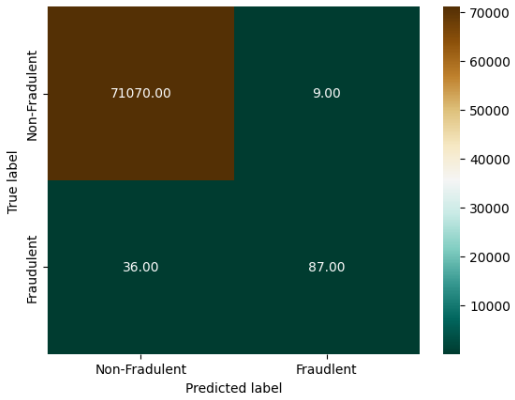

Figure 5.5: Confusion matrix of Base Decision Tree Classifier

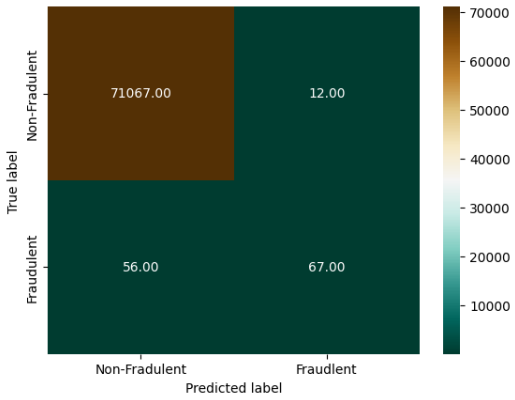

Figure 5.6: Confusion matrix of Reduced Decision Tree Classifier

increasing the false negative instances, F1 score of 0.6633 and a roc_auc of 0.7722, which is less than the results of the algorithm with all features.

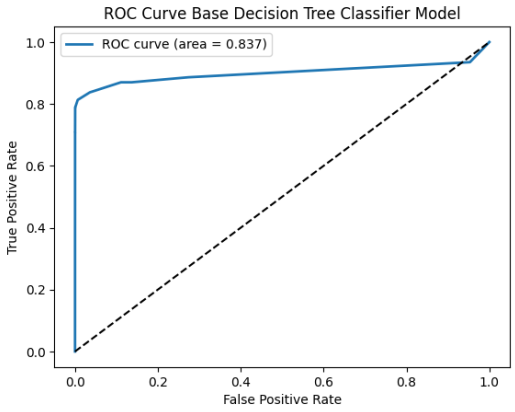

Figure 5.7: ROC_AUC curve of Base Decision Tree Classifier

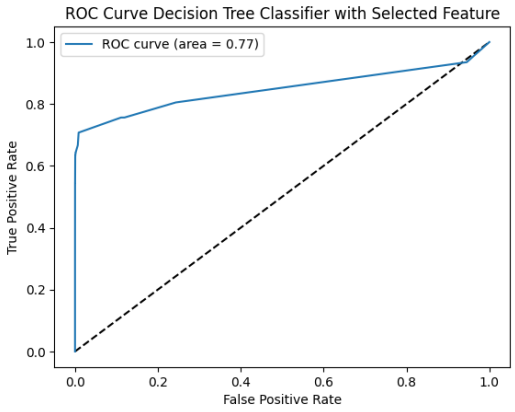

Figure 5.8: ROC_AUC curve of Reduced Decision Tree Classifier

With fraud classification correctness of 67.47% on the test data, the base decision tree model is selected for comparison with other algorithms. The reduced model may have inferior performance compared to the base model due to a decrease in features leading to a loss of information and limited discriminative power, ultimately resulting in underfitting.

5.3 K-Nearest Neighbors

Best Parameter: n_neighbors=8, weights='distance'

kNN is an instance-based algorithm that does not use coefficients or weights for features. At the same time, RFE assigns weights to features to determine their importance in the algorithm; this is the reason RFE do not work with the kNN algorithm. Adiitionally, kNN is a distance-based technique and removing features can lead to suboptimal performance of the algorithm. Therefore, feature selection is not done in this algorithm.

Using 5-fold crossvalidation hyperparameter tunning on the training dataset, the optimal hyperparameter chosen is a value of 8 for n_neighbors(k), which signifies the number of neighbours considered when making a prediction. Choosing a good value for k is important to maintain bias-variance trade-off. If k is too small, the model may not generalize the data well, leading to high variance. On the other hand, if k is too large, there may be a higher bias. The weight is selected by distance, meaning closer neighbours have a more significant influence on the prediction.

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC-AUC**
K-Nearest Neighbors Classifier	0.9994	0.9450	0.6991	0.8037	0.8495

Table 5.3: Evaluation Measures for K-Nearest Neighbors Classifier

The kNN results in an accuracy of 0.9994, accompanied by excellent scores of 0.9455, 0.6991, and 0.8037 for precision, recall, and f1-score, respectively, for the training dataset, with a good balance of precision-recall with good f1 score, the algorithm was able to accurately predict 86 out of the 123 fraudulent transactions and 94.55% correct predictions of all fraudulent predictions made by the model. The ROC_AUC score of 0.8495 is equally impressive when compared to previous algorithms.

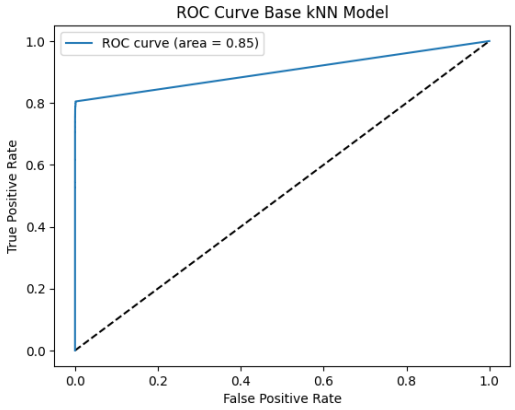

Figure 5.9: ROC_AUC curve of K-Nearest Neighbors Classifier

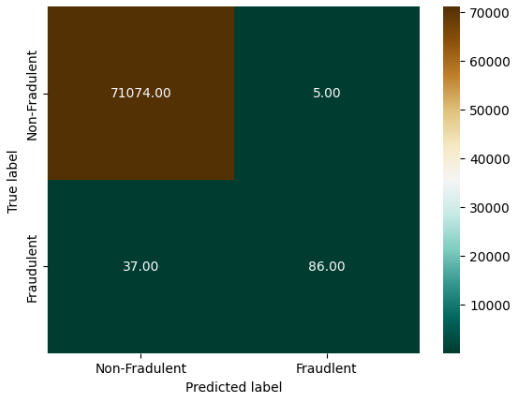

Figure 5.10: Classification Matrix of K-Nearest Neighbors Classifier

5.4 Naive Bayes Classifier

Best Parameter: var_smoothing=1.0

Naive Bayes works with probabilities and conditional probabilities through hyperparameter tuning; a smoothing hyperparameter of 1 is chosen. Naive Bayes classifiers rely on the assumption of features independence given the class label, and this assumption is violated if features are removed during the RFE process. Therefore, feature selection is avoided in this method.

The performance metrics of the algorithm are illustrated in the Table 5.4. The algorithm

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC-AUC**
Naive Bayes Classifier	0.9950	0.1463	00.3902	0.2128	0.6931

Table 5.4: Evaluation Measures for Naive Bayes Classifier

attains an accuracy of 99.50, with a precision score of 0.1463, which is very low, with only 14.63% of all the predicted fraud were accurate and the rest being false positive, recall score of 0.3902 and 0.2128 F1 score. However, it is worth noting that only 39.02% of fraudulent transactions were correctly classified, and the ROC_AUC score was only 0.6931, indicating poor performance metrics.

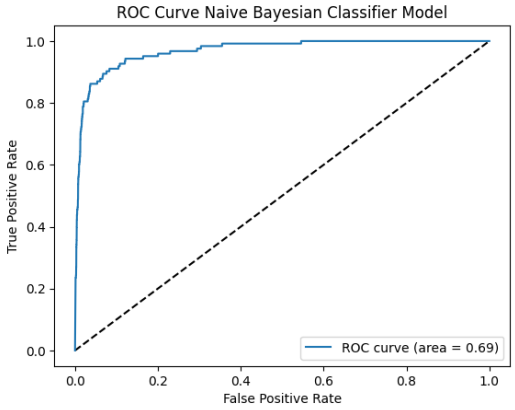

Figure 5.11: ROC_AUC curve of Naive Bayes Classifier

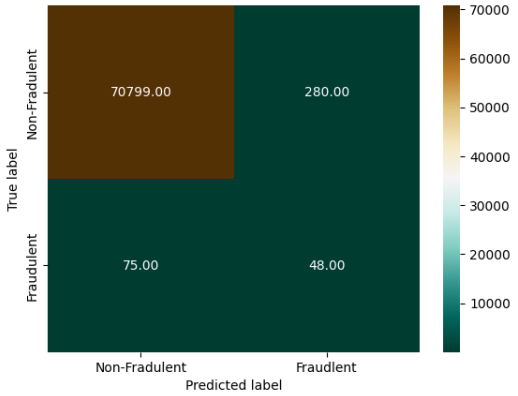

Figure 5.12: Classification Matrix of Naive Bayes Classifier

5.5 Random Forest Classifier

Base Model

Best Parameters: max_features=5, n_estimators=70

Random forest is bagging ensemble learning, which uses decision tree as the base learner.

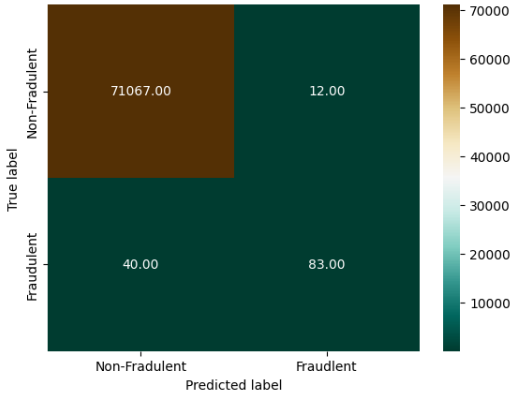

Figure 5.13: Confusion Matrix of Base Random Forest Classifier

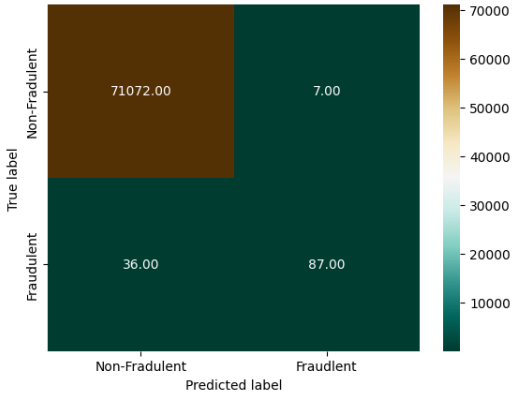

Figure 5.14: Confusion Matrix of Reduced Random Forest Classifier

By hyperparameter tunning 70 trees ensemble learning model performed optimal for the dataset, and a maximum feature of 5 was selected for building the model. The algorithm achieved an F1 score of 0.7614 in classifying fraudulent and non-fraudulent cases, which is decent, with a precision of 0.8736, meaning accurately identifying 87.36% of predicted fraud cases and a recall of 0.6747, this indicates that the ensemble method performed effectively despite imbalanced data. A ROC_AUC score of 0.8373 was also achieved, which is considered

satisfactory.

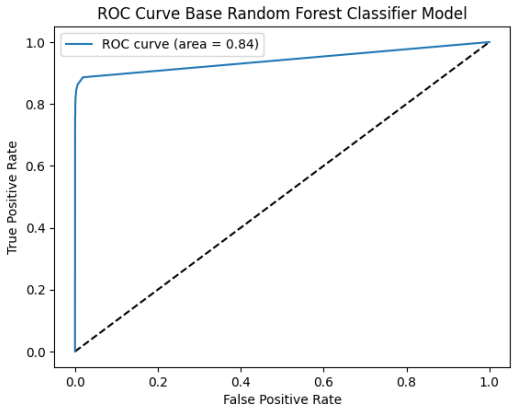

Figure 5.15: ROC_AUC curve of Base Random Forest Classifier

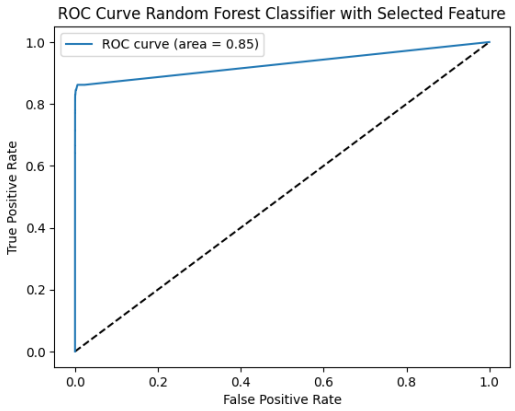

Figure 5.16: ROC_AUC curve of Reduced Random Forest Classifier

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC-AUC**
Base Randon Forest Classifier	0.9992	0.8736	0.6747	0.7614	0.8373
Reduced Random Forest Classifier	0.9993	0.9255	0.7073	0.8018	0.8536

Table 5.5: Evaluation Measures for Random Forest Classifier

Reduced Model

Best Parameters: max_features=5, n_estimators=110

The random forest algorithm was trained using the most important 25 selected features, and its performance was compared. As a result, the number of decision tree weak learners was increased to 110 using hyperparameter tuning by 5-fold cross-validation. The algorithm's performance was enhanced by increasing the number of weak learners and reducing the number of features. It resulted in a boost in the f1 score to 0.8018, an increase in recall to 0.7073, an increase of approximately 3% in correctly identifying fraudulent transactions in the test dataset, and an improvement in precision to 0.9255, which increased its ability to reduce false negative predictions. Additionally, the ROC_AUC score saw an increase to 0.8536.
Table 5.5 presents the evaluation measure of both models; comparing both, we can observe that the model with the selected feature outperforms base model in all the measures. Therefore, it is selected for further comparison.

5.6 AdaBoost

Base Model

Best Parameters: learning_rate=0.1, n_estimators=350

By 5-fold random search cross-validation 350 tree ensemble model and with a learnig rate of 0.1 were selected for the optimal performance of the classifier. The bosting algorithm achieved an f1 score of 0.7555 which represent a decent balance between precision- recall score with a precision score of 0.8333 and a recall score of 0.6910, representing around 70% of the fraudulent transactions in the test dataset were correctly classified. Figure 5.17 shows a ROC_AUC score of 0.8454 for the algorithm, which is good as it shows the model is able to differentiate positive and negative classes very well.

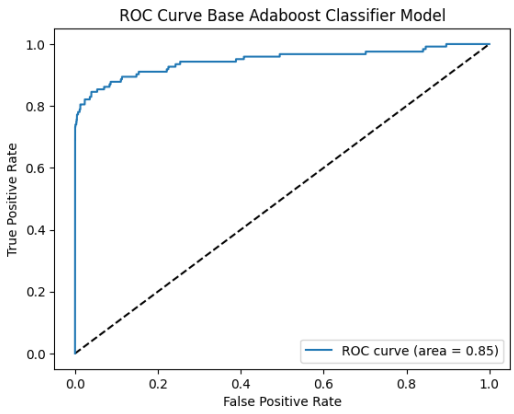

Figure 5.17: ROC_AUC curve of Base AdaBoost Classifier

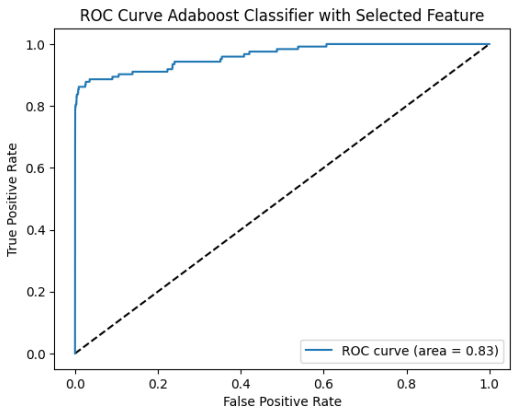

Figure 5.18: ROC_AUC curve of Reduced AdaBoost Classifier

Reduced Model

The AdaBoost classifier algorithm with the same hyperparameters was trained on 26 most import features selected by RFE, and its performance was compared. The algorithm's performance declined because of the decrease in the features of all its evaluation measures precision, recall, f1-score, and ROC_AUC score decreased to 0.8265, 0.6585, 0.7330, 0.8291, respectively, but the accuracy remained almost the same, but it is because of an imbalance of the class in the dataset.

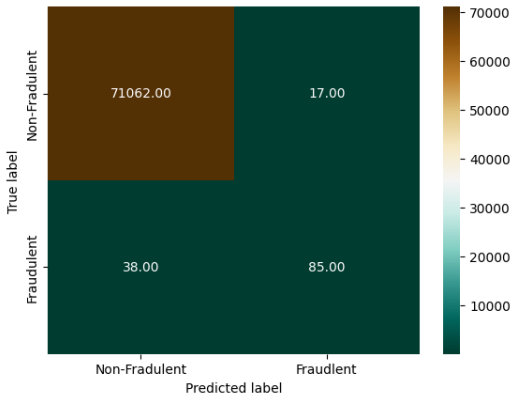

Figure 5.19: Confusion Matrix of AdaBoost Classifier

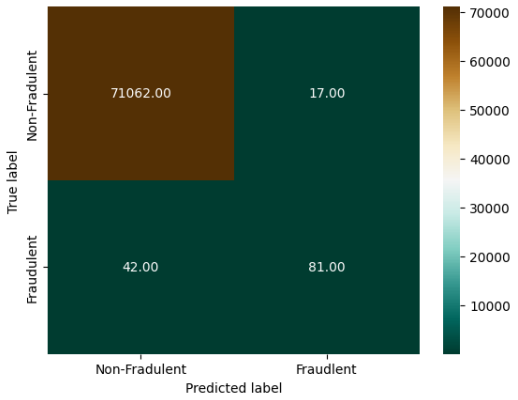

Figure 5.20: Confusion Matrix of AdaBoost Classifier

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC-AUC**
Base AdaBoost Classifier	0.9992	0.8333	0.6910	0.7555	0.8454
Reduced AdaBoost Classifier	0.9993	0.8265	0.6585	0.7330	0.8291

Table 5.6: Evaluation Measures for AdaBoost Classifier

Table 5.6 presents the comparison of evaluation measures between AdaBoost classifier with all features and with reduced features. Its ability to detect fraudulent transactions decreased by around 4% in the reduced model, with only 65.85% cases detected from a total of 123 positive cases in the test dataset. Therefore, the base AdaBoost model is chosen for further comparison.

5.7 Support Vector Machine

Base Model Best Parameters: C=0.1, class_weight='balanced', gamma=0.001, kernel='linear'

Optimal selection of hyperparameters is essential for the support vector machine classifier to select the hyperplane that divides the classes. By employing 5-fold cross-validation, the optimal cost function is identified as 0.1. It's important to note that selecting a larger cost function value can cause overfitting of the training data, while a lower value can lead to underfitting of the model. Additionally, the gamma value of 0.001 is chosen to control the shape of the decision boundary.

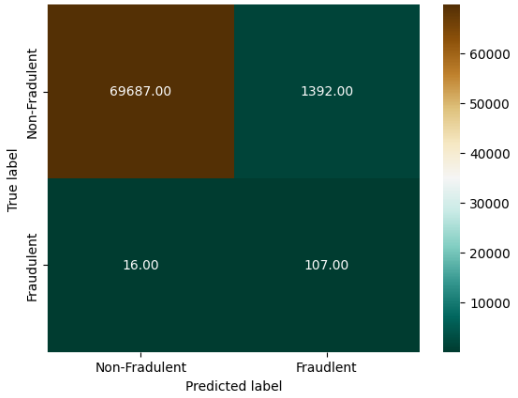

Figure 5.21: Confusion Matrix of Base SVM Classifier

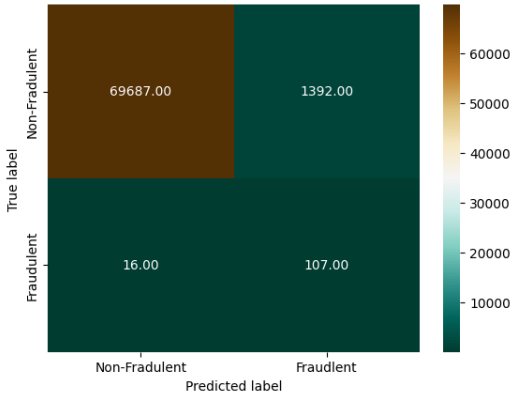

Figure 5.22: Confusion Matrix of Reduced SVM Classifier

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC-AUC**
Base SVM Classifier	0.9802	0.0713	0.8699	0.1319	0.9251
Reduced SVM Classifier	0.9588	0.0348	0.8536	0.0669	0.9063

Table 5.7: Evaluation Measures for AdaBoost Classifier

Despite achieving a high recall score of 0.8699, the precision of the model was very low at 0.07133, resulting in a poor f1 score of 0.1318. This indicates that out of all the fraud predictions made by the model, only 7% were actually fraud, while the rest were false positives. However, the model did achieve a high ROC_AUC of 0.8291. Based on the outcome obtained from SVM, it is clear that it is necessary to use a combination of all the evaluation measures used in the study to assess the predictive capability of a classifier.

Reduced Model

Best Parameters: C=10, class_weight='balanced', gamma=0.001, kernel='linear'

After selecting the top 10 feature using rfe and hyperparameter tunning, the gamma value of 0.001 and C value of 10 is selected as optimal hyperparmeters. The results show an even further decline in the f1 score to 0.0669 of the model. This is because of the low precision value of 0.0348, which represents a high number of false positives.

By comparing the base model with the reduced model we can choose the base model as the better performing model on the dataset.

5.8 Artificial Neural Network

ANN is a versatile and robust algorithm for the classification problem; Hyperparameter selected are 3 hidden layers with the activation function of rectified linear unit (relu) for non-linearity in the network, and output layer 'sigmoid' function for binary classification, the learning rate of 0.001 used to in gradient descent to update the weight and the batch size of 6.

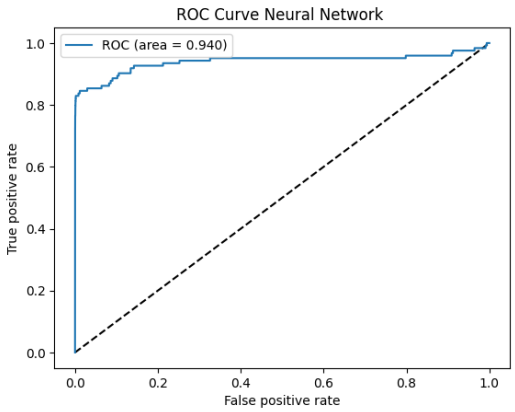

Figure 5.23: ROC_AUC curve of Artificial neural network

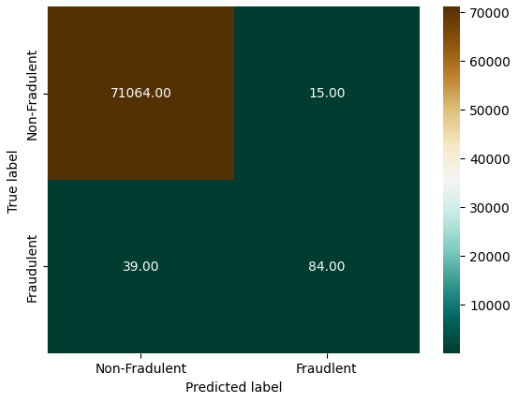

Figure 5.24: Confusion matrix curve of Artificial neural network

ANN model achieved an f1 score of 0.7567, with a good precision score of 0.8484 and a recall score of 0.6829; model was able to correctly predict 84 fraud out of a total 123 fraudulent transactions in the testing data. With a 0.940 ROC_AUC score model has efficiently been able to generalise positive and negative classes.

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC-AUC**
Artificial Neural Network	0.9992	0.8484	0.6829	0.7567	0.9401

Table 5.8: Evaluation Measures for Artificial Neural Network Classifier

5.9 Evaluation of Classifiers

A total of eight supervised data mining methods are utilised in detecting fraudulent transactions in the given dataset. The evaluation metrics for all of the classifiers used are presented in Table 5.9. Out of a total of 213605 observations, 25% of the data was allocated for testing

purposes, with both the training and testing datasets maintaining the same ratio of fraudulent transactions. The test data set consisted of 71202 transactions, 123 of which were fraudulent.

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC-AUC**
Logistic Regression	0.9991	0.8588	0.5934	0.7019	0.7966
Decision Tree Classifier	0.9992	0.8736	0.6747	0.7614	0.8373
K-Nearest Neighbors Classifier	0.9994	0.9450	0.6991	0.8037	0.8495
Naive Bayes Classifier	0.9950	0.1463	0.3902	0.2128	0.6931
Random Forest Classifier	0.9993	0.9255	0.7073	0.8018	0.8536
AdaBoost Classifier	0.9992	0.8333	0.6910	0.7555	0.8454
SVM Classifier	0.9802	0.0713	0.8699	0.1319	0.9251
Artificial Neural Network	0.9992	0.8484	0.6829	0.7567	0.9401

Table 5.9: Final Evaluation Table of all Classifiers

When it comes to classification algorithms, accuracy can't always be trusted due to data imbalance. Instead, ROC_AUC offers a superior generalisation of class labels in the model, which is why it was chosen as the preferred scoring method for algorithms. Using a combination of the f1 score and ROC_AUC provides more effective evaluation criteria for classification algorithms.

Based on the analysis, it was found that among the different algorithms used, Decision Tree, K-Nearest Neighbors, and Random Forest performed better than others across all measures, While SVM classifier had the best recall score, indicating that it predicted the most true positive cases, it also had a precision score of only 0.07, which means it misclassified many non-fraudulent transactions as fraud. Random Forest provided the best recall score of 0.70 and a precision score of 0.92. This means that it not only had a correct fraud detection rate of 70% but also had only 8% false positive cases.

Even though K-Nearest Neighbors relies on distance calculations, it does an exceptional job at identifying fraudulent activity with an impressive f1 score of 0.8037. Its recall and precision scores are also noteworthy at 0.6991 and 0.9450, respectively. Decision Tree, on the other hand, ranks third in performance with precision and recall scores of 0.8736 and 0.6747. Additionally, it boasts a high ROC_AUC score of 0.8373. Resampling methods are utilised

further to enhance the performance of these top three algorithms.

5.10 Evaluation of Resampling Techniques

In this section, resampling techniques were applied to the top three performing algorithms to achieve the fifth objective of the study. The evaluation scores of the selected algorithms with three different resampling techniques are shown in the table 5.10.

Classifier	Resamplin Method	Accuracy	Precision	Recall	F1 Score	ROC-AUC
Decision Tree	Without Resampling	0.9992	0.8736	0.6747	0.7614	0.8373
	Random Undersampling	0.8955	0.0143	0.8780	0.0282	0.8868
	Random Oversampling	**0.8955**	**0.7542**	**0.7235**	**0.7385**	**0.8615**
	SMOTE	0.8955	0.1646	0.7642	0.2708	0.8787
K-Nearest Neighbors	Without Resampling	0.9994	0.9450	0.6991	0.8037	0.8495
	Random Undersampling	0.9745	0.0561	0.8699	0.1055	0.9223
	Random Oversampling	**0.9987**	**0.6163**	**0.7967**	**0.6950**	**0.8979**
	SMOTE	0.9970	0.3470	0.8211	0.4879	0.9092
Random Forest	Without resampling	0.9993	0.9255	0.7073	0.8018	0.8536
	Random Undersampling	0.9720	0.0514	0.8699	0.0970	0.9210
	Random Oversampling	**0.9994**	**0.9484**	**0.7479**	**0.8363**	**0.8739**
	SMOTE	0.9993	0.8773	0.7560	0.8122	0.8779

Table 5.10: Evaluation Measures with Resampling Technique.

In the decision tree classifier model, we can observe that the model without any sampling technique performed best in terms of accurately identifying 87.36% of the predicted frauds with fewer false positives. On the other hand, the precision value of 0.7542 is also excellent for the oversampling technique. Regarding recall, undersampling had the best performance with a value of 0.8780, which means that it correctly identified 108 fraudulent transactions out of 123 in the test data. However, its precision value of 0.0143 was very low, indicating a high presence of false positives; it is because of the loss of information caused by discarding majority class observations. Although SMOTE had a recall score of 0.7642, it still struggled to accurately identify frauds in predicted fraud cases with a precison score of only 0.1646. Considering all evaluation measures and taking a balanced approach, the decision tree classifier with oversampling is considered the best result.

In K-nearest neighbors algorithm, the best-resampled method also came out to be oversampling, which resulted in an increase of approximately 10% in terms of recall. It achieved a recall value of 0.7967, correctly predicting 98 fraud transactions out of 123 available. Despite a lower f1 score than that of the kNN model that did not use any resampled data, it performed better in generalising positives and negatives in the dataset with a ROC_AUC score 0.8979. SMOTE and undersampled resampled algorithms showed an improvement in recall value; their precision values were relatively low, leading to a high number of false positive cases. This could cause dissatisfaction among credit card consumers.

After comparing the performance of the random forest model on the original dataset and the resampled dataset, it was found that the oversampled data produced the best overall results. This dataset was able to identify 74.79% of the fraudulent transactions in the dataset, which is lower than captured by algorithm with SMOTE oversampling, The model's greatest strength was its ability in minimizing false positives with a precison score of 0.9484, out of 97 fraud predictions made by the model, 92 were accurate. In general, the random forest model performed consistently well with all resampled data, with an F1 score greater than 0.800.
In conclusion of this section we can observe that all the algorithm using oversampling techniques presents best balanced results, it is because overssampling prevent information loss by retaing all the observations in the dataset and reducing model bias toward the majority class.

5.11 Final Model Selection

Based on the findings from the above section, it can be inferred that the Random Forest algorithm, utilizing oversampled data, was the most effective among the various algorithms evaluated in this study, with a f1 score of 0.8363, it demonstrated a precise balance between precision and recall scores, accurately detecting true positive cases of fraud while minimizing false positives and true negatives. The results perfectly show the efficiency of ensemble learning in improving the classification algorithms.
Therefore, this study recommends random forest combined with random oversampling as the optimal choice for classifying fraudulent transactions in a credit card fraud detection system.

CHAPTER 6

Conclusions and Future Works

In this study, we explored the problem of credit card fraud and its implications on the financial sector. By analysing the existing fraud detection system and conducting research on preventative measures, the study aimed to improve the effectiveness of the system. The study also compared various data-driven technologies and their potential to enhance the performance of a fraud detection system.

To utilise these data mining techniques aopen source European credit card transaction data provided by Worldline and the Machine Learning Group of ULB (Universitè Libre de Bruxelles) is used; the dataset contains 284,807 transactions which are highly imbalanced with only 0.172% fraudulent transactions. By analysing extensive transaction data and the use of advanced data-driven techniques, valuable insights were gained that would aid in the fight against fraudulent credit card use. A total of eight machine learning and deep learning algorithms were implemented in the dataset and their performance was measured and compared on multiple evaluation metrics to find the best algorithm suited to detect the fraudulent transaction. In datasets with imbalanced classes, it can be difficult for a classifier to accurately predict the correct class due to the bias towards the majority class. To overcome this challenge, data-level resampling methods were employed.

After evaluating different techniques, the study proposes a random forest model with oversampled data to be incorporated into a fraud detection system. It demonstrated the

best results in identifying the fraudulent transactions in the given dataset. It was able to generalise the data best and with fewer false positives, leading to fewer transactions being wrongly classified as fraud. Additionally, the ensemble learning approach employed by the random forest classifier helped to minimise overfitting by combining various models and effectively balancing the bias-variance tradeoff.

The proposed method combined with the ruled based system, the efficiency of a fraud detection system can be enhanced, leading to a reduction in financial losses and an increase in consumer trust in the security of credit cards as a preferred financial tool.

Despite the suggested approach proves to be highly effective at predicting instances of fraud in the provided data, there are some limitations of the proposed method. The main limitation arises due to data availability and quality. The study utilized data that had already undergone principal component analysis and had unlabeled features. Therefore, it was not feasible for us to identify the most significant feature that impacts fraud in real-life situations. One of the study's drawbacks is that it utilized data from only European customers, but in cases where user data of another demographic region is used for the model the results might differ.

Fraudsters are actively adapting to detection systems available in open source and coming up with new innovative ways to evade the system. In order to stay ahead of scammers, it is important to update and incorporate new technologies constantly. A hybrid approach using a combination of various algorithms on a significant amount of transaction data using balancing techniques could yield better classification results than the suggested model. The availability of a large, reliable dataset is at the core of any data-driven model, by utilizing multiple demographics datasets in the modelling process can also contribute to developing a robust fraud detection system that can be implemented by financial institutions worldwide.

Bibliography

[1] "Digital Payments - Worldwide | Statista Market Forecast," *Statista*, 2022. Available: https://www.statista.com/outlook/dmo/fintech/digital-payments/worldwide, Accessed 2 Aug. 2023.

[2] UK Finance,"UK PAYMENT MARKETS SUMMARY 2022," 2022, Available:https://www.ukfinance.org.uk/system/files/2022-08/UKF%20Payment%20Markets%20Summary%202022.pdf, Accessed 2 Aug. 2023.

[3] C. Mullen, "Card industry faces $400B in fraud losses over next decade, Nilson says," Payments Dive, Dec. 14, 2021. Available: https://www.paymentsdive.com/news/card-industry-faces-400b-in-fraud-losses-over-next-decade-nilson-says/611521/, Accessed 2 Aug. 2023.

[4] S. Haqqi, "Credit Card Statistics 2022 | money.co.uk," www.money.co.uk, Feb. 01, 2023." Available: https://www.money.co.uk/credit-cards/credit-card-statistics, Accessed 2 Aug. 2023.

[5] "Fight card-not-present fraud (2021 solutions)," Thalesgroup.com, 2021. 2023." Available: https://www.thalesgroup.com/en/markets/digital-identity-and-security/banking-payment/digital-banking/dcv, Accessed 1 Aug. 2023.

[6] F. Carcillo, Y.A. Borgne, O. Caelen, Y. Kessaci, F. Oble, G. Bontempi, Combining unsupervised and supervised learning in credit card fraud detection, 2021 Information Sciences, Volume 557, Pages 317-331, ISSN 0020-0255,

[7] S. Khatri, A. Arora and A. P. Agrawal, "Supervised Machine Learning Algorithms for Credit Card Fraud Detection: A Comparison," 2020, 10th International Conference on

Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2020, pp. 680-683.

[8] H. Naik, & P. Kanikar, Credit card Fraud Detection based on Machine Learning Algorithms. 2019, International Journal of Computer Applications.

[9] N. Khanna, "J48 Classification (C4.5 Algorithm) in a Nutshell," Medium, Aug. 18, 2021." Available: https://medium.com/@nilimakhanna1/j48-classification-c4-5-algorithm-in-a-nutshell-24c50d20658e, Accessed 1 Aug. 2023.

[10] Jain, Y. & Tiwari, N. & Dubey, S. & Jain, Sarika. (2019). A comparative analysis of various credit card fraud detection techniques. International Journal of Recent Technology and Engineering. 7. 402-407.

[11] Vijay Kotu, Bala Deshpande, Chapter 2 - Data Mining Process, Editor(s): Vijay Kotu, Bala Deshpande, Predictive Analytics and Data Mining, Morgan Kaufmann, 2015, Pages 17-36, ISBN 9780128014608,

[12] E. Esenogho, I. D. Mienye, T. G. Swart, K. Aruleba, and G. Obaido, "A Neural Network Ensemble With Feature Engineering for Improved Credit Card Fraud Detection," IEEE Access, vol. 10, 2022.

[13] L. Chen, "Basic Ensemble Learning (Random Forest, AdaBoost, Gradient Boosting)- Step by Step Explained," Towards Data Science, Jan. 02, 2019. Available: https://towardsdatascience.com/basic-ensemble-learning-random-forest-adaboost-gradient-boosting-step-
Accessed 1 Aug. 2023.

[14] Chen, Joy Iong-Zong and Kong-Long Lai. "Deep Convolution Neural Network Model for Credit-Card Fraud Detection and Alert." 2021

[15] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M.-S. Hacid, and H. Zeineddine, "An Experimental Study With Imbalanced Classification Approaches for Credit Card Fraud Detection," 2019 IEEE Access, vol. 10, 2022.

[16] M. Krikorian, "Fraud Detection applying Unsupervised Learning techniques," Medium, Jun. 29, 2021. Available: https://medium.com/southworks/fraud-detection-applying-unsupervised-learning-techniques-4ae6f71b266f Accessed 1 Aug. 2023.

[17] Bodepudi, Hariteja. Credit Card Fraud Detection Using Unsupervised Machine Learning Algorithms. 2021 International Journal of Computer Trends and Technology. 69. 1-3.

[18] A. K. Rai & R. K. Dwivedi, "Fraud Detection in Credit Card Data using Unsupervised Machine Learning Based Scheme," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2020, pp. 421-426.

[19] S. Jiang, R. Dong, J. Wang, & M. Xia, "Credit Card Fraud Detection Based on Unsupervised Attentional Anomaly Detection Network," Systems, vol. 11, no. 6, p. 305, Jun. 2023.

[20] Priya & K. Sharma, Fraud Detection Model Using Semi-supervised Learning, 2023 , In: Thakur. M., Agnihotri. S., Rajpurohit B.S., Pant M., Deep K., Nagar, A.K. (eds) Soft Computing for Problem Solving. Lecture Notes in Networks and Systems, vol 547. Springer, Singapore.

[21] Dzakiyullah. Nur, Andri Pramuntadi, & Anni Karimatul Fauziyyah. "Semi-Supervised Classification on Credit Card Fraud Detection using AutoEncoders." Journal of Applied Data Sciences [Online], 2.1 (2021): 01-07. Web. 7 Aug. 2023

[22] R. Inc, "4 Major Challenges facing Fraud Detection; Ways to Resolve Them using Machine Learning," Medium, Apr. 25, 2019. Available: https://medium.com/razorthink-ai/4-major-challenges-facing-fraud-detection-ways-to-resolve-them-using-m Accessed 1 Aug. 2023.

[23] A. Dal Pozzolo, R. Johnson, O. Caelen, S. Waterschoot, N. V. Chawla and G. Bontempi, "Using HDDT to avoid instances propagation in unbalanced and evolving data streams," 2014 International Joint Conference on Neural Networks (IJCNN), Beijing, China, 2014, pp. 588-594,

[24] Dal Pozzolo A, Boracchi G, Caelen O, Alippi C, Bontempi G., Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy. IEEE Trans Neural Netw Learn Syst. 2018 Aug;29(8):3784-3797

[25] Andrea Dal Pozzolo, & Gianluca Bontemp *"Adaptive Machine Learning for Credit Card Fraud Detection.*, 2015.

[26] R.J. Bolton & D.J. Hand. " Unsupervised Profiling Methods for Fraud Detection," 2002.

[27] S. Bhattacharyya & S. Jha & K. Tharakunnel & J.C. Westland, "Data mining for credit card fraud: A comparative study, " Decision Support Systems, 2011.

[28] "Artificial Neural Networks and its Applications," GeeksforGeeks, Jun. 24, 2020. Available: https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/, Accessed 2 Aug. 2023.

[29] I. D. Mienye and Y. Sun, "A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects, " vol. 10, pp. 99129-99149, 2022.

[30] A. Biswal, "What is Bagging in Machine Learning And How to Perform Bagging" Simplilearn.com, Jul. 22, 2021. Available: https://www.simplilearn.com/tutorials/machine-learning-tutorial/bagging-in-machine-learning, Accessed 2 Aug. 2023.

[31] "Boosting in Machine Learning | Boosting and AdaBoost,"GeeksforGeeks, May 03, 2019. Available: https://www.geeksforgeeks.org/boosting-in-machine-learning-boosting-and-adaboost/, Accessed 2 Aug. 2023.

[32] "The logistic sigmoid function." Wikimedia Commons, Jul. 02, 2008. Available: https://commons.wikimedia.org/wiki/File:Logistic-curve.svg, Accessed 2 Aug. 2023.

[33] "Decision Tree Classifier | Note of Thi," dinhanhthi.com. Available: https://dinhanhthi.com/decision-tree-classifier/, Accessed 2 Aug. 2023.

[34] A.R. Deepthi "KNN visualization in just 13 lines of code," Medium, Sep. 24, 2019. Available: https://towardsdatascience.com/knn-visualization-in-just-13-lines-of-code-32820d72c6b6, Accessed 2 Aug. 2023.

[35] "Random Forest Classifier Tutorial: How to Use Tree-Based Algorithms for Machine Learning," freeCodeCamp.org, Aug. 06, 2020. Available: https://www.freecodecamp.org/news/how-to-use-the-tree-based-algorithm-for-machine-learning/, Accessed 2 Aug. 2023.

[36] V. Alto, "Understanding AdaBoost for Decision Tree," Medium, Jan. 31, 2020. Available: https://towardsdatascience.com/understanding-adaboost-for-decision-tree-ff8f07d2851, Accessed 2 Aug. 2023.

[37] "Scikit-learn SVM Tutorial with Python (Support Vector Machines)," www.datacamp.com Available: https://www.datacamp.com/tutorial/svm-classification-scikit-learn-python, Accessed 2 Aug. 2023.

[38] "Receiver operating characteristic," Wikipedia, Mar. 15, 2023. Available: https://en.wikipedia.org/wiki/Receiver_operating_characteristic#/media/File:Roc_curve.svg, Accessed 2 Aug. 2023.

[39] "Credit Card Fraud Detection," www.kaggle.com, 2018. Available: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud, Accessed 2 Aug. 2023.

[40] A. D. Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating Probability with Undersampling for Unbalanced Classification, " 2015 IEEE Symposium Series on Computational Intelligence, Dec. 2015, Available: https://doi.org/10.1109/ssci.2015.33,

[41] SciKit-Learn, "3.1. Cross-validation: evaluating estimator performance, scikit-learn 0.21.3 documentation," Scikit-learn.org, 2009. Available: https://scikit-learn.org/stable/modules/cross_validation.html, Accessed 2 Aug. 2023.