

Interpretable Credit Card Fraud Detection Using Deep Learning Leveraging XAI

by

Joy Biswas
18301180

Md. Tahfimuazzaman
18301086

Abir Ahmed Mridha
18301178

Tamanna Afroz
18301153

Ahnaf Tahmid Samin
18301111

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
September 2022

© 2022. Brac University
All rights reserved.

Declaration

It is hereby declared that our submitted thesis is our own authentic work while completing the Bachelor Degree in Computer Science and Engineering at Brac University. Except as properly referenced, there are not any previously published or submitted works included in this thesis.

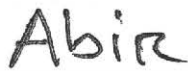
Student's Full Name & Signature:



Joy Biswas
18301180



Md. Tahfimuazzaman
18301086



Abir Ahmed Mridha
18301178



Tamanna Afroz
18301153



Ahnaf Tahmid Samin
18301111

Approval

The thesis titled “Interpretable Credit Card Fraud Detection Using Deep Learning Leveraging XAI” submitted by

1. Joy Biswas(18301180)
2. Md. Tahfimuazzaman(18301086)
3. Abir Ahmed Mridha(18301178)
4. Tamanna Afroz(18301153)
5. Ahnaf Tahmid Samin(18301111)

Of Academic Year Summer 2022 has been accepted as satisfactory in partial fulfillment of the requirements for the Bachelor Degree of Computer Science and Engineering.

Examining Committee:

Supervisor:
(Member)



Dr. Muhammad Iqbal Hossain
Associate Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)



Jumana
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Due to the internet's widespread accessibility, more and more businesses are bringing their offerings online. Besides, because of the growth of E-commerce websites, both individuals and businesses that deal in finances are more dependent on internet administrations to handle their business. Since more and more people are using online banking and making purchases online, credit card fraud has increased. Fraudsters can also use anything to disrupt the existing fraud detection system's systematic operation. As a result, we took on the issue of improving the existing fraud detection system to the highest possible level. This research seeks to develop an efficient fraud detection system by utilizing deep learning (DL) as well as the machine learning methods that are responsive to shifting patterns of customer behavior and have a tendency to reduce fraud manipulation through the identification and filtering of fraudulent activity in real time. The techniques in our research include Artificial Neural Network, Convolutional Neural Network, Recurrent Neural Network, Logistic Regression, K-Nearest Neighbor, Naive Bayes, Meta-Learning, and Explainable Artificial Intelligence (XAI). This research suggests that the K-Nearest Neighbor is the most effective algorithm with an accuracy of 99.75% among many others.

Keywords: Artificial Neural Network, Convolutional Neural Network, Recurrent Neural Network, Logistic Regression, K-Nearest Neighbor, Naive Bayes, Meta-Learning, Explainable AI.

Dedication

We would like to dedicate this thesis to our supportive parents as well as the respected faculty members in our department who have provided us with support and guidance in a variety of different ways during this entire process.

Acknowledgement

We want to start by giving thanks to Allah, without the grace of Allah, we could not have finished our thesis.

Second, we would like to thank our supervisor, Dr. Muhammad Iqbal Hossain sir. Sir helped us a lot regarding our thesis. Sir was there for us whenever we required assistance.

Last but not the least, our parents, without whose unwavering support none of this would have been possible. We are very grateful for their continued encouragement and prayers as we approach our graduation.

Table of Contents

Declaration	i
Approval	ii
Abstract	iv
Dedication	v
Acknowledgment	vi
Table of Contents	vii
List of Figures	ix
List of Tables	x
Nomenclature	xi
1 Introduction	1
1.1 Motivation	1
1.2 Problem Statement	2
1.3 Research Objectives and Contributions	2
1.4 Thesis Structure	3
2 Background	4
2.1 Literature Review	4
2.2 Algorithms Description	6
2.2.1 Artificial Neural Network	6
2.2.2 Convolutional Neural Network	6
2.2.3 Recurrent Neural Network	7
2.2.4 Logistic regression	8
2.2.5 K-Nearest Neighbor	9
2.2.6 Naive Bayes	10
2.2.7 Meta-Learning	11
2.3 Technique Description	12
2.3.1 XAI	12
3 Proposed Model	13
3.1 The Description of Dataset	13
3.1.1 Data Manipulation	13

3.1.2	Dataset Preprocessing	14
3.1.3	Features of the dataset	14
3.2	Model Description	15
4	Experimentation	17
4.1	Simulator	17
4.2	The Metrics of Data Analysis	17
4.2.1	Confusion Matrix	17
4.2.2	Precision, Recall, Accuracy & F1-Score	18
5	Result Analysis Discussion	19
5.1	Full Dataset	19
5.2	Balanced Dataset	20
5.3	ROC CURVE ANALYSIS	22
5.4	Confusion Matrix Analysis	26
5.5	XAI Analysis (SHAP)	28
5.6	Comparison	31
5.7	Discussion	31
6	Conclusion	33
6.1	Conclusion & Future work	33

List of Figures

2.2.1 ANN Model	7
2.2.2 CNN Model	7
2.2.3 RNN Model	8
2.2.4 Logistic Regression Model	9
2.2.5 KNN Model	10
2.2.6 Naive Bayes Model	11
2.2.7 Meta-Learning Model	11
3.2.1 The working mechanism of proposed model	16
5.1.1 Description of the full dataset	19
5.2.1 Description of the balanced dataset	21
5.3.1 ROC for ANN	23
5.3.2 ROC for RNN	23
5.3.3 ROC for CNN	24
5.3.4 ROC for Logistic Regression	24
5.3.5 ROC for Naive Bayes	25
5.3.6 ROC for KNN	25
5.3.7 ROC for Meta-Learning	26
5.4.1 KNN Confusion Matrix	27
5.4.2 ANN Confusion Matrix	27
5.4.3 RNN Confusion Matrix	27
5.5.1 XAI (SHAP) for KNN	28
5.5.2 XAI (SHAP) for Logistic Regression	29
5.5.3 XAI (SHAP) for Naive Bayes	29
5.5.4 XAI (SHAP) for ANN	29
5.5.5 XAI (SHAP) for CNN	30
5.5.6 XAI (SHAP) for RNN	30

List of Tables

3.1.1 Details about the features	15
5.1.1 Description of the full dataset	19
5.1.2 Performance Metrics of Full Dataset	20
5.2.1 Description of the balanced dataset	21
5.2.2 Performance Metrics of Balanced Dataset	22
5.6.1 Comparison between relevant papers and our model	31

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

<i>AE</i>	Autoencoders
<i>ANN</i>	Artificial Neural Network
<i>AUC</i>	Area Under the Curve
<i>CART</i>	Classification And Regression Trees
<i>CNN</i>	Convolutional Neural Network
<i>DAG</i>	Directed Acyclic Graph
<i>DL</i>	Deep Learning
<i>FN</i>	False Negative
<i>FP</i>	False Positive
<i>GE</i>	Genetic Algorithm
<i>IDE</i>	Integrated Development Environment
<i>KNN</i>	K-Nearest Neighbor
<i>ML</i>	Machine Learning
<i>RNN</i>	Recurrent Neural Network
<i>ROC</i>	Receiver Operating Characteristic
<i>SHAP</i>	SHapley Additive exPlanations
<i>SVM</i>	Support Vector Machine
<i>TN</i>	True Negative
<i>TP</i>	True Positive
<i>XAI</i>	Explainable Artificial Intelligence

Chapter 1

Introduction

1.1 Motivation

Electronic commerce has been applied by the majority of businesses, organizations, and government agencies in order to increase their levels of working. Electronic commerce platforms are increasingly vulnerable to large-scale fraud because of the fact that they are used by both legitimate customers and dishonest individuals. A crime that is performed with the intention of acquiring money through deception is referred to as fraud.

The purpose of a fraud control system is to protect sophisticated technology from fraud by avoiding its occurrence. However, this strategy is inadequate to avoid fraud. Detection of fraud is frequently advised as a means of enhancing a system's security. Here, the detection of credit card fraud identifies the fraudulent transactions and notifies the system administrator. Besides, credit card fraud detection by a machine or system is a tough phenomena. A system must be intensively trained with relevant data in order to achieve the process of detecting fraud. Moreover, deep learning and machine learning are the best approaches to solve these kinds of issues nowadays. The term "Deep Learning" (DL) refers to a specific type of machine learning that makes use of Artificial Neural Networks with several processing layers to extract more complex characteristics from raw input. For the following reasons, we have moved to both deep learning-based and machine learning-based algorithms for detecting credit card fraud:

- It will give higher accuracy of detecting frauds
- It will continuously processes and analyze new data
- Manual work is less needed here
- It has the ability to identify new patterns.
- It has the adaptability to change easily with the new environment.

Indeed, "Deep learning" is an artificial intelligence (AI) and machine learning (ML) technology that aims to imitate how people learn certain knowledge. Deep learning is crucial in data science, which also includes areas like statistics and predictive modeling. Since deep learning and machine learning improves and speeds the processes

involved in obtaining, analyzing, and interpreting enormous volumes of information, it is a valuable tool for data scientists. In contrast, not all datasets are precise. Data processing is thus an important factor in both deep learning and machine learning methods. Obviously, processing a dataset will increase its use for deep learning and machine learning. In addition, data processing involves the creation of a suitable data collection mechanism. As stated prior, to find error-free or accurate data is quite impossible. Therefore, the dataset which we are going to use, has also faults that can be addressed. It is possible that we will require some form of data shaping and modifying for the dataset.

1.2 Problem Statement

There are a few difficulties that make it hard for researchers to take this fraud detection technology into implementation. One of the main issues is the lack of available test results and accurate data. The reason for this is the sensitive nature of the customer's financial information that is involved in the fraud and must remain completely confidential. Here, we present a list of requirements that should be met by every reliable fraud detection system:

- As credit card fraud accounts for such a small fraction of all transactions, the system should be able to accommodate non-normal distributions.
- The system should be provided with an appropriate method for dealing with the noise. The inaccuracies that are contained within the dataset, such as inaccurate dates, are what is referred to as "noise." Regardless of how extensive the training set is, the accuracy of the generalization that may be obtained is restricted by the noise that exists in the actual dataset.
- The presence of overlapping information is another drawback connected with this field. This means that some transactions may be mistakenly identified as phony while they are, in fact, legitimate, and vice versa.
- When new forms of fraud emerge, the system must be ready to handle them. Since successful fraud strategies become ineffective over time as they become standards, proficient fraudsters consistently look for new and efficient ways to do their activity.
- In order to evaluate the classifier system, precise measurements are required. Even with a high degree of precision, the majority of fraudulent transactions are frequently categorized as real. Thus, a skewed dispersion cannot be used as a measure of precision in general. Besides, both the financial loss from fraud and the cost of investigating it should be taken into account by the system. This should be done simultaneously.

1.3 Research Objectives and Contributions

Using supervised deep learning and machine learning methods, we plan to address the issue of identifying fraudulent credit card transactions. Therefore, we demonstrate a technique for feature extraction from datasets for the express purpose of

training models to identify fraudulent financial transactions. The objectives that our proposed system will fulfill are as follows:

- To effectively and accurately identify credit card fraud transactions.
- To construct a high-performing model by utilizing the aforementioned dataset's labels and features.
- To determine whether a specific transaction is fraudulent using the trained model.
- To provide tools for detecting fraudulent transactions while reducing expenses.

1.4 Thesis Structure

The following sections are included in our thesis. In Chapter 2, the background provides various related techniques researchers used to solve this similar problem, as well as the algorithm explanations of our thesis. Chapter 3 describes the dataset as well as the planned model for our thesis. We described our experiments in detail in Chapter 4. Moreover, Chapter 5 presents the experimental results and discussion. Finally, Chapter 6 offers a summary of the study, and Chapter 7 contains the references.

Chapter 2

Background

2.1 Literature Review

Engineers have been continuously looking for innovative solutions to provide more convenient, safe, and precise transactions in the field of Credit Card Fraudulence Discovery. This subject has become even more important in light of the recent development in machine learning and data science. Numerous significant research findings have been made in this area, serving as a foundation for ongoing and future research. Researchers have tried out different concepts and have worked with machine learning using different algorithms. As part of our background research, we came across studies in which tests were successfully carried out using a variety of machine learning (ML) and deep learning (DL) techniques.

In their paper, J. Galindo and P. Tamayo compared the effectiveness of KNN, Neural Networks, and CART model for analysis of credit risk using data on house mortgage loans supplied to them by Mexico's securities exchange commission. Each entry in the dataset—which had around 4,000 in total—represented a customer account having a total of 24 attributes. It just needed small data pre-processing which was necessary before they could begin using their preferred algorithms on it. Following the three chosen algorithms' predictions, they tabulated and graphically represented the findings and conducted a comparison. In comparison to the neural network and KNN model, it was observed that CART was the most accurate. However, it was also found that, to perform better the CART needs at least 22,000 entries. But it is fine when it comes to developing a risk prediction model for a company like CNBV [1].

It is important to separate the pertinent data and influencing aspects used in the risk calculation, as well as to choose the appropriate models for risk analysis, in order to effectively estimate risk while adhering to the essential criteria. Support Vector Machine, or SVM, would be a useful technique in such a circumstance, as stated in a work by Gudas, S., Garsva, G. and Danenas, P. [2]. The main benefit of SVM over other AI-based solutions is that the solution produced by it will not be under local minima. Finding the unique data points that will be utilized as the solution's support vectors is important in order to put this strategy into practice.

Addo P.M. has developed an alternative method to identify defaulters in this risk

analysis by introducing the Elastic Net algorithm [3]. This technique makes use of several extensions of linear regression. This technique has multinomial and logical functions as well as a strong error-checking system that improves accuracy while lowering error. Using an estimation process this approach provides information with a high degree of accuracy regarding loan repayment. Elastic net penalty is identified by constructing a graph in which x stands for predictors and y for response variables. Here, two elastic net algorithm equations are used. This has been carried out with the assistance of some additional algorithms, such as the gradient boosting machine and random forest modeling.

The Bayesian Classifier approach, where DAG (directed acyclic graph) strategy is used which helps to find loan repayment probability as per Pandey T.N. [4]. In this method, the nodes are random variables and the edges are dependencies of them. The accuracy is based on how well the datasets and dependencies are connected in the network. The Naive Bayesian classifier is another altered variation of Bayesian classifier where the dataset attributes are represented as independent variables requiring less datasets. Additionally, with a non parametric approach, KNN works with training sets having positive and negative cases. The testing and the training phase are the two divisions of the functionality. This approach computes the Euclidean Distance between the training points during the testing phase. The most equivalent instance is used as the output after producing instances using regression. The K-Means, support vector machine, multilayer perceptron etc are some further techniques used in this research.

In terms of developing a credit rating system, LS-SVM and Neural Network algorithms perform better, according to Baesens B. [5]. They adopted three different SVM implementation methodologies for credit rating in their study. Additionally, they used two UCI datasets to assess the SVM's accuracy. This method's accuracy is almost comparable to neural network and decision tree methods, having to require less input features. Additionally, the use of parameters can be reduced using genetic algorithms combined with SVM (GA-SVM).

A hybrid HGA-NN approach, which combines genetic algorithms and neural networks, was utilized in the study of Oreski G. [6]. For the first stage of feature ranking, they used a quick filter technique. Improvements are also made to the genetic algorithm's incremental step, which is used to construct the initial population. The accuracy and expandability of the credit risk assessment can both be improved by this hybrid method. Additionally, they tested the system using actual data from the Croatian Bank.

Analysis of credit risk is covered in a study by Abida, F, Aziz, R. S., Islam, S. A. and Ahmed, A. [7]. The main focus is on assigning each person a credit score that would indicate their creditworthiness. The Credit Score primarily plays a significant function to assess the risk of the loans given by banks or financial organizations for a person's personal or professional purposes. To develop a reliable software mechanism that assures that accurate calculation of credit scores and running without issue was the goal of this study. The conclusions depicted the process of precisely and accurately assessing an individual. The algorithm's accuracy ratings were discovered to

be lower than 90%. They used the CART model utilizing the Gradient Boosting Method (GBM) as well as XGBoost, which addresses a variety of issues, including user-defined and regression ranking difficulties. Moreover, a two-step architecture hybrid model was suggested in the paper itself which can be put into practice in order to offer a firm foundation to obtain the assessment of defaulters. An overview in details of conclusion and result was also given.

A variation of the logistic regression is demonstrated by Lawi, A., Syarif and Aziz. This model is Generalized Linear algorithm [8]. Independent variables are used to create a binary response by the classification process known as logistic regression. Confidence bound can be obtained with the help of Generalized Linear model algorithm along with a favorable result. Their model was shown to have a respectably high accuracy as well as a comparatively high prediction confidence.

Using Ensemble Logistic Regression on UCI repository's datasets of Australia and Germany Svrlaka, A and Nalic, J offered additional improvements, and they also achieved quite high accuracies where they also used with Gradient Boost [9] [10].

2.2 Algorithms Description

2.2.1 Artificial Neural Network

ANN refers to the Artificial Neural Network. Inspiration behind this algorithm is the biological nervous system. This algorithm mimics the biological nervous system. The calculation or storage units used in this algorithm are referred to as artificial neurons. Neurons in this algorithm generally have multiple inputs. When a neuron gets an input, it processes the input, performs some calculations and generates an output. This output is then passed on to the next neuron connected with this one as input. Not all inputs can activate a neuron. The inputs are fed into a transfer function that compares the inputs with a set threshold. The weights represent the strength of a connection. Some connections are strong and some are weak. Training the algorithm means to adjust the weights until we get the best prediction performance from the algorithm. The neural network must be aware of its distance from the actual target in order for the prediction to be correct. To do this a loss function is used. A loss function is used to quantify how far off the algorithm's output is from the true goal. Then the goal is to determine the appropriate weight parameters so that we can minimize the loss function. To get the desired outcome we adjust the weights parameters so that we can get predictions closer and closer to the actual target [11].

2.2.2 Convolutional Neural Network

CNN is short for Convolutional Neural Network. This algorithm uses convolutions which is a type of linear mathematical operation. We have created a 1D model of the dataset and run CNN on that. CNN can have many layers, each layer detects different features from a tabular data. On the training data, different filters are applied on different resolutions. Output from one layer is used as input for the

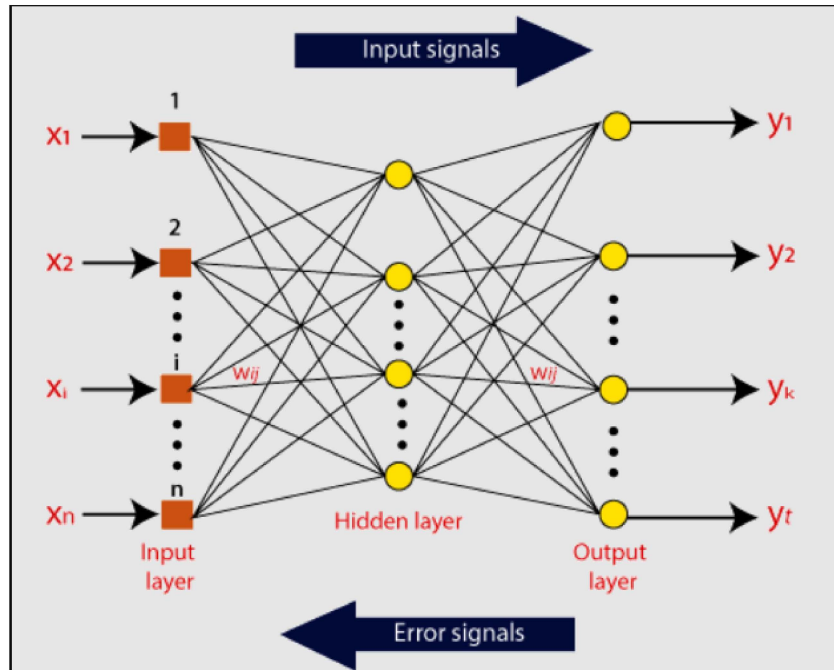


Figure 2.2.1: ANN Model

next layer. Filters can be the pattern of the data, those patterns of the data or some other thing that uniquely defines the data. CNN generally has three layers. They are pooling, convolution, and rectified linear units (ReLU). The convolution layer activates some unique features of data. Negative values are mapped to zero and the positive values are preserved by the ReLU layer. The subsequent layer receives the activated features. Pooling optimizes the output, conducts nonlinear downsampling, and lowers the amount of parameters the network needs. These processes are repeated many times to identify different features of data [12].

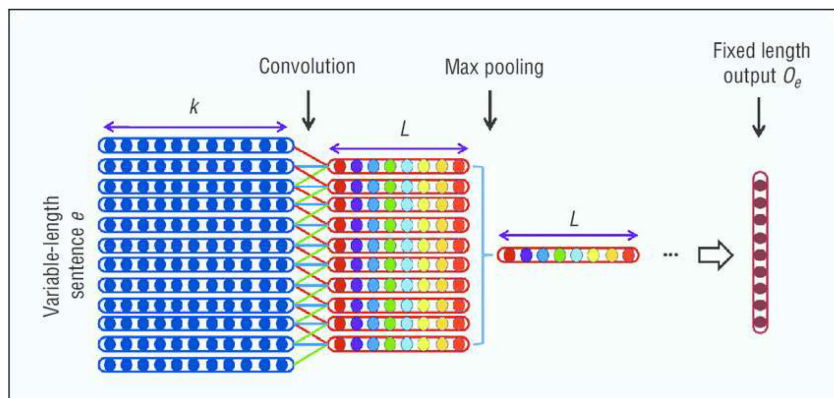


Figure 2.2.2: CNN Model

2.2.3 Recurrent Neural Network

RNN is short for Recurrent Neural Network. RNN can model sequential data for recognizing sequences and prediction. This algorithm is used for recognizing hand-writings, image to text and speech recognition. RNN is basically an ANN that has

recurrent neural connections. The neural network's complexity and power both rise in proportion to the number of recurrent connections it has. Different representations of data and different kinds of feature extraction are all within the capabilities of a sophisticated RNN. Higher layers of the neural network remove unwanted data. The recurrent connections of the RNN provide the ability to process, store and remember past signals for a long time. RNN is a supervised deep learning model. There are three layers in a basic RNN. Here, one is the input layer, the next one is the output layer, and the hidden layer with recurrent connections. A weighted matrix defines the connections between the input layer and the hidden layer. The input layer and the hidden layer are entirely connected here. How the small units initiate the hidden layer has a significant impact on the network's performance and stability. The weights are optimized and instantiated to decrease training loss of the RNN [13].

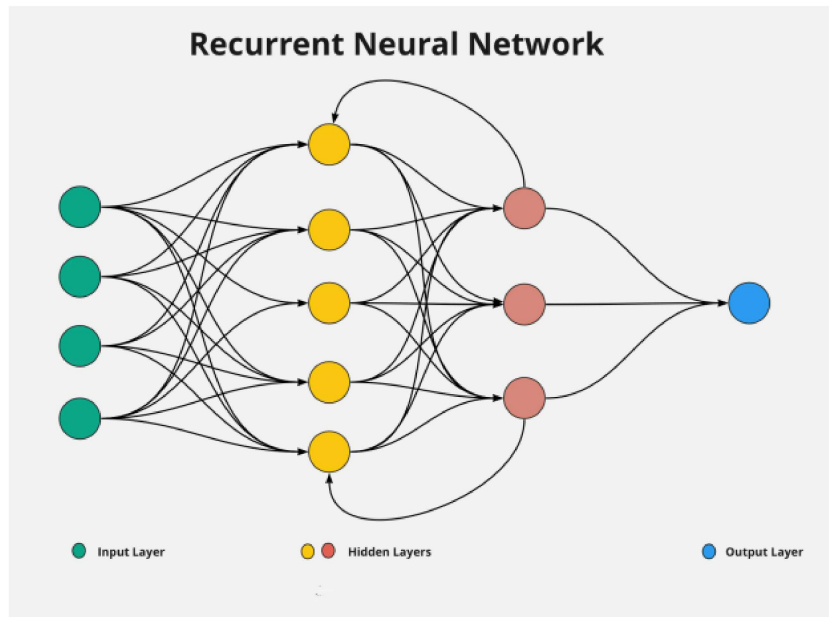


Figure 2.2.3: RNN Model

2.2.4 Logistic regression

The statistical field mostly uses the method of logistic regression. Logistic regression is used to resolve binary classification issues. In order to forecast the output value using this method, input values are linearly mixed with weights or coefficients. $y = \frac{e^{(b_0+b_1x)}}{1+e^{(b_0+b_1x)}}$ is the general equation for logistic regression. b_0 is the bias in this instance, b_1 is the coefficient, and y is the expected value of the output. Instead of producing a numeric value, a logistic regression produces a binary value, which can only be either 0 or 1. For predicting outcomes, trend forecasting, or causal analysis, logistic regression is utilized [14].

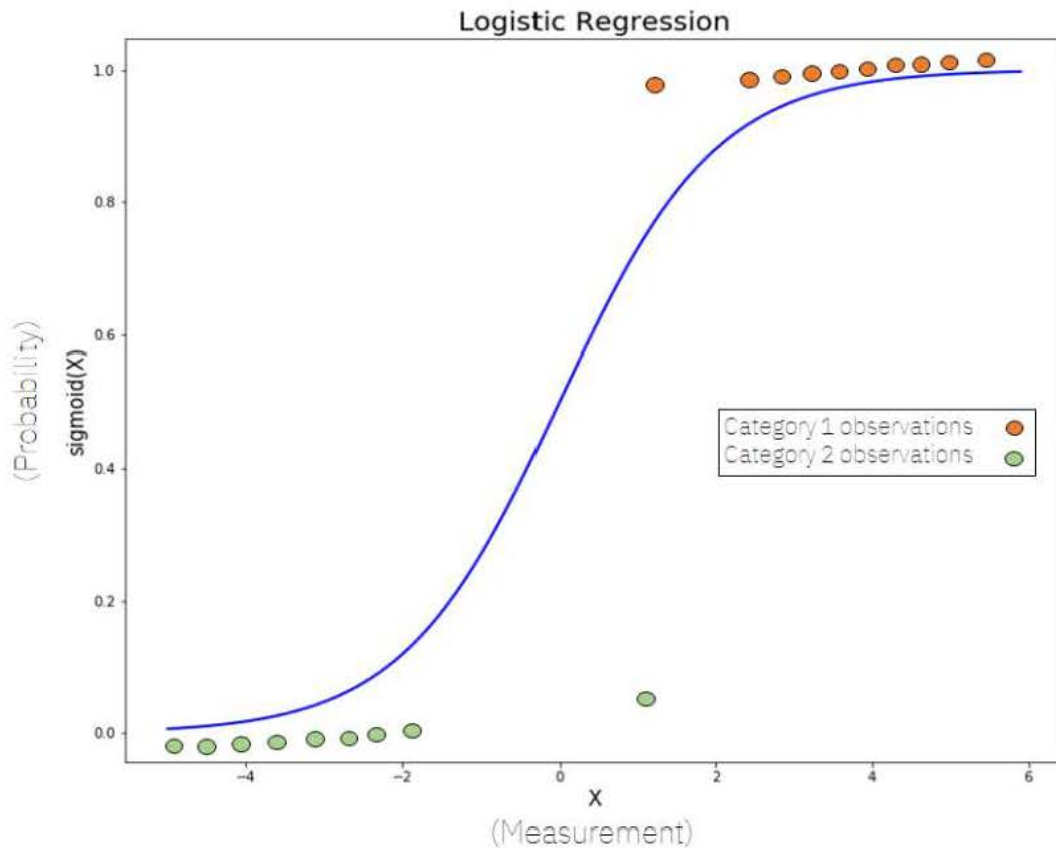


Figure 2.2.4: Logistic Regression Model

2.2.5 K-Nearest Neighbor

KNN stands for "K-Nearest Neighbor," an abbreviation of its longer form. Here, K means user input. This algorithm is frequently used to categorize data instances and predict group membership. Classification is a technique used in data mining. KNN algorithm classifies new instances based on similarity measures with other data instances or uses some distance measure like euclidean or cosine. KNN is a non-parametric algorithm. It's sometimes referred to as a lazy learner because it doesn't require any further input beyond the training set in order to make a classification. The output of KNN depends on the value of k. KNN classifies a data instance by comparing it with its k nearest neighbors. If a data instance has 3 neighbors around when the value of k is 4, and 2 out of the 3 is similar to the data instance, then KNN will classify the data instance with the 2 similar neighbors. But if the value of k is changed to 5 and there number of neighbors increases to 10, and the number of similar neighbors becomes 7 which are different from the previous neighbors chosen, then KNN will classify the data instance with the 7 similar neighbors [15].

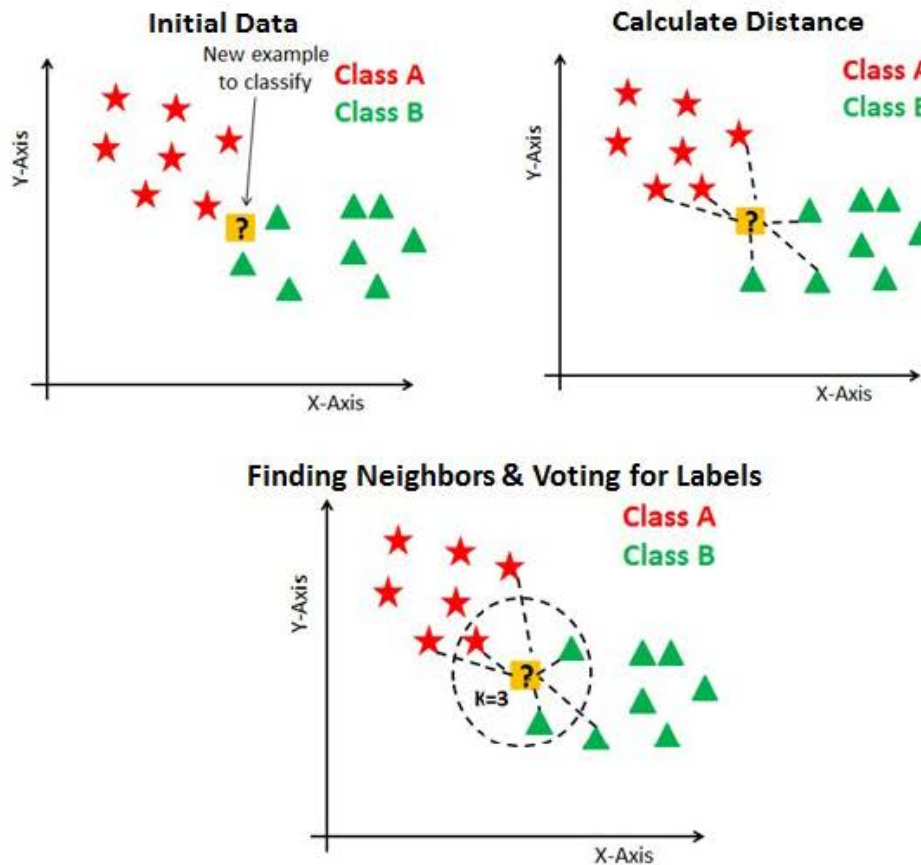


Figure 2.2.5: KNN Model

2.2.6 Naive Bayes

A supervised learning technique called Naive Bayes. It is based on the Bayes theorem. It is employed to address classification issues. Naive Bayes makes the assumption that the existence of one feature in a class has no bearing on the existence of other features. It is predicated on the idea that every feature contributes equally and independently to the final product. It is known as Naive Bayes for this reason. The data set is transformed into a frequency table in Naive Bayes in order to categorize a data instance. A likelihood table is created when the probabilities have been calculated. The Bayes Theorem's formula is $P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$. Here, the probabilities $P(A|B)$ and $P(B|A)$ are posterior probabilities, likelihood of probabilities, prior probabilities, and marginal probabilities, respectively. Using this equation, the posterior probability of each class is determined. The prediction's result is the class with the highest probability. Because Naive Bayes is a quick learner, it is employed in real-time prediction [16].

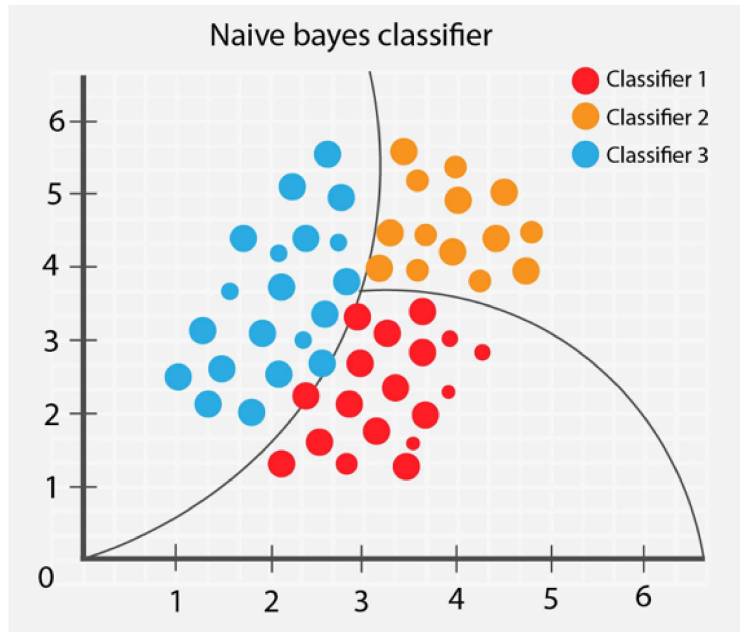


Figure 2.2.6: Naive Bayes Model

2.2.7 Meta-Learning

Meta-learning is the study of learning itself. Meta-learning is an approach to machine learning that seeks to train a model on several learning tasks so that it can acquire new learning tasks with a finite set of training samples. There are two sets of tasks in meta-learning. A training set and a testing set. Learning is done on labeled tasks, not on labeled instances. Meta-learning works with a small number of resources at a faster rate. It can be supervised, unsupervised or reinforcement learning also. Meta-learning delineates construction of higher-level components associated with deep neural networks [17].

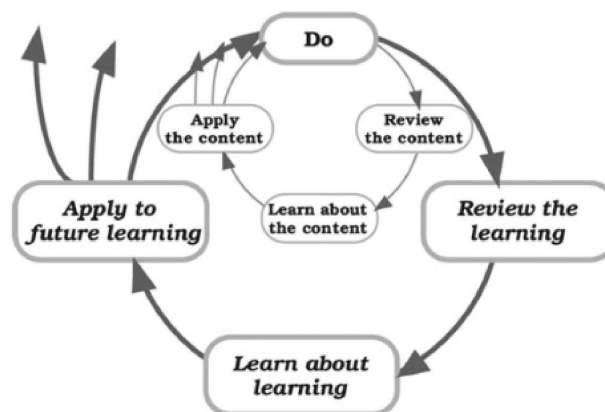


Figure 2.2.7: Meta-Learning Model

2.3 Technique Description

2.3.1 XAI

Explainable AI, or XAI, is a new subfield that seeks to provide human-level explanations for the results of Machine Learning models. This opaque model, such as the popular Deep Learning models, is represented by the black box since it is too complicated to read. Not all Machine Learning models are opaque; for example, linear/logistic regression and decision trees are very simple examples of such models. These models may be simpler to grasp since they provide information about the connection between the feature value and the desired result. But this is not the case when dealing with more intricate models. There are many models in XAI such as LIME, SHAP, ELI5, etc [18].

Chapter 3

Proposed Model

3.1 The Description of Dataset

While the convenience of digital payments increases, so do the opportunities for cybercrime. Card-Present and Card-Not-Present fraud are still widespread, as seen by the daily theft of more than 5 million records, as reported by the Data Breach Index. It is difficult to detect fraud in today's digital world, when trillions of card transactions occur daily.

Most deep learning algorithms can deal with mixed datasets. We are utilizing a hybrid approach in order to establish a system that can accurately carry out the work because our dataset solely contains quantitative data. Some nameless institutions simulated transactions in our simulated dataset are both legitimate and fraudulent. The total dataset includes 10,00,000 transactions, 87,403 of which were genuine fraudulent, and 8 columns containing transaction information [19]. We molded and pruned the dataset for our simulation by removing unnecessary transactions into more narrowed down ones.

3.1.1 Data Manipulation

Nowadays, many AI systems and services that enhance automation by handling analytical and physical activities without human interaction are driven by deep learning and machine learning. Products and services we use every day, as well as those on the horizon, all rely on deep learning and machine learning technology.

Training the dataset with numerical examples from the dataset is the requirement. The train and test scores for each algorithm we utilized indicate the fitness and rigorousness of the approaches used. The following steps comprise our entire data processing, modification, and experimentation process:

- **Data Noise Reduction:** Data reduction is the elimination of irrelevant information. It is a process known as reducing data noise. Initially, the dataset we gathered from Kaggle contained 8 columns or features describing things like `distance_from_home`, `distance_from_last_transaction`, `ratio_to_median_purchase_price`, `repeat_retailer`, `used_chip`, `used_pin_number` and `online_order`.

Though some transitions were necessary for training both deep learning and machine learning models, others could be safely ignored.

- **Data division into label and feature:** We separated the columns into label and feature categories so that we could give instructions to the model algorithms. The dataset's input columns are called "features," and the target column, "label," is the one we are trying to forecast. We use the "is fraud" column as our dataset's label and use the remaining columns as features.
- **Division of the dataset for model training:** We next split the remaining 10,00,000 transaction records from the modified and shaped dataset into a "train set" and a "test set." A total of 1,39,844 records, or 80%, were included in our training set. This information was used by the machine to train a variety of deep learning and machine learning algorithms in preparation for future fraud detection testing.
- **Division of the dataset for model testing:** The remaining 20%, or 34,962 transaction records, were used to test the efficacy of both the deep learning and machine learning algorithms in identifying fraudulent activity. Given that certain algorithms are more effective than others, the test score varies depending on the specific algorithm being tested.
- **Data conversion and mapping:** Successfully completing the necessary training and testing, we moved on to analyzing how certain features were related to fraudulent transactions. Also, we had to derive several important information from other features that were previously considered to be unimportant, such as `distance_from_home` and `distance_from_last_transaction`. Data conversion is the process of giving numerical values to significant but non-numeric aspects. Data mapping is another term for this process. As there is no non-numeric value in our dataset, we do not need to do conversion.

3.1.2 Dataset Preprocessing

Originally, there were 8 columns in the dataset; 7 of these columns described important details about the transactions, while the eighth or last column indicated whether the transactions were fraudulent or non-fraudulent. We denoted this last column as the target variable or label.

Here, we utilized 8 characteristics of our dataset. Moreover, we separated the dataset into X and Y, where X represents "Features" and Y represents "Label." After that we separated our entire dataset into training and testing sets. Training set containing 80% and testing set containing 20% of the entire data.

3.1.3 Features of the dataset

An essential step in detecting credit card fraud is extracting the most notable and pertinent details of a transaction. It is possible that attributes like `distance_from_`

home, distance_from_last_transaction, ratio_to_median_purchase_price and others that have nothing to do with fraudulence were included in the dataset used to train the models. If the card’s owner or card itself has nothing to do with the fraud, then any aspects that would reveal the owner’s personal credentials should be disregarded. Instead, functionality associated with exchanges, consumers, and business owners, respectively, should be utilized. For our convenience, we have compiled all the characteristics we translated and retrieved for use in validation and model training in Table 3.1.1.

SL	Features	Descriptions
1	distance_from_home	The distance from home where the transaction happened
2	distance_from_last_transaction	The distance from last transaction happened
3	ratio_to_median_purchase_price	Ratio of purchased price transaction to median purchase price
4	repeat_retailer	Is the transaction happened from the same retailer
5	used_chip	Is the transaction through chip (credit card)
6	used_pin_number	Is the transaction happened by using a PIN number
7	online_order	Is the transaction an online order
8	fraud	Is the transaction fraudulent

Table 3.1.1: Details about the features

3.2 Model Description

It is crucial to have a clear plan or concept of the process before beginning a large plan like the one completed for this thesis, as even the smallest of omissions can make a significant impact on the final outcomes. To train and validate the machine, we have implemented seven well-known deep learning and machine learning algorithms to analyze the collected data. These seven methods are reliable and show great potential for using both deep learning and machine learning. The first step to do a complete thesis based on deep learning and machine learning is to choose a dataset that contains the right amount and type of data; the second step is to choose ML and DL algorithms that will be responsible for making predictions about the target variable; the third step is to pre-process the dataset; the fourth step is to split the dataset into training and testing sets by using splitting train test method. Training set containing 80% of the entire data; the fifth step is to train the algorithms on the dataset; and the sixth step is Afterwards, it’s important to check if additional pre-processing improves the anticipated values in order to make sure the algorithms are operating as effectively as possible. Finally, it is necessary to determine the efficacy of the ML and DL algorithms and to provide a user-interactive demonstration of data-exchange as a means of reinforcing the argument of the thesis and another model we have implemented, which is Meta-Learning, which works basically learning to learner, first learning through some implemented ML and DL algorithms, and then detecting. At the end, every model’s results are explained through SHAP (a model of XAI or Explainable Artificial Intelligence).

The working mechanism of our proposed model is showed below:

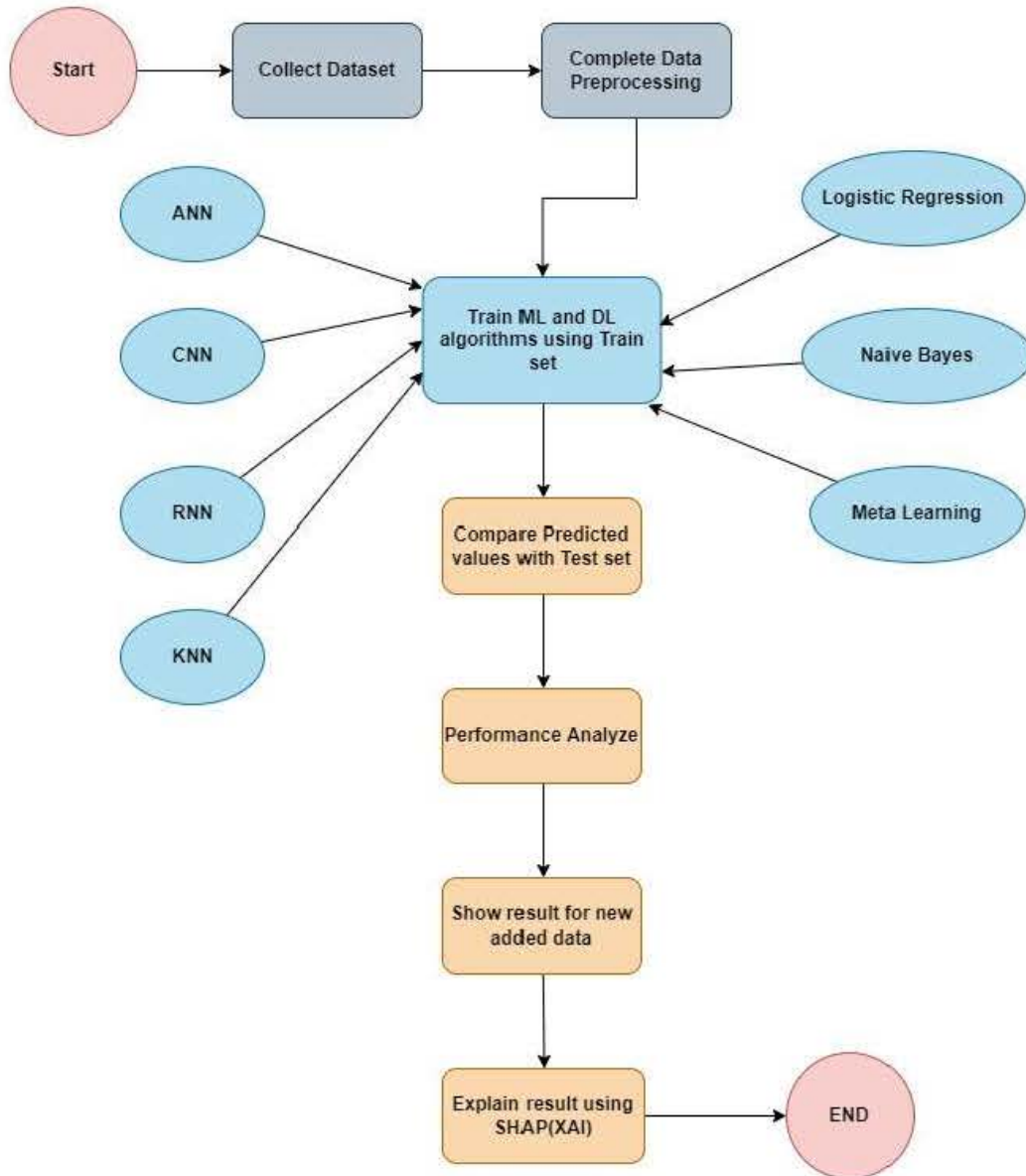


Figure 3.2.1: The working mechanism of proposed model

Chapter 4

Experimentation

4.1 Simulator

We used Colab, a Python-based Web IDE (Integrated Development Environment), where we have run our research and make use of machine learning and deep learning techniques. Python was the programming language we employed. We completed our analysis of the dataset using all necessary libraries including Numpy, Pandas, Matplotlib, Seaborn and others. Finally, we presented our findings in Chapter 5.

4.2 The Metrics of Data Analysis

The simulation data is analyzed by computing various evaluation metrics such as precision, recall, F1-Score, and accuracy. Using these metrics, we can compare our simulated data effectively.

4.2.1 Confusion Matrix

In order to make sense of the data generated by an algorithm, the findings are sometimes displayed in a graphical format known as a Confusion Matrix. It's primary applications are in the study of deep learning and machine learning, the solution of statistical classification issues. These factors all play significant parts in a confusion matrix produced by a deep learning algorithm:

- **True Positive (TP):** A true positive result is one in which the model successfully predicts the occurrence of the positive class.
- **True Negative (TN):** A true negative result is one in which the model predicts the occurrence of the negative class.
- **False Positive (FP):** A false positive result is one in which the model predicts the occurrence of the negative class but the actual class will be positive.

- **False Negative (FN):** A false negative result is one in which the model predicts the occurrence of the positive class but the actual class will be negative.

4.2.2 Precision, Recall, Accuracy & F1-Score

We calculated the training score and the testing score for a single algorithm as well as the precision, recall, and accuracy metrics for different models. Any technique for pattern recognition that assists in locating a certain pattern within a given set of data should prioritize precision, recall, and accuracy as its primary metrics for measuring its performance.

- **Precision:** Precision is a metric of the performance of a machine learning model and the quality of a model's accurate prediction. Precision is the ratio of the number of genuine positives to the overall number of positive forecasts. It is also known as the "positive predictive value." In mathematics, precision is indicated by:

$$Precision = \frac{TP}{TP + FP} \quad (4.1)$$

- **Recall:** Similar to precision, recall is an essential component of pattern recognition, retrieval of information and classification. It is the percentage of appropriately returned instances. In math, it can be represented as:

$$Recall = \frac{TP}{TP + FN} \quad (4.2)$$

- **Accuracy:** The best measure for evaluating the results of a model simulation is accuracy. It is the proportion between all of the accurate predictions and all of the predictions.

$$Recall = \frac{TP + TN}{TP + TN + FN + FP} \quad (4.3)$$

- **F1-Score:** The F1-Score is the last metric used. As a stronger metric, it is calculated utilizing the accuracy and recall values. The formula for determining the model's F1-Score is provided below:

$$F1_Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4.4)$$

Chapter 5

Result Analysis Discussion

5.1 Full Dataset

We evaluated the models using our dataset of 10,00,000 transactions and found significant findings. However, there are 9,12,597 non-fraud transactions and 87,403 fraud transactions, making this dataset very imbalanced. Therefore, To resolve this issue we used a technique of using an equal proportion of both fraud and non-fraud transactions. Since there are only 87,403 fraud transactions, we utilized the same amount of fraud and non-fraud transactions to compare the metrics of all the models.

Category	Values
Fraud Transactions	87403
Non-Fraud Transactions	912597
Train set	800000
Test set	200000
Total Transaction	1000000

Table 5.1.1: Description of the full dataset

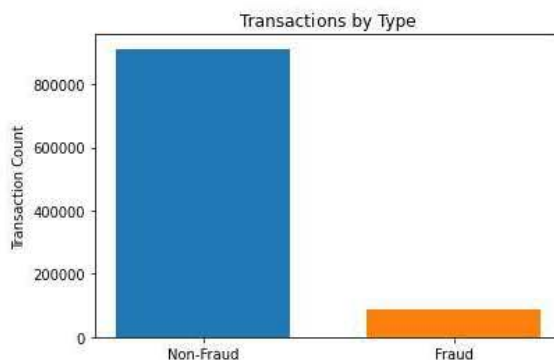


Figure 5.1.1: Description of the full dataset

In order to run simulations of our models, we use the splitting train test method to split the full dataset into "train" and "test" sets. We know that supervised machine learning and deep learning models require training data; we chose 8,00,000(80%) transactions from the full dataset to train the model and 2,00,000(20%) transactions from the full dataset to test the model.

We used accuracy, precision, recall, F1-Score metrics in our entire dataset to evaluate how well the models performed. Table 5.1.2 shows that the results are excessively accurate and over-fitting, suggesting that an unbalanced dataset cannot produce reliable results. The performance metrics for the full dataset model simulations are presented in Table 5.1.2.

Models	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)
KNN	99.89	99.57	99.16	99.37
CNN	99.87	99.14	99.41	99.27
RNN	99.81	98.78	99.32	99.06
ANN	99.76	98.75	98.50	98.62
Meta-Learning	98.1	92.7	91.2	96.4
Logistic Regression	95.91	89.50	60.26	72.08
Naive Bayes	95.14	79.15	60.23	68.40

Table 5.1.2: Performance Metrics of Full Dataset

From the table, we can see that there are some discrepancies among the values when we took the full dataset. It means that for the imbalanced dataset, there we cannot see the consistency among the values. It is true that the accuracies are quite close for all the models, however, the numbers are quite high here. Moreover, in the precision, we can see a huge difference between the KNN and Naive Bayes. Also, the Recall and F1-Score, the numbers decreased from the above models. Therefore, when we take the unbalanced dataset, we cannot produce reliable results.

5.2 Balanced Dataset

We have measured the models' efficacy using the precision, recall, accuracy, and F1-score metrics. Because our dataset was so unbalanced, we changed our approach based on sampling approximately the same number of actual and fraudulent transactions. So that we can quickly evaluate the highest performing models. To make our dataset a balanced one, we have taken the equal amount of fraud and non-fraud transactions. Here, we have taken 87,403 fraud transactions as well as 87,403 non-fraud transactions.

Category	Values
Fraud Transactions	87403
Non-Fraud Transactions	87403
Train set	139844
Test set	34962
Total Transactions	174806

Table 5.2.1: Description of the balanced dataset

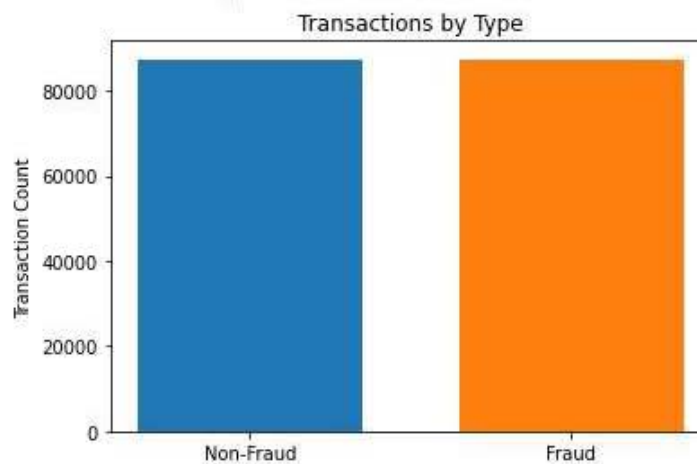


Figure 5.2.1: Description of the balanced dataset

Now, for the balanced dataset, to run simulations of our models, we split the dataset into "train" and "test" sets. Here, we have modified our dataset into a balanced dataset as well as the train set and test set. Given that supervised deep learning and machine learning models require training data, we decide to train 1,39,844 transactions on the model and to test 34,962 transactions on the model.

For the whole dataset, we used accuracy, precision, recall, F1-Score metrics to evaluate how well those models performed. Table 5.2.1 show the results for the balanced dataset. The performance metrics for the balanced dataset model simulations are shown in Table 5.2.2.

Models	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)
KNN	99.75	99.58	99.91	99.75
RNN	99.66	99.49	99.73	99.61
ANN	99.61	99.48	99.74	99.60
CNN	99.26	98.73	99.81	99.27
Meta-Learning	98.1	99.04	97.03	97.8
Naive Bayes	94.01	96.54	91.61	94.01
Logistic Regression	93.92	93.34	94.58	93.96

Table 5.2.2: Performance Metrics of Balanced Dataset

With all of the evaluation metrics mentioned above, KNN (when $k = 3$) has achieved a maximum accuracy of 99.75%. Besides, the second highest is the RNN around 99.66%. ANN also gives very similar results like RNN, its accuracy is 99.61%, whereas CNN gives 99.26% accuracy and Meta-Learning gives 98.1% accuracy. However, we can see some decrease in accuracy in Naive Bayes (94.01%) and Logistic Regression (93.92%).

5.3 ROC CURVE ANALYSIS

For the purpose of demonstrating the diagnostic efficacy of binary classifiers, a Receiver Operator Characteristic (ROC) curve is plotted graphically. In order to create a ROC curve, one must first plot the true positive rate (TPR) versus the false positive rate (FPR). A test's true positive rate is the fraction of tested positives that match the expected positives $\frac{TP}{(TP+FN)}$. The false positive rate $\frac{FP}{(TN+FP)}$ is the percentage of negative observations that are mistakenly projected to be positive.

If a classifier only provides back the class it has predicted, it only has one point on the ROC plot. Instead, we have developed a curve by adjusting the score threshold for probabilistic classifiers, which provide a probability or score to each instance that represents the degree to which it belongs to one class rather than another. It is possible to "look inside" the instance statistics of many discrete classifiers and transform them into scoring classifiers [20]. This score is known as AUC (Area Under the Curve) score. Every ROC curve is generated here by calculating the AUC score. Below, we have shown all the ROC curves for all the models.

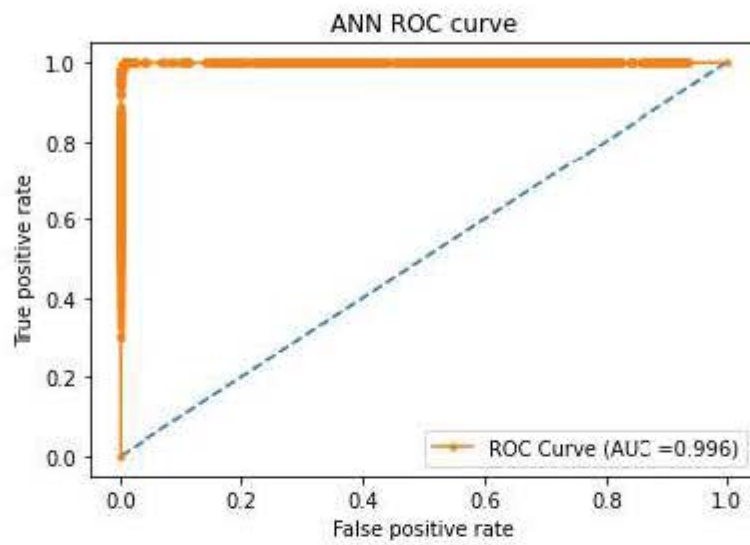


Figure 5.3.1: ROC for ANN

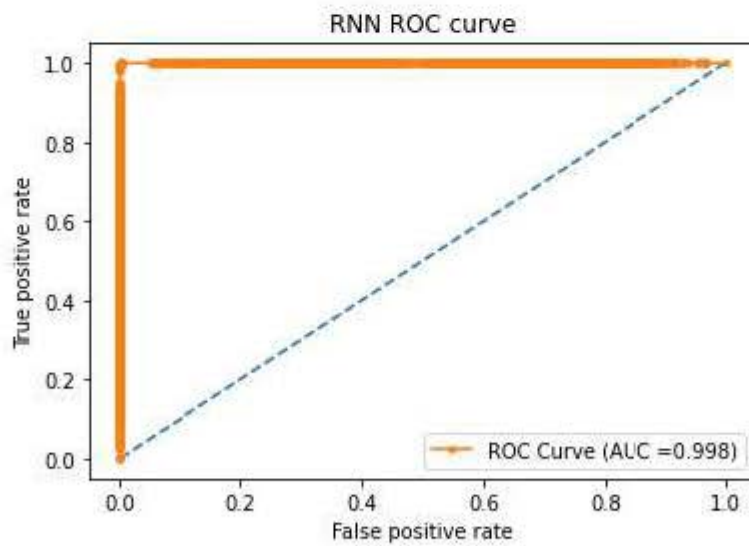


Figure 5.3.2: ROC for RNN

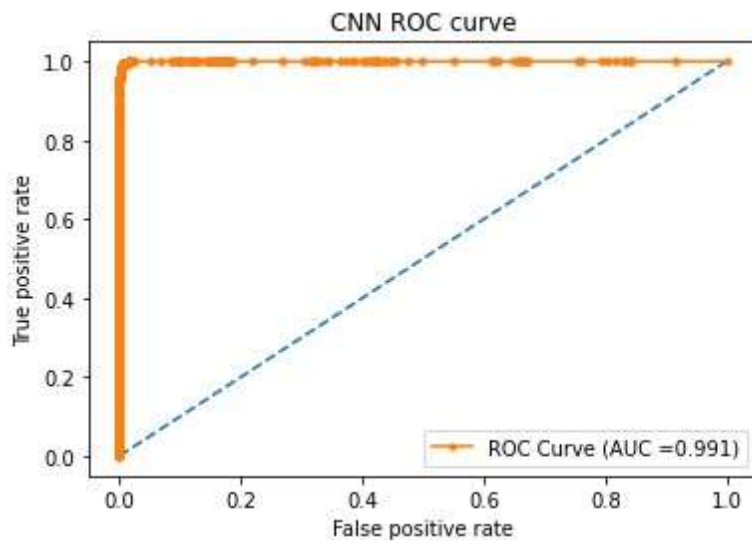


Figure 5.3.3: ROC for CNN

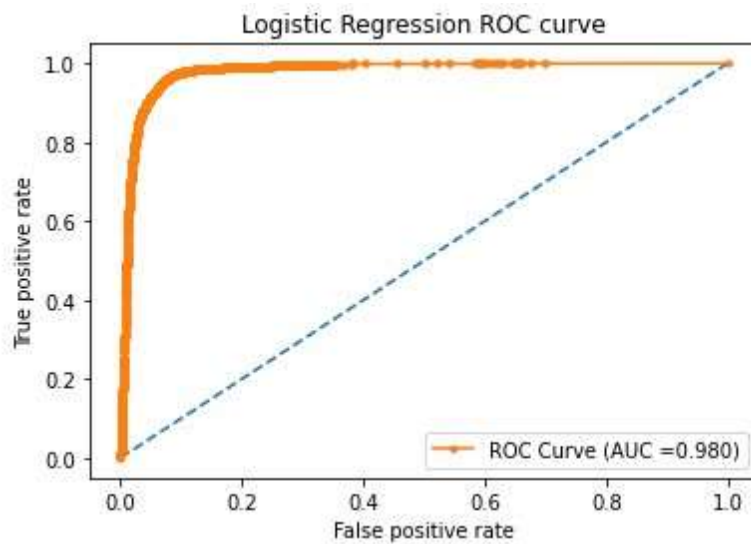


Figure 5.3.4: ROC for Logistic Regression

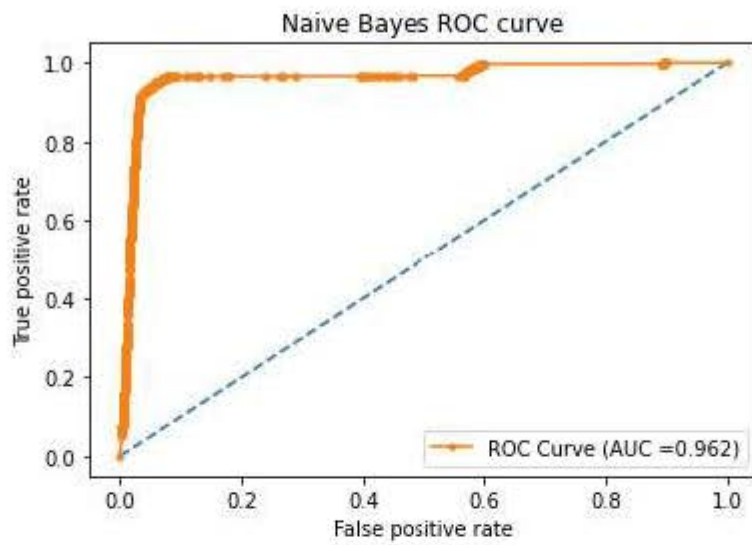


Figure 5.3.5: ROC for Naive Bayes

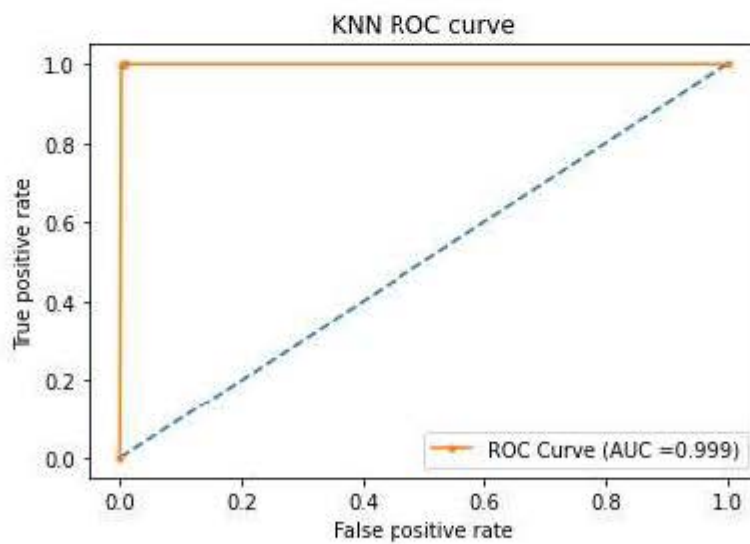


Figure 5.3.6: ROC for KNN

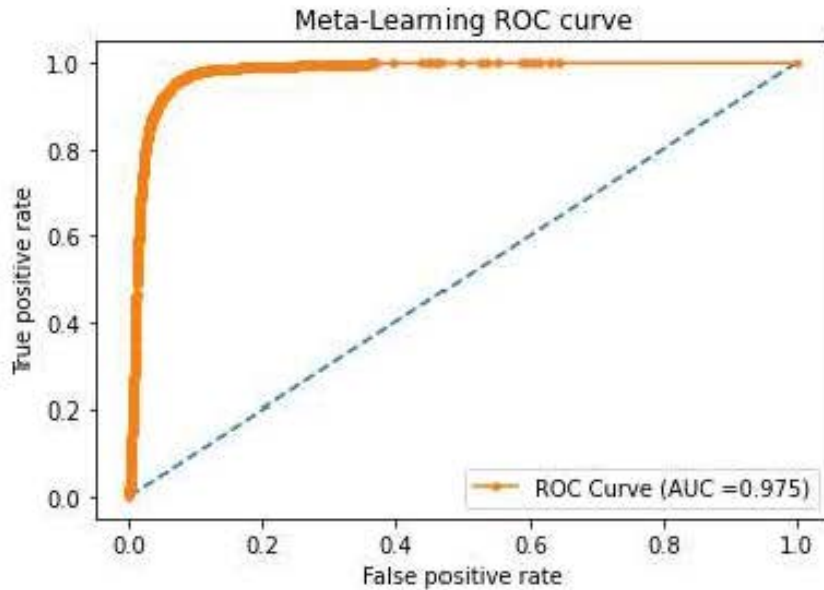


Figure 5.3.7: ROC for Meta-Learning

From all the above ROC curves, KNN has achieved the highest AUC score which is 0.999. Where all the other ROC curves have lower AUC scores compared to KNN. That's why the ROC curve for KNN is situated near the top left corner of the graph which mentioned that KNN has a more efficient model than others.

It is the trade-off between sensitivity (or TPR) and specificity (or FPR) that is displayed by the ROC curve ($1 - \text{FPR}$). Better performance is indicated by classifiers whose output curves are shifted to the upper left. A random classifier's output should be a set of points on the diagonal ($\text{FPR} = \text{TPR}$) as a starting point. If the curve approaches the 45 degree diagonal of the ROC space, the test's accuracy decreases.

ROC is independent of the number of classes. As a result, it can be put to good use in assessing classifiers' abilities to foresee unusual occurrences. In contrast, classifiers that always predict a negative result for uncommon events would fare better if performance was evaluated using $\text{accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN}) + \text{FN} + \text{FP}}$ [20].

5.4 Confusion Matrix Analysis

Here, we have shown the confusion matrices for KNN, ANN, and RNN models below which we have implemented so far.

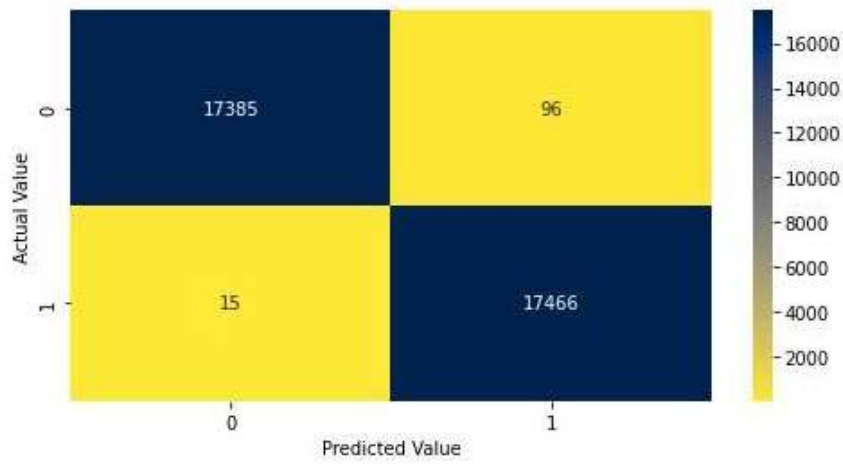


Figure 5.4.1: KNN Confusion Matrix

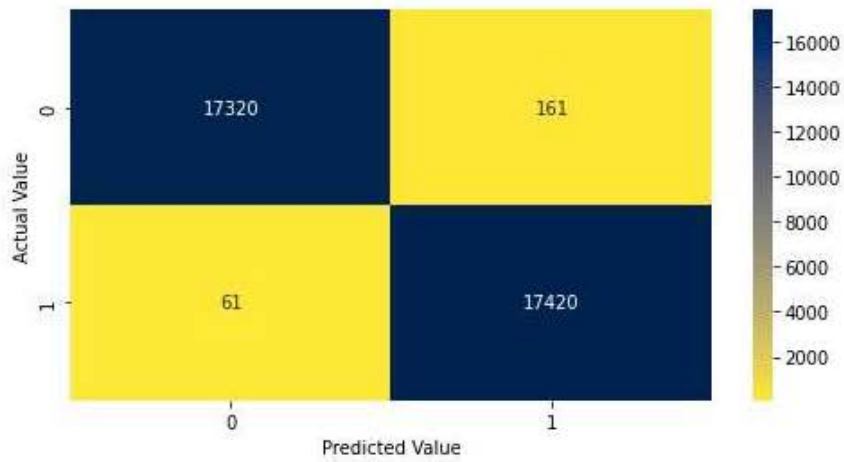


Figure 5.4.2: ANN Confusion Matrix

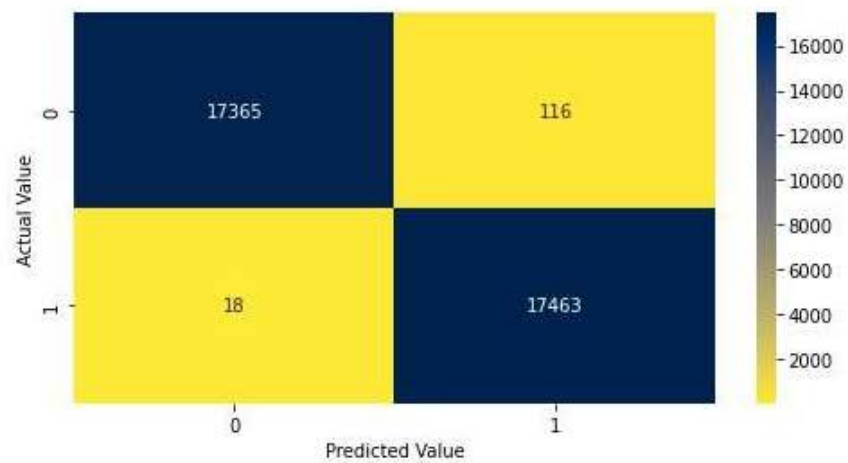


Figure 5.4.3: RNN Confusion Matrix

5.5 XAI Analysis (SHAP)

SHAP is an abbreviation for "SHapley Additive exPlanations". It is a technique for determining the effect of a given factor on the value of the target variable. Every model can be understood with the help of SHAP. One key idea is that a feature's significance is dependent not only on that feature but also on all of the features included in the dataset as a whole. SHAP uses combinatorial calculus to retrain the model through all possible combinations of features that contain the one we are examining, and then estimates the effect of each feature on the target variable (the SHAP value). We may evaluate how significant a characteristic is by looking at its average absolute value of impact against a target variable. One advantage of SHAP is that it is model-agnostic. Specifically, it is not tied to any one model. It's a great way to shed light on models that don't provide their own assessment of feature significance. In this part, we will utilize SHAP in KNN, Logistic regression, Naive Bayes, ANN, CNN, and RNN to determine the relative importance of each feature.

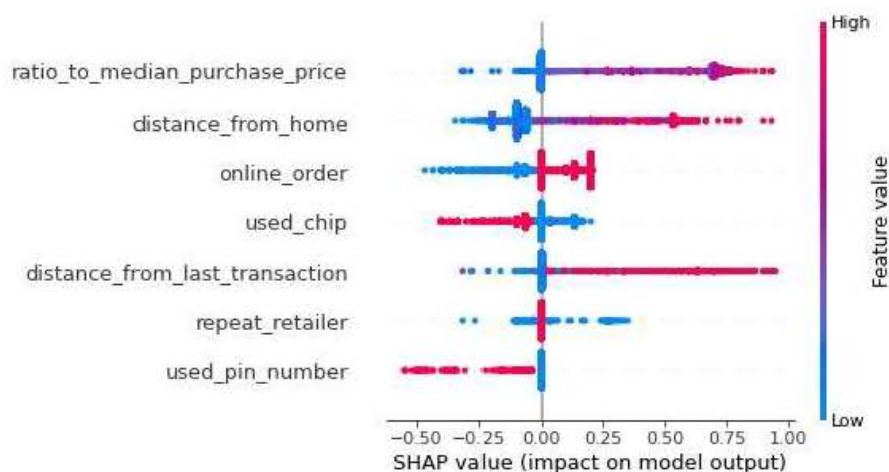


Figure 5.5.1: XAI (SHAP) for KNN

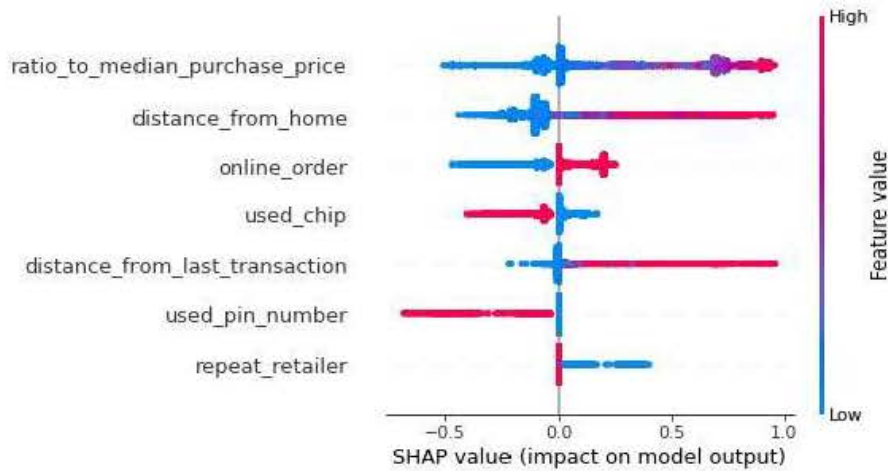


Figure 5.5.2: XAI (SHAP) for Logistic Regression

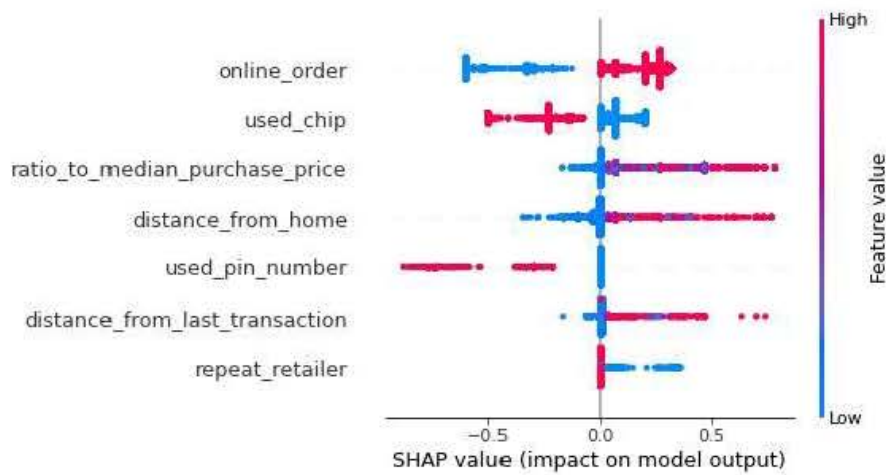


Figure 5.5.3: XAI (SHAP) for Naive Bayes

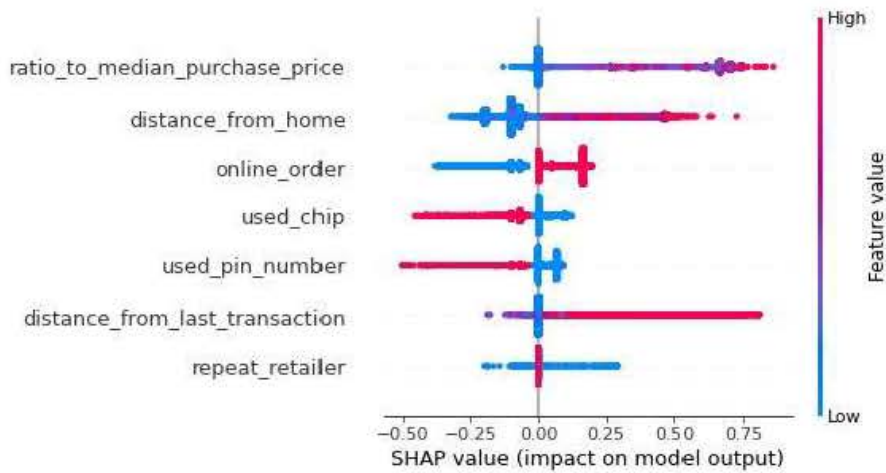


Figure 5.5.4: XAI (SHAP) for ANN

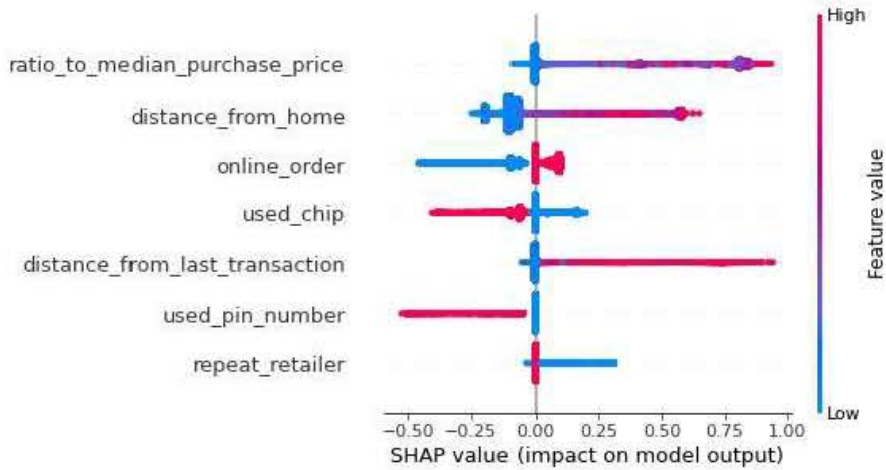


Figure 5.5.5: XAI (SHAP) for CNN

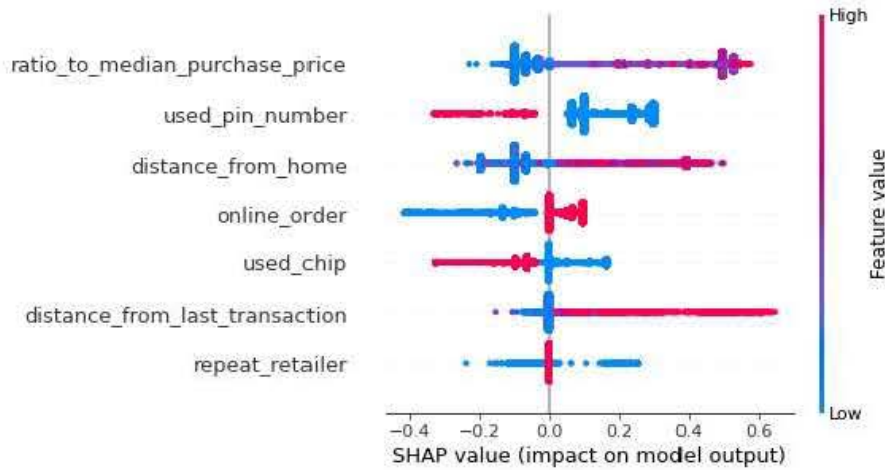


Figure 5.5.6: XAI (SHAP) for RNN

Each column in every row represents a record in the dataset. There is a hierarchy established for the features, from most important to cheapest. For all models except Naive Bayes, the `ratio_to_median_purchase_price` feature stands out as the most important one. For Naive Bayes, `online_order` is the most important feature. Moreover, of all the models except KNN, the `repeat_retailer` feature stands out as the least important one. For KNN, `used_pin_number` is the least important one. A higher value for this feature has a more advantageous effect on the target. This contribution will be increasingly negative as this value decreases. SHAP is a highly effective method when it comes to explaining models that can't grasp the value of features on their own [21].

5.6 Comparison

Some related papers also showed their results based on the similar dataset. From our model, from Table 5.2.2, we showed that KNN (when $k = 3$) has scored the highest accuracy and its around 99.75%. Besides, the precision, recall, F1-Score are also scored the highest value among the other models. However, we have made a comparison among some other relevant papers with our thesis research and displayed the result in the Table 5.6.1.

Reference	Best Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
[22]	Isolation Forest	97	99.8	99.45	99.75
[23]	ANN with GA	99.83	50.70	97.27	66.66
[24]	Random Forest Classifier	96	-	76	65
[25]	ANN	99.48	21.34	86.39	-
[26]	AE	99.90	89.55	57.14	69.77
[27]	Support Vector Machine	94.99	95.98	-	-
[28]	Support Vector Machine	94.99	95.98	95.12	95.11
[29]	Xgboost	96.44	96	97	96
[30]	Deep NN	98.12	-	83.52	87.65
Our Model	KNN	99.75	99.58	99.91	99.75

Table 5.6.1: Comparison between relevant papers and our model

5.7 Discussion

From Table 5.1.2, we can see that our deep learning and machine learning models have an accuracy of around 99.89%, indicating that they are working at a level which is almost perfect. But in most cases, this level of accuracy is very high. On the contrary, In our full dataset, fraud transactions occurred only 0.087% from all transactions. This causes over-fitting in our models which is shown in Table 5.1.1 and Table 5.1.2.

Unfortunately, the accuracy of the simulated models dropped below the full-dataset level once we partitioned our data into a balanced dataset. Table 5.2.2 demonstrates the accuracy of our balanced datasets. K-Nearest Neighbor (KNN) has delivered the greatest accuracy for both models among the two models we utilized for simulation and evaluation. The accuracy of KNN (when $k = 3$), CNN, RNN, ANN, Meta-Learning, Logistic Regression and Naive Bayes Support in full-dataset are 99.89%, 99.87%, 99.81%, 99.76%, 98.1%, 95.91% and 95.14% respectively. Additionally, We

also performed simulations using 87,403 fraud transactions to test the models' reliability. The accuracy of KNN (when $k = 3$), RNN, ANN, CNN, Meta-Learning, Naive Bias and Logistic Regression in balanced-dataset are 99.75%, 99.66%, 99.61%, 99.26%, 98.1%, 94.01% and 93.92% respectively. Finally, we can easily get the accuracy of both datasets. Among them, the accuracy of full-dataset models is more than the accuracy of balanced-dataset models where the number of non-fraud is just 87,403.

In order to evaluate the new balanced datasets, we obtained model performance metrics and compared them with some of the relevant papers. Isolation Forest has the best accuracy of 97% shown in [22]. Besides, in this paper [23], ANN with GA got 99.83% accuracy whereas its precision is around 50.70% which is very low here. Moreover, the best algorithm in [27] & [28] is Support Vector Machine, with the same accuracy 94.99%, and precision 95.98%. With these outcomes in mind, the best algorithm we have is KNN, which achieves an 99.75% accuracy, 99.58% precision, 99.91% recall, and 99.75% F1-Score. When compared to other findings, our KNN performs best in terms of accuracy, recall, precision and F1-Score. Possible explanations include our effective approach of splitting the original unbalanced dataset into balanced datasets with varying quantities of non-fraud transactions.

Chapter 6

Conclusion

6.1 Conclusion & Future work

One cannot overstate the dishonesty of credit card fraud. Fraud using credit cards is a growing problem for banks. New fraud strategies are often developed by fraudsters. Due to the dynamic nature of fraud, a powerful classifier is necessary. This paper reviews recent developments in this sector and identifies the most prevalent types of fraud and how they might be detected. Along with the method, pseudocode, description of its implementation, and experimental findings for detecting fraud, this paper also explains how machine learning and deep learning might be applied to achieve better outcomes. The primary goal of any fraud detection system should be to accurately forecast fraud situations while minimizing false positives. According to the specifics of each business scenario, ML and DL approaches may or may not be effective. When it comes to machine learning and deep learning, the nature of the input data is the most important determining element. The effectiveness of a model for identifying Credit Card fraud relies heavily on the amount of characteristics it utilizes, the number of transactions it processes, and the level of correlation between those features.

With the right data trimming, noise reduction, feature extraction, and model training, real-time credit card fraud detection is now a reality. K-Nearest Neighbor (KNN) had the highest accuracy (99.75%) out of the seven supervised machine learning and deep learning models. Although oversampling and undersampling are two common approaches to addressing dataset imbalance, we choose instead to use a novel approach, Meta-Learning. While our strategy is effective and satisfactory in the end, it does take time to find enough legitimate data to train the model.

There was little to no difference in performance among algorithms, but we can speculate that, with further training on additional real-world data, accuracy and precision might improve. We are not yet at 100% accuracy, despite having implemented a number of data mining techniques, but we are working to improve it by integrating several algorithms that, taken together, can improve our accuracy. We will try to reduce False Negatives so that, as our learning progresses, our accuracy may improve. Future experiments will include testing whether or not we can improve results by training models with additional data and using genetic algorithms [31].

Bibliography

- [1] Jorge Galindo and Pablo Tamayo. “Credit risk assessment using statistical and machine learning: basic methodology and risk modeling applications”. In: *Computational economics* 15.1 (2000), pp. 107–143. DOI: <https://doi.org/10.1089/big.2014.0018>. URL: https://link.springer.com/article/10.1023/A:1008699112516?error=cookies_not_supported&code=b5845b26-7112-45cc-a601-7a5b71457440.
- [2] Paulius Danenas, Gintautas Garsva, and Saulius Gudas. “Credit risk evaluation model development using support vector based classifiers”. In: *Procedia Computer Science* 4 (2011), pp. 1699–1707. DOI: <https://doi.org/10.1016/j.procs.2011.04.184>.
- [3] Peter Martey Addo, Dominique Guegan, and Bertrand Hassani. “Credit risk analysis using machine and deep learning models”. In: *Risks* 6.2 (2018), p. 38. DOI: [10.3390/risks6020038](https://doi.org/10.3390/risks6020038).
- [4] Trilok Nath Pandey et al. “Credit risk analysis using machine learning classifiers”. In: *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*. IEEE. 2017, pp. 1850–1854. DOI: [10.1109/ICECDS.2017.8389769](https://doi.org/10.1109/ICECDS.2017.8389769).
- [5] Bart Baesens et al. “Benchmarking state-of-the-art classification algorithms for credit scoring”. In: *Journal of the operational research society* 54.6 (2003), pp. 627–635. DOI: [10.1057/palgrave.jors.2601545](https://doi.org/10.1057/palgrave.jors.2601545).
- [6] Stjepan Oreski and Goran Oreski. “Genetic algorithm-based heuristic for feature selection in credit risk assessment”. In: *Expert systems with applications* 41.4 (2014), pp. 2052–2064. DOI: [10.1016/j.eswa.2013.09.004](https://doi.org/10.1016/j.eswa.2013.09.004).
- [7] A Ahmed et al. “Building a credit scoring model to assign a reference score based on credit transaction and relevant profile data”. PhD thesis. Ph. D. dissertation, 2019.
- [8] Armin Lawi, Firman Aziz, and Syafruddin Syarif. “Ensemble GradientBoost for increasing classification accuracy of credit scoring”. In: *2017 4th international conference on computer applications and information processing technology (CAIPT)*. IEEE. 2017, pp. 1–4. DOI: [10.1109/CAIPT.2017.8320700](https://doi.org/10.1109/CAIPT.2017.8320700).
- [9] Jasmina Nalić and Amar Švraka. “Importance of data pre-processing in credit scoring models based on data mining approaches”. In: *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE. 2018, pp. 1046–1051. DOI: [10.23919/MIPRO.2018.8400191](https://doi.org/10.23919/MIPRO.2018.8400191).

- [10] J. Nalić and A. Švraka. “Using data mining approaches to build credit scoring model: Case study—Implementation of credit scoring model in microfinance institution”. In: *2018 17th International Symposium Infoteh-Jahorina (INFOTEH)*. IEEE. 2018, pp. 1–5.
- [11] Yuanyuan Tian, Mi Shu, and Qingren Jia. “Artificial Neural Network”. In: *Encyclopedia of Mathematical Geosciences*. Springer, 2022, pp. 1–4.
- [12] Md Kabir. *Convolutional Neural Network*. Jan. 2021. DOI: 10.13140/RG.2.2.29424.69127.
- [13] Raul Fernandez et al. “Prosody contour prediction with long short-term memory, bi-directional, deep recurrent neural networks.” In: *Interspeech*. 2014, pp. 2268–2272.
- [14] Rollin Brant. “Digesting logistic regression results”. In: *The American Statistician* 50.2 (1996), pp. 117–119. DOI: 10.1080/00031305.1996.10474358.
- [15] Anjali Ganesh Jivani et al. “The adept K-nearest neighbour algorithm—an optimization to the conventional K-nearest neighbour algorithm”. In: *Transactions on Machine Learning and Artificial Intelligence* 4.1 (2016), p. 52. URL: <https://doi.org/10.14738/tmlai.41.1876>.
- [16] Sona Taheri and Musa Mammadov. “Learning the naive Bayes classifier with optimization models”. In: *International Journal of Applied Mathematics and Computer Science* 23.4 (2013), pp. 787–795. URL: <https://doi.org/10.2478/amcs-2013-0059>.
- [17] Jun Wu and Jingrui He. “A Unified Meta-Learning Framework for Dynamic Transfer Learning”. In: *arXiv preprint arXiv:2207.01784* (2022). URL: <https://doi.org/10.48550/ARXIV.2207.01784>.
- [18] Zeynep Betül Arıkan. *An introduction to explainable artificial intelligence (XAI)*. URL: <https://www.mobiquity.com/insights/an-introduction-to-explainable-artificial-intelligence>.
- [19] DHANUSH NARAYANAN R. *Credit Card Fraud*. 2022. URL: <https://www.kaggle.com/datasets/dhanushnarayananr/credit-card-fraud>.
- [20] Carmen Chan. *What is a ROC curve and how to interpret it*. Aug. 2022. URL: <https://www.displayr.com/what-is-a-roc-curve-how-to-interpret-it/>.
- [21] Gianluca Malato. *How to explain neural networks using shap*. Nov. 2021. URL: <https://www.yourdatateacher.com/2021/05/17/how-to-explain-neural-networks-using-shap/>.
- [22] Shubham Jaiswal, R Brindha, and Shubham Lakhota. “Credit Card Fraud Detection Using Isolation Forest and Local Outlier Factor”. In: *Annals of the Romanian Society for Cell Biology* (2021), pp. 4391–4396.
- [23] Anuruddha Thennakoon et al. “Real-time credit card fraud detection using machine learning”. In: *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE. 2019, pp. 488–493.
- [24] Yeasin Arafath et al. “Developing a Framework for Credit Card Fraud Detection”. In: *Proceedings of the International Conference on Big Data, IoT, and Machine Learning*. Springer. 2022, pp. 637–651.

- [25] John O Awoyemi, Adebayo O Adetunmbi, and Samuel A Oluwadare. “Credit card fraud detection using machine learning techniques: A comparative analysis”. In: *2017 international conference on computing networking and informatics (ICCNi)*. IEEE. 2017, pp. 1–9.
- [26] Arjwan H. Almuteer et al. “Detecting Credit Card Fraud using Machine Learning”. In: *International Journal of Interactive Mobile Technologies (IJIM)* 15.24 (Dec. 2021), pp. 108–122. DOI: 10.3991/ijim.v15i24.27355. URL: <https://online-journals.org/index.php/i-jim/article/view/27355>.
- [27] KS Varun Kumar et al. “Credit Card Fraud Detection using Machine Learning Algorithms”. In: *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 9* (2020).
- [28] Sheo Kumar et al. “Credit Card Fraud Detection Using Support Vector Machine”. In: *Proceedings of the 2nd International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications*. Springer. 2022, pp. 27–37.
- [29] Krishna Kumar Mohbey, Mohammad Zubair Khan, and Ajay Indian. “Credit Card Fraud Prediction Using XGBoost: An ensemble Learning Approach”. In: *International Journal of Information Retrieval Research (IJIRR)* 12.2 (2022), pp. 1–17.
- [30] Dinara Rzayeva and Saber Malekzadeh. “A Combination of Deep Neural Networks and K-Nearest Neighbors for Credit Card Fraud Detection”. In: *arXiv preprint arXiv:2205.15300* (2022).
- [31] D Tanouz et al. “Credit card fraud detection using machine learning”. In: *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE. 2021, pp. 967–972.