CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING APPROACH

A PROJECT REPORT

Submitted by

BADARNISA (TKM17MCA017)

to

The APJ Abdul Kalam Technological University

In partial fulfillment of the requirements for the award of the Degree of

MASTER OF COMPUTER APPLICATIONS

Thangal Kunju Musaliar College of Engineering
Kerala
DEPARTMENT OF COMPUTER APPLICATIONS

MAY 2022

DECLARATION

I undersigned hereby declare that the project report "CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING APPROACH", submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Prof. Jasmin M R. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Kollam

Date:27-05-22

Badarnisa

DEPARTMENT OF COMPUTER APPLICATIONS

TKM COLLEGE OF ENGINEERING

C E R T I F I C A T E

This is to certify that, the report entitled ***"CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING APPROACH"*** submitted by **BADARNISA**, to the **APJ Abdul Kalam Technological University** in partial fulfillment of the requirements for the award of the Degree of **Master of Computer Applications** is a bonafide record of the project work carried out by her under our guidance and supervision.This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor | Head of the Department | External Examiner

ACKNOWLEDGEMENT

First and foremost I thank GOD almighty and my parents for the success of this project. I owe sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project.

I am extremely grateful to **Dr.Fousia M Shamsudeen,** Head of the Department, for providing us with best facilities.

I would like to thank my project guide **Prof.Jasmin M R,** Department of Computer Applications, who motivated me throughout the project.

I profusely thank all other faculty members in the department and all other members of TKM College of Engineering, for their guidance and inspirations throughout the course of study.

I owe my thanks to our friends and all others who have directly or indirectly helped us in the successful completion of this project.

Badarnisa

ABSTRACT

Payment cards offer a simple and convenient method for making purchases. Generally credit card fraud activities can happen in both online and offline. But in today's world online fraud transaction activities are increasing day by day. It is therefore crucial to implement mechanisms that can detect the credit card fraud.Features of credit card frauds play important role when machine learning is used for credit card fraud detection, and they must be chosen properly.So in order to find the online fraud transactions various methods have been used in existing system.In this proposed project designed a model to detect the fraud activity in credit card transactions. This system can provide most of the important features required to detect illegal and illicit transactions.As technology changes constantly it is becoming difficult to track the behavior and pattern of criminal transactions.

The algorithms such as : Random Forest and Extreme Gradient Boosting "XGBoost". This algorithms is based unsupervised learning algorithm. After classification of data set a confusion matrix is obtained. The performance of the algorithm is evaluated based on the confusion matrix.

Contents

List of Figures

List Of Abbreviations

3.3	SMOTE	Synthetic Minority Over-sampling Technique
3.4	ADASYN	Adaptive Synthetic
3.4.2	XGBoost	Extreme Gradient Boosting

Chapter 1

Introduction

Many types of payment cards, which include credit, debit, and prepaid cards, are widely available nowadays.Increase in the use of services such as e-commerce, tap and pay systems, online bills payment systems etc.As a consequence, fraudsters have also increased activities to attack transactions that are made using credit cards. Nowadays Credit card usage has been drastically increased across the world, now people believe in going cashless and are completely dependent on online transactions. The credit card has made the digital transaction easier and more accessible. A huge number of dollars of loss are caused every year by the criminal credit card transactions. Fraud is as old as mankind itself and can take an unlimited variety of different forms. The PwC global economic crime survey of 2017 suggests that approximately 48 percentage of organizations experienced economic crime. Therefore positively a necessity to unravel the matter of credit card fraud detection.Moreover, the growth of new technologies provides supplementary ways in which criminals may commit a scam.The use of credit cards is predominant in modern day society and credit card fraud has been kept on increasing in recent years.Huge Financial losses have been fraudulent effects so not only merchants and banks but also the individual person who are using the credits.For example, if a cardholder is a victim of fraud with a certain company, he may no longer trust their business and choose a competitor. Fraud Detection is the process of monitoring the transaction behavior of a cardholder to detect whether an incoming transaction is authentic and authorized or not otherwise it will be detected as illicit.

1.1 Existing System

In exististing system the real payment card database used is limited .Use of single models for developing the fraud detection frame work

Disadvantages

- We predicted only banking sector customers' dataset.
- In today's technological conditions, new data are being produced by different sources in many sectors.
- However, it is not possible to extract the useful information hidden in these data sets, unless they are processed properly.
- In order to find out this hidden information, various analyses should be performed using data mining, which consists of numerous methods.

1.2 Problem Statements

For many banks, retaining high profitable customers is the number one business goal. Banking fraud, however, poses a significant threat to this goal for different banks. In terms of substantial financial losses, trust and credibility, this is a concerning issue to both banks and customers alike.In the banking industry, credit card fraud detection using machine learning is not only a trend but a necessity for them to put proactive monitoring and fraud prevention mechanisms in place. Machine learning is helping these institutions to reduce time-consuming manual reviews, costly charge backs and fees as well as denials of legitimate transactions.The project detect fraudulent credit card transactions with the help of Machine learning models.

1.3 Objectives

The main objectives of the project are as follows:

- The main objective is to find a fraudlent transaction in credit card transaction
- Comparison between the supervised algorithm using different technologies
- Detect fraudlent credit card transaction with the help of Machine learning models

Chapter 2

Literature Survey

Along with increasing credit card and fraud rises sharply. How to enhance the detection and bar of credit card fraud becomes the main target of risk management.With growing advancement within the electronic commerce field, fraud is spreading all over the world, causing major financial losses. In current scenario, Major cause of financial losses is credit card fraud. It not only affects trades person but also individual clients. Decision tree, Genetic algorithm, Meta learning strategy, neural network, HMM are the presented methods used to detect credit card frauds., classification models supported on Random forest and XGBoost are developed and applied on credit card fraud detection Problems. The study is one of the first to compare the performance of Ra ndom forestand XGBoost methods in credit card fraud detection with a real data set

2.1 Purpose of the Literature Review

1. It gives readers easy access to research on a particular topic by selecting high quality articles or studies that are relevant, meaningful, important and valid and summarizing them into one complete report.

2. It provides an excellent starting point for researchers beginning to do research in a new area by forcing them to summarize, evaluate, and compare original research in that specific area.

3. It ensures that researchers do not duplicate work that has already been done.

4. It can provide clues as to where future research is heading or recommend areas on which to focus.

5. It highlights the key findings.

6. It identifies inconsistencies, gaps and contradictions in the literature.

7. It provides a constructive analysis of the methodologies and approaches of other researchers.

2.2 Literature Review of Credit card fraud detection

Here, we take some of the papers rela methods,

Hennakoon et al.[1] Credit card fraud events take place frequently and then result in huge financial losses.The number of online transactions has grown in large quantities and online credit card transactions holds a huge share of these transactions. The paper focuses on four main fraud occasions in real-world transactions. Each fraud is addressed using a series of machine learning models and the best method is selected via an evaluation.Take the use of predictive analytics done by the implemented machine learning models and an API module to decide if a particular transaction is genuine or fraudulent

Varmedja et al.[2] Credit card fraud refers to the physical loss of credit card or loss of sensitive credit card information. Many machine-learning algorithms can be used for detection. The research shows several algorithms that can be used for classifying transactions as fraud or genuine one. Credit Card Fraud Detection data set was used in the research. Because the data set was highly imbalanced, SMOTE technique was used for oversampling.The algorithms used in the experiment were Logistic Regression, Random Forest, Naive Bayes and Multilayer

Awoyemi et al.[3] Financial fraud is an ever growing menace with far consequences in the financial industry. Data mining had played an imperative role in the detection of credit card fraud in online transactions.The performance of fraud detection in credit card transactions is greatly affected by the sampling approach on data set, selection of variables and detection technique(s) used.Algorithm used in naïve bayes, k-nearest neighbor and logistic regression on highly skewed

credit card fraud data.The three techniques are applied on the raw and preprocessed data. The work is implemented in Python.

Pillai et al.[4] Frauds in credit card transactions are common today as most of us are using the credit card payment methods more frequently.fraudsters find ways to steal the credit card information of the user by sending fake SMS and calls, also through masquerading attack, phishing attack and so on.using the multiple algorithms of Machine learning such as support vector machine (SVM), k-nearest neighbor (Knn) and artificial neural network (ANN) in predicting the occurrence of the fraud.

Mohammed Azhan et al .[5] In general, fraudulent activities are always intended to cause financial detriment to the second party. With the aggrandizement of digital money in various countries, the fraudulent activities will be even more increased. The proposed research work discusses more about the different fraudulent activities associated with credit cards. While all of them cannot be dealt simultaneously, this research work discusses how Machine Learning and Neural Networks can be used to determine the potential fraudsters by referring to their previous mistakes and details of previous fraudsters. Machine Learning algorithms such as Multinomial Naive Bayes, Random Forest Regression, Logistic Regression, Support Vector Machine and a basic Neural Network are also used.

Randhawa et al.[6] Credit card fraud is a serious problem in financial services. Billions of dollars are lost due to credit card fraud every year. There is a lack of research studies on analyzing real-world credit card data owing to confidentiality issues. In this paper, machine learning algorithms are used to detect credit card fraud. Standard models are first used.To evaluate the model efficacy, a publicly available credit card data set is used. Then, a real-world credit card data set from a financial institution is analyzed.

Dhankhad et al. [7]The goal of data analytics is to delineate hidden patterns and use them to support informed decisions in a variety of situations. Credit card fraud is escalating significantly with the advancement of the modernized technology and become an easy target for fraudulent. Credit card fraud is a severe problem in the financial service and costs billions of a dollar every year. The design of fraud detection algorithm is a challenging task with the lack of real-world transaction dataset because of confidentiality and the highly imbalanced publicly available datasets. In this paper, we apply different supervised machine learning algorithms to detect credit card fraudulent

transaction using a real-world dataset.

Alaiselvi et al.[8] An infrastructure build in the neural network platform is reliable to detect the fraudulence in credit card system for transaction. The issues resulting from the fraud in credit card transaction may involve a number of customers who drift their habits evolve and fraudsters who change their strategies over time. The vast majority of learning algorithms that have been proposed for fraud detection rely on assumptions that hardly hold in a real-world fraud-detection system.

Bakshi et al.[9] Credit Card Fraud Detection is a consistently developing threat with far outcomes in the money related industry. With the quick improvement of electronic business, the quantity of exchanges by credit cards are expanding quickly.To distinguish misrepresentation conduct, bank and credit card organizations are utilizing different techniques for information mining, for example, data based mining, neural system, fuzzy bunching approach, hidden markov model or cross context approach of these strategies.

Mishra et al.[10] Nowadays, as internet speed has increased and the prices of mobile have decreased very much in past few years. Also the data prices too are very much affordable to most of the people.Analyzing various classification techniques using various metrics for judging various classifiers. The model aims at improving fraud detection rather than misclassifying a genuine transaction as fraud.

Kho et al.[11] The proliferation of the EMV (Europay-MasterCard-VISA) chip card design in the credit card business mostly resolved the problem posed by the old Magnetic stripe card technology. However, several papers are starting to question the design and implementation of the EMV. The paper is suggesting that a detection model must be available to capture the possible anomalous transactions - a fallback in case the technology will fail. Several classifiers were evaluated during the model creation however only the Random Tree and J48 yielded the highest accuracy value of 94.32 percentageand 93.50 percentage respectively.

Kavitha et al.[12]Fraud detection in credit card transactions has several major challenges including the huge volume and high velocity of the transactions, data imbalance and frequent change in the fraud patterns.It presents a real-time tree based meta-classifier TBMC that can be used to identify fraudulent transactions in huge imbalanced data. The developed meta-classifier based model operates based on predictions in two levels. The first level performed by Random Forest classifier, and the second by an ensemble created with Decision Trees and Gradient Boosted Trees.

Chapter 3

Methodology

Use of credit card is increased day by day.As a consequance increasing the fraud cases. so that problem can prevent by using some machine learning algorithms and methods.First load the data set and split the data fo training and testing.After the testing data can be preprocessed using different techniques.They are Random over sampler,SMOTE,ADASYN.After preprocessing that can be cross validated using repeatedKfolds and StratifiedKfolds. Comparing best method is choosen. Using machine learning algorithm such as Random forest and XGBoost.After implementing slgorithm the model can be run and predict the accuracy.Finally choose the best one

3.1 System Architecture

The system architectural design is used to abstract the overall outline of the software system and the relationships, constraints, and boundaries between components. It is an important tool as it provides an overall view of the physical deployment of the software system.

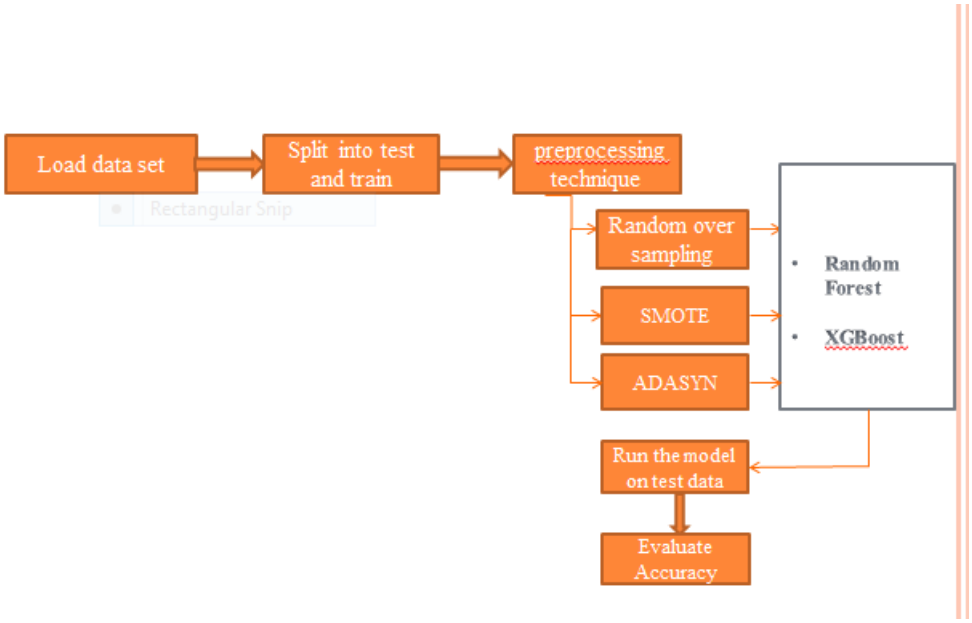

Figure 3.1: System Architecture

3.2 Data set

The dataset contains transactions made by credit cards in September 2013 by European cardholders.This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for0.172.

It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, . . . V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for examplededependant cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

3.3 Data Preprocessing

Data preprocessing is an integral step in Machine Learning as the quality of data and the useful] information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we preprocess our data before feeding it into our model. Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner.The data set has been pre-processed to handle various situations like attributes with missing data, attributes having no values or attributes having the value of and NA's. Misleading data like having different datatype or the format of the feature is different from what is other than required has also been taken into consideration. Missing values are replaced with the mode of the feature if it is categorical data and median of the feature if it is a continuous data. For continuous data median is used because if the data has outlier the mean will be huge and if we remove that outlier it may give very less mean for that reason replace the NA's of continuous data using median.

- **Random Over Sampler)**

  Random oversampling involves randomly selecting examples from the minority class, with replacement, and adding them to the training dataset. Random undersampling involves randomly selecting examples from the majority class and deleting them from the training dataset.They are referred to as "naive resampling" methods because they assume nothing about the data and no heuristics are used. This makes them simple to implement and fast to execute, which is desirable for very large and complex datasets.importantly, the change to the class distribution is only applied to the training dataset. The intent is to influence the fit of the models. The resampling is not applied to the test or holdout dataset used to evaluate the performance of a model Generally, these naive methods can be effective, although that depends on the specifics of the dataset and models involved

- **SMOTE**

  Imbalanced classification involves developing predictive models on classification datasets that have a severe class imbalance. The challenge of working with imbalanced datasets is that most machine learning techniques will ignore, and in turn have poor performance on, the minority class, although typically it is performance on the minority class that is most important. One approach to addressing imbalanced datasets is to oversample the minority class. The simplest approach involves duplicating examples in the minority class, although these examples don't add any new information to the model. Instead, new examples can be synthesized from the existing examples. This is a type of data augmentation for the minority class and is referred to as the Synthetic Minority Oversampling Technique, or SMOTE for short. SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line. Specifically, a random example from the minority class is first chosen. Then k of the nearest neighbors for that example are found (typically k=5). A randomly selected neighbor is chosen and a synthetic example is created at a randomly selected point between the two examples in feature space SMOTE first selects a minority class instance a at random and finds its k nearest minority class neighbors. The synthetic instance is then created by choosing one of the k nearest neighbors b at random and connecting a and b to form a line segment in the feature space. The synthetic instances are generated as a convex combination of the two chosen instances a and b

- **ADASYN**

  ADASYN (Adaptive Synthetic) is an algorithm that generates synthetic data, and its greatest advantages are not copying the same minority data, and generating more data for "harder to learn" examples.The essential idea of ADASYN is to use a weighted distribution for different minority class examples according to their level of difficulty in learning, where more synthetic data is generated for minority class examples that are harder to learn compared to those minority examples that are easier to learn. As a result, the ADASYN approach improves learning with respect to the data distributions in two ways

  - reducing the bias introduced by the class imbalance

– adaptively shifting the classification decision boundary toward the difficult examples. Simulation analyses on several machine learning data sets show the effectiveness of this method across five evaluation metrics.

3.4 Build Model

The dataset has been made as a selected features.The selected data has been partitioned into two parts as training data and testing data. The training data has been used to build a machine learning model and the testing data has been used to validate the model.The model which was build on the training data has been tested on the test data to validate the performance of model based on the accuracy.

Build two different model for calculating performance, Random forest model , and XG-Boost model.

1.Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. In Eq.(3.3) uses the class and probability to determine the Gini(G) of each branch on a node, determining which of the branches is more likely to occur. Here, pi represents the relative frequency of the class you are observing in the dataset and c represents the number of classes.

$$G = 1 - \sum (p_i)^2 \tag{3.1}$$

$$Entropy = \sum_{i=1}^{c} -(p_i * log_2 p_i) \tag{3.2}$$

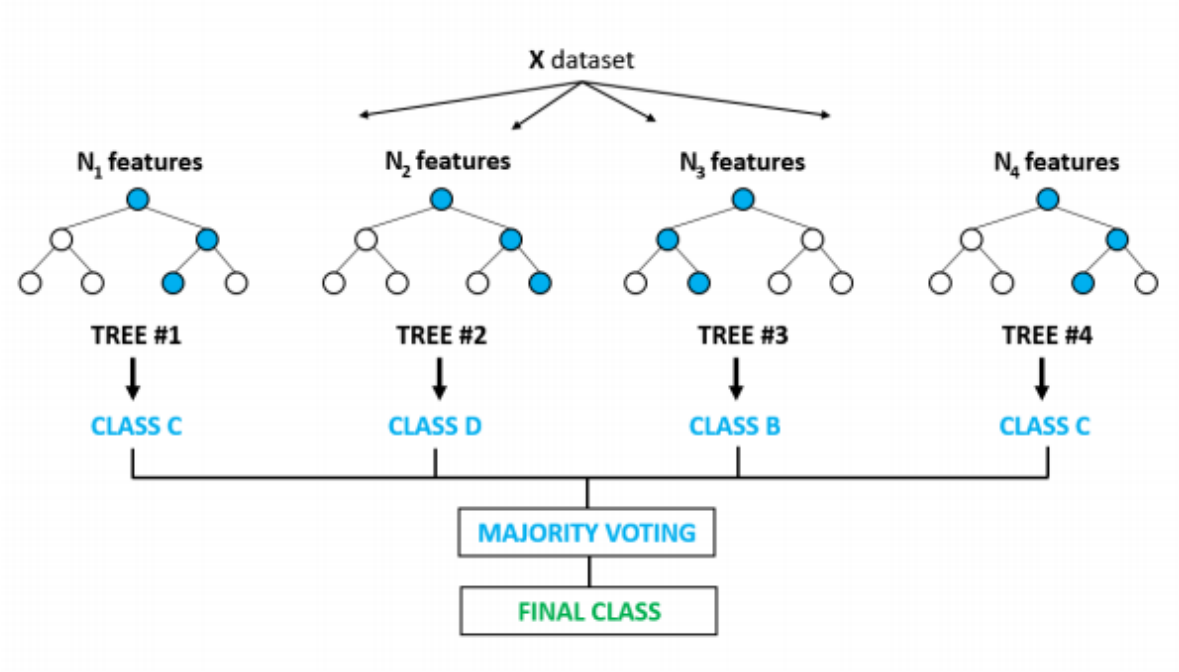

Figure 3.2: Random Forest classifier.

2.XGBoost

Extreme Gradient Boosting (XGBoost) is a scalable and improved version of the gradient boosting algorithm (terminology alert) designed for efficacy, computational speed and model performance. It is an open-source library and a part of the Distributed Machine Learning Community. XGBoost is a perfect blend of software and hardware capabilities designed to enhance existing boosting techniques with accuracy in the shortest amount of time. Here's a quick look at an objective benchmark comparison of XGBoost with other gradient boosting algorithms trained on random forest with 500 trees.

Boosting:

N new training data sets are formed by random sampling with replacement from the original dataset, during which some observations may be repeated in each new training data set. The observations are weighted and therefore some of them may get selected in the new datasets

more often.

The boosting ensemble technique consists of three simple steps:

- An initial model F0 is defined to predict the target variable y. This model will be associated with a residual (y – F0)
- A new model h1 is fit to the residuals from the previous step.
- Now, F0 and h1 are combined to give F1, the boosted version of F0. The mean squared error from F1 will be lower than that from F0:

$$F_1(x) < -F_0(x) + h_1(x) \tag{3.3}$$

To improve the performance of F1, we could model after the residuals of F1 and create a new model F2:

$$F_2(x) < -F_1(x) + h_2(x) \tag{3.4}$$

This can be done for 'm' iterations, until residuals have been minimized as much as possible:

$$F_m(x) < -F_{m-1}(x) + h_m(x) \tag{3.5}$$

Bagging:

N new training data sets are formed by random sampling with replacement from the original dataset, where each observation has the same probability to appear in a new data set.

Given a set of n independent observations Z1, ..., Zn, each with variance 2, the variance of the mean Z of the observations is given by 2/n.This means that averaging a set of observations reduces variance. Hence, by taking many training sets from the population, building a separate prediction model using each training set and averaging the resulting predictions, we can reduce the variance and consequently increase the prediction accuracy of the method. In particular, we calculate ˆf1(x),ˆf2(x), ...,ˆfB(x) using B separate training sets, and average them in order to obtain a single low-variance statistical model, given by:

$$\hat{f}_{avg} = 1/B \sum_{b=1}^{B} \hat{f}^b(x) \tag{3.6}$$

However, in most use cases, it might not be possible to access multiple training sets. That is where the bootstrap method becomes useful. Bootstrap consists in taking repeated samples from the original training data set. It generates B different bootstrapped training data sets. Then, the model is fit on the both bootstrapped training set, resulting in the prediction ˆfb(x). All the predictions are averaged to obtain:

$$\hat{f}_{bag} = 1/B \sum_{b=1}^{B} \hat{f}^* b(x) \tag{3.7}$$

Bagging can easily be applied to a classification problem, to predict a qualitative outcome Y . For a given test observation, each B tree predict a class and we choose the overall prediction as the most commonly occurring class among B predictions.

Gradient Boosting:

It is an additive and sequential model where trees are grown in sequential manner which converts weak learners into strong learners by adding weights to the weak learners and reduce weights of the strong learners. So each tree learns and boosts from the previous tree grown.

We would like to do the same thing but here our solution must be a tree. Also importantly, we don't want to simply minimize loss on the training set but generalize to new data.A potential solution is to induce a tree at the mth iteration whose predictions tm are as close as possible to the negative gradient.

$$\theta_m = argmin_\theta \sum_{i=1}^{N} (_{gim} - T(x_i;\theta))^2 \tag{3.8}$$

1. Initialize $f_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{N} L(y_i, \gamma)$
2. For $m = 1$ to M:
   - For $i = 1, 2, ..., N$ compute

$$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}$$

   - Fit a regression tree to the targets r_{im} giving terminal regions $R_{jm}, j = 1, 2, ..., J_m$
   - For $j = 1, 2, ..., J_m$ compute

$$\gamma_{jm} = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{jm}} L(Y - i, f_{m-1}(x_i) + \gamma)$$

   - Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} \mathbf{1}(x \in R_{jm})$
3. Output $\hat{f}(x) = f_M(x)$

Figure 3.3: Algorithm of Gradient Boost

Algoritham of XGBoost

Here are simple steps you can use to crack any data problem using xgboost Algorithm:

Step 1: Load all the libraries

Assuming you have a working SciPy environment, XGBoost can be installed easily using pip.

Step 2: Load the dataset

we will load the data from file and prepare it for use for training and evaluating an XGBoost model.Next, we can load the CSV file as a NumPy array using the NumPy function loadtext().

Step 3: Train the XGBoost Model

XGBoost provides a wrapper class to allow models to be treated like classifiers or regressors in the scikit-learn framework.

This means we can use the full scikit-learn library with XGBoost models.

The XGBoost model for classification is called XGBClassifier. We can create and and fit it to our training dataset. Models are fit using the scikit-learn API and the model.fit() function. Parameters for training the model can be passed to the model in the constructor.

Step 4: Make Predictions with XGBoost Model

We can make predictions using the fit model on the test dataset.

To make predictions we use the scikit-learn functio model.predict().

First, the dataset will be split into two parts. The first part is used for parameter tuning, while the second part will be used for training and testing the developed models.In the second step, the parameters of XGBoost are tuned .This is very important step because XGBoost is very sensitive to the initial values of its many parameters.The next step is the oversampling method. After the oversampling step, the XGBoost is trained using the oversampled data and tested on the testing data which is not oversampled. • After applying the cross validation process, the performance of the XGBoost is evaluated using the common classification metrics which are : the accuracy rate, precision, recall, and F1-measure.

3.5 Software requirements and specification

The software used for the project:

• Python

• Google Colab

1.**PYTHON**

Python is an object-oriented language control users to manage and management data structures or objects to make and run programs. Everything in Python is, in fact, top-notch. All objects, data types, functions,methods, and classes take an equal position in Python. Programming languages are created to satisfy the requirements of programmers and users for an efficient tool to develop applications that impact lives, lifestyles, economy,and society.

they assist build lives better by increasing productivity, enhancing communication, and rising potency. Languages die and become obsolete once they fail to live up to expectations and are replaced and superseded by languages that are more powerful. Python programming language artificial language that has stood the test of time and has remained relevant across industries and businesses and among programmers, and individual users. it's a living, thriving, and extremely helpful language that's extremely recommended as a primary programming language for those that want to dive into and experience programming.

Browser incompatibilities

When a user receives a page which incorporates JavaScript, the JavaScript interpreter of his browser kicks in and tries to execute the script. Currently,the main downside here is that the assorted browsers each use their own interpreter, which generally browser vendors have chosen to not implement a bit of JavaScript. Their reasons were typically associated with a business advantage over the competitors. therefore the dreaded browser incompatibilities.In addition, every new browser version understands more JavaScript and permits more and more components of the HTML page to be modified by scripts. This results in even more incompatibilities. It is best to resolve compatibility issues on a case by case basis. In fact, most pages on this website have been written precisely thanks to browser incompatibilities. So scan on to know more. But I warn you: you need to digest quite a ton of information. thus it is best to unravel the problem at hand and leave the rest of the knowledge alone till you need it.

- Python is Interpreted : Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive : You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented : Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

- Python is a Beginners Language : Python is a great language for the beginnerlevel programmers and supports the development of a wide rangrnditemizee of applications from simple text processing to WWW browsers to games.

- an easy and intuitive language just as powerful as those of the major competitor
- open source, so anyone can contribute to its development
- code that is as understandable as plain English

2.**Google Colab**

Google Colab was developed by Google to provide free access to GPU's and TPU's to anyone who needs them to build a machine learning or deep learning model. Google Colab can be defined as an improved version of Jupyter Notebook.Jupyter Notebook is an application that allows editing and running Notebook documents through a web browser or an Integrated Development Environment(IDE).
Google Colab provides tons of exciting features that any modern IDE offers, and much more. Some of the most exciting features are listed below.

- Interactive tutorials to learn machine learning and neural networks.
- Write and execute Python 3 code without having a local setup.
- Execute terminal commands from the Notebook. Import datasets from external sources such as Kaggle.
- Save your Notebooks to Google Drive.
- Import Notebooks from Google Drive.
- Free cloud service, GPUs and TPUs.
- Integrate with PyTorch, Tensor Flow, Open CV

Chapter 4

Result Analysis

Accuracy testing is used as a measure of the perfomance of our classification model.For the testing of the accuracy we are using confusion matrix.

4.1 Result Analysis

Accuracy testing is used as a measure of the perfomance of our classification model.For the testing of the accuracy we are using confusion matrix.

Confusion matrix

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. It allows easy identification of confusion between classes e.g. one class is commonly mislabeled as the other. Most performance measures are computed from the confusion matrix.

The important terms included in confusion matrix are as following:

- True Positive (TP) :Observation is positive, and is predicted to be positive.
- False Negative (FN) : Observation is positive, but is predicted negative.
- True Negative (TN) : Observation is negative, and is predicted to be negative.
- False Positive (FP) : Observation is negative, but is predicted positive.

- 0: No
- 1: Yes

The confusion matrix represent like this

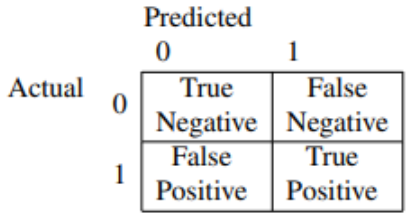

Figure 4.1: Confusion matrix

- Precision

Precision is the ability of a classifier not to label an instance positive that is actually negative. For each class it is defined as the ratio of true positives to the sum of true and false positives.

Precision = Accuracyofpositivepredictions

Precision = TP/TP+FP

- Recall

Recall is the ability of a classifier to find all positive instances. For each class it is defined as the ratio of true positives to the sum of true positives and false negatives. Said another way,"for all instances that were actually positive, what percent was classified correctly?"

Recall = F raction of positives that were correctly identified

Recall =TP /TP + FN

- f1 Score

The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. Generally speaking, F1 scores are lower than accuracy measures as they embed precision and recall into their computation. As a rule of thumb, the weighted average of F1 should be used to compare classifier models, not global accuracy.

f1Score = Whatpercentofpositivepredictionswerecorrect

f1Score =2 (Recall * precision)/ Recall + Precision

- Support

Support is the number of actual occurrences of the class in the specified dataset. Imbalanced support in the training data may indicate structural weaknesses in the reported scores of the classifier and could indicate the need for stratified sampling or re-balancing. Support doesn't change between models but instead diagnoses the evaluation process.

- Accuracy

Accuracy =Sum of correct classification/ Total no .of classification

Accuracy =T P + T N/ T P + T N + F P + F N

The Proposed system for the credit card fraud detection has an classification report and accuracy rate. Used Random Forest,and XGBoost model for the Credit Card Fraud Detection using Credit Card dataset.

Following figures(4.2,4.3,4.4 shows the result of r confusion matrix using different technology

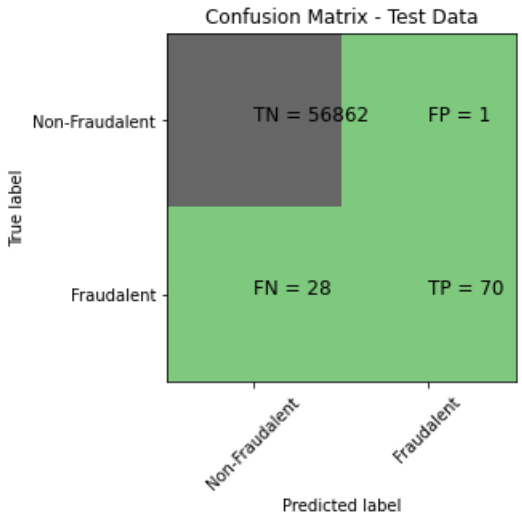

Figure 4.2: Result of confusion matrix using Random Over Sampler

Following figure shows(4.5) the result of Random forest by precision,recall,f1 score and support

Figure (4.6,4.7,4.8) shows the result ofXGBoost confusion matrix using different technology

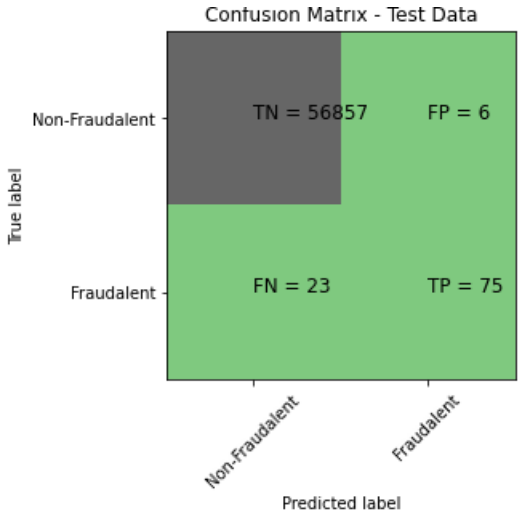

Figure 4.3: Results of Random forest confusion matrix using SMOTE

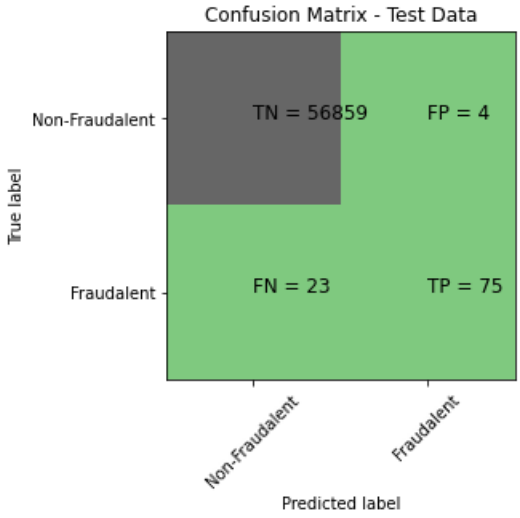

Figure 4.4: Results of Random forest confusion using ADASYN

Predicted label

classification Report

	precision	recall	f1-score	support
0	1.00	1.00	1.00	56863
1	0.97	0.74	0.84	98
accuracy			1.00	56961
macro avg	0.99	0.87	0.92	56961
weighted avg	1.00	1.00	1.00	56961

Figure 4.5: Results of Random forest by precision, recall, f1 score metrics,support

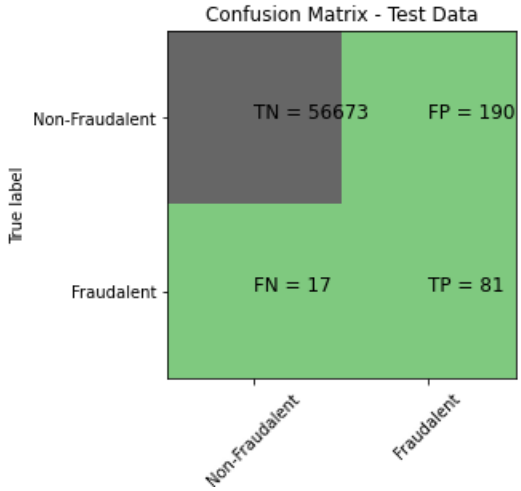

Figure 4.6: Results of XGBoost confusion matrix using random over sampler

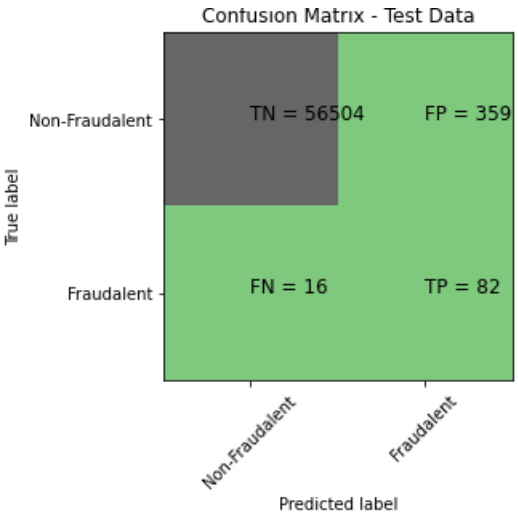

Figure 4.7: Results of XGBoost confusion matrix using SMOTE

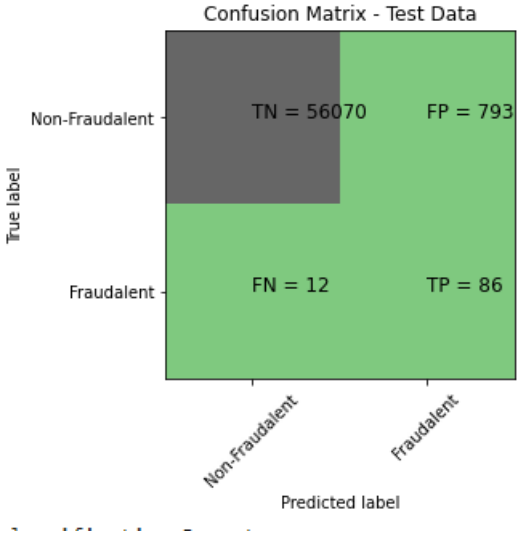

Figure 4.8: Results of XGBoost confusion matrix using ADASYN

classification Report

	precision	recall	f1-score	support
0	1.00	1.00	1.00	56863
1	0.94	0.74	0.83	98
accuracy			1.00	56961
macro avg	0.97	0.87	0.91	56961
weighted avg	1.00	1.00	1.00	56961

Figure 4.9: Results of XGBoost by precision, recall, f1 score metrics,support

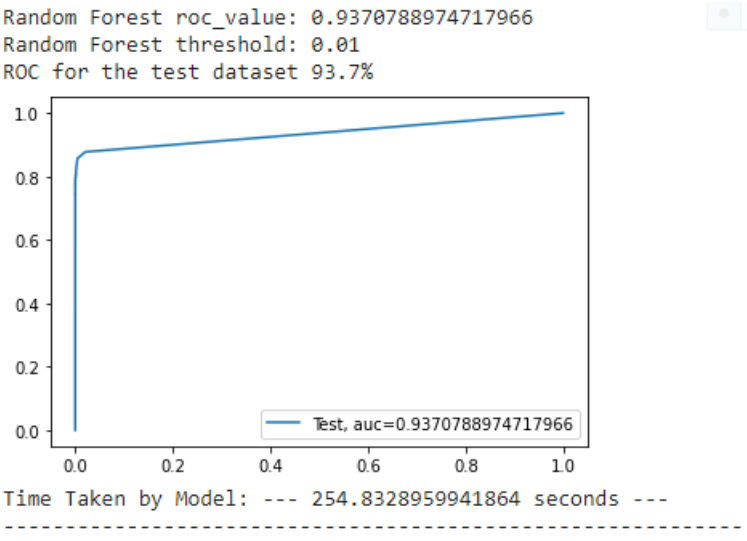

Figure 4.10: Random forest ROC curve

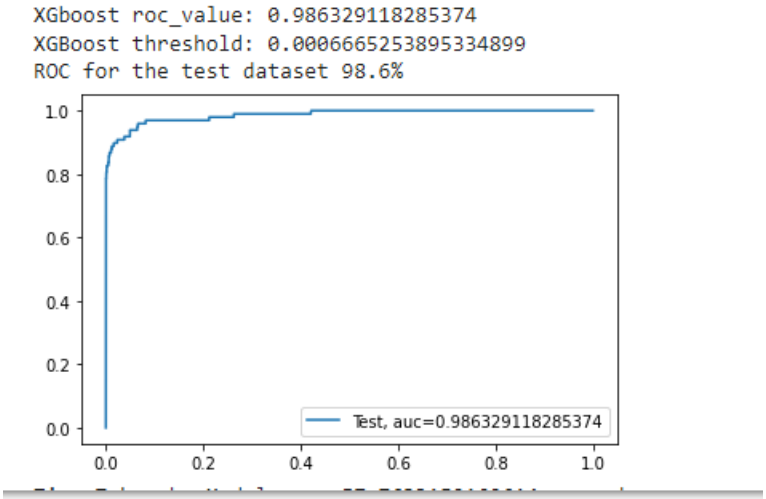

Figure 4.11: XGBoost ROC curve

Chapter 5

Conclusion

From the comparitive study among two different models,proposed XGBoost model with random oversampling with stratifiedKFold cross validation shows the best accuracy to detect the fraudlent cases

5.1 Advantages

The key Features are :

- The system provide more accuracy and efficiency
- Parellel Processing
- Handling Missing Values

5.2 Future Enhancement

Future work may focus on different areas, starting by proposing data preprocessing techniques to overcome the drawback of the missing values. Additionally, different methods of feature selection and extraction should be investigated in the credit card domain and to

determine its impact on prediction accuracy. An investigation of the most appropriate hybrid model among the state-of-the-art machine learning algorithms to determine the main concern for future studies.

REFERENCES

[1] Thennakoon, Anuruddha; Bhagyani, Chee; Premadasa, Sasitha; Mihiranga, Shalitha; Kuruwitaarachchi, Nuwan (2019). [IEEE 2019 9th International Conference on Cloud Computing, Data Science Engineering (Confluence) - Noida, India (2019.1.10-2019.1.11)] 2019 9th International Conference on Cloud Computing, Data Science Engineering (Confluence) - Real-time Credit Card Fraud Detection Using Machine Learning. , (), 488–493.

[2] Varmedja, Dejan; Karanovic, Mirjana; Sladojevic, Srdjan; Arsenovic, Marko; Anderla, Andras (2019). [IEEE 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH) - East Sarajevo, Bosnia and Herzegovina (2019.3.20-2019.3.22)] 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH) - Credit Card Fraud Detection - Machine Learning methods

[3] Awoyemi, John O.; Adetunmbi, Adebayo O.; Oluwadare, Samuel A. (2017). [IEEE 2017 International Conference on Computing Networking and Informatics (ICCNI) - Lagos, Nigeria (2017.10.29-2017.10.31)] 2017 International Conference on Computing Networking and Informatics (ICCNI) - Credit card fraud detection using machine learning techniques: A comparative analysis. , (), 1–9.

[4] Pillai, Thulasyammal Ramiah; Hashem, Ibrahim Abaker Targio; Brohi, Sarfraz Nawaz; Kaur, Sukhminder; Marjani, Mohsen (2018). [IEEE 2018 Fourth International Conference on Advances in Computing, Communication Automation (ICACCA) - Subang Jaya, Malaysia (2018.10.26-2018.10.28)] 2018 Fourth International Conference on Advances in Computing, Communication Automation (ICACCA) - Credit Card Fraud Detection Using Deep Learning Technique. , (), 1–6

[5] Mohammed Azhan;Shazli Meraj; (2020). Credit Card Fraud Detection using Machine Learning and Deep Learning Techniques . 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), (),

[6] .Randhawa, Kuldeep; Loo, Chu Kiong; Seera, Manjeevan; Lim, Chee Peng; Nandi, Asoke K. (2018). Credit card fraud detection using AdaBoost and majority voting. IEEE Access, (), 1–1.

[7] Dhankhad, Sahil; Mohammed, Emad; Far, Behrouz (2018). [IEEE 2018 IEEE International Conference on Information Reuse and Integration (IRI) - Salt Lake City, UT, USA (2018.7.6-2018.7.9)] 2018 IEEE International Conference on Information Reuse and Integration (IRI) - Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study. , (), 122–125.

[8] Kalaiselvi, N; Rajalakshmi, S; Padmavathi, J; Karthiga, Joyce. B. (2018). [IEEE 2018 International conference on computation of power, energy, Information and Communication (ICCPEIC) - Chennai, India (2018.3.28-2018.3.29)] 2018 Internat2018 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC)ional conference on computation of power, energy, Information and Communication (ICCPEIC) - Credit Card Fraud Detection Using Learning to Rank Approach. , (), 191–196.

[9] Bakshi, Sonali (2018). [IEEE 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) - Palladam, India (2018.8.30-2018.8.31)] 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference on - Credit Card Fraud Detection : A classification analysis. , (), 152–156.

[10] Mishra, Ankit; Ghorpade, Chaitanya (2018). [IEEE 2018 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) - Bhopal, India (2018.2.24-2018.2.25)] 2018 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) - Credit Card Fraud De-

tection on the Skewed Data Using Various Classification and Ensemble Techniques. , (), 1–5.

[11] Kho, John Richard D.; Vea, Larry A. (2017). [IEEE TENCON 2017 - 2017 IEEE Region 10 Conference - Penang (2017.11.5-2017.11.8)] TENCON 2017 - 2017 IEEE Region 10 Conference - Credit card fraud detection based on transaction behavior. , (), 1880–884.

[12] Kavitha, M.; Suriakala, M. (2017). [IEEE 2017 International Conference on Inventive Computing and Informatics (ICICI) - Coimbatore (2017.11.23-2017.11.24)] 2017 International Conference on Inventive Computing and Informatics (ICICI) - Real time credit card fraud detection on huge imbalanced data using meta-classifiers. , (), 881–887.