Investigation on Fraud Detection Using Machine Learning Algorithms

A project report submitted in partial fulfillment of the requirements for B.Tech. Project

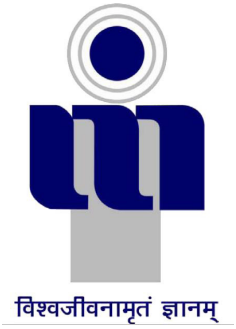
Integrated Post Graduation.

by

Vishal Jain (2017IMT- 090)

under the supervision of

Dr. Vinal Patel



ABV-INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT
GWALIOR - 474015

2020

CANDIDATES DECLARATION

I hereby certify that the work, which is being presented in the report, entitled **An Investigation on Fraud Detection Using Machine Learning Algorithms**, in partial fulfillment of the requirement for the award of the Degree of **Integrated Post Graduation** and submitted to the institution is an authentic record of my own work carried out during the period *August 2020* to *November 2020* under the supervision of **Dr. Vinal Patel** . I also cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.

Date: Signatures of the Candidates

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Date: Signatures of the Research Supervisors

ABSTRACT

In the past few years machine learning has played a crucial role in resolving various problems of the businesses like spam detection, recommendation, object detection etc. Fraud detection has been a painful task for the banking, medical, and commerce industry. Hackers and criminals are continuously working on new ways to find loopholes in the system as a result it is becoming more and more difficult for the companies to authenticate their transaction. According to the Fraud Benchmark Report by cyber source 83% American businesses are conducting manual reviews, and on an average basis, they review 29 % of orders manually. India is witnessing a drastic change to online transactions [1]. It was ranked in top 5 countries with respect to credit card fraud and cases related to debit/credit card fraud have surged to 42% in 2017 according to the Hindustan Times. According to RBI data, the number of credit cards rose by a quarter upto 38 million in the last 12 months ending in May, while the number of debit cards jumped 17% to over 925 million in the same period [2].

India is considered to be an emerging market for credit card market since average credit card that an indian user own is very less compared to developed markets like USA which was 3.7 cards in 2009. The concept of cashback, instant discount and reward points play a crucial in their growth story. But the down side is that it is just a form of debt. The national average credit card debt in USA sits around 5,331 dollars and it's growing, hence frauds in such scenario puts burden on both individual and financial institution. According to federal reserve of USA 70% of united states population use credit card and 34% of them carries more than one credit card. Even after the best efforts of companies these frauds cases are on rise. Visa, Mastercard, American Express and discover are the four major players. In 2018 24.26 dollars billion was lost due to credit card frauds all around the world. USA is the most prone country with 38.6% cases in 2018 and identity theft was the third largest reason behind it. Lack of consumer awareness is also a major cause, people are often contacted on calls or mails for various reasons like overdue loans or prize scams. Hackers often use skimmers to get information and later use them to produce fake cards. If the card falls in unsafe hands than it can be used for frauds. Survey revealed that age group of 30-50 is most prone to these frauds. Identity theft resource center in 2018 conducted survey on people to analyze health problem and it was observed that 84.1% suffers sleep problems, 77.3% increased stress levels and 85.71% felt worried angry and frustrated.

Keywords: Credit card, Fraud detection, Regression, Machine learning, Ensembling algorithm, Boosting algorithm, Evaluation metrics.

ACKNOWLEDGEMENTS

I would like to express my special thanks of gratitude to my teacher **Dr. Vinal Patel**, who gave me this wonderful opportunity to do project in machine learning and also helped me in completing my project. I am really thankful to him for the knowledge he shared as i learned various new things and aspects of machine learning algorithms. After weeks of learning about machine learning algorithms, i am aware about their pros and cons. Throughout the whole process his expert advice and encouragement was precious.

Finally i would like to thank my friends and family for their constant support.

Many thanks for this opportunity.

(Vishal Jain)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

TP	True Positives
TN	True Negatives
FN	False Negatives
FP	False Positives
MCC	Matthew correlation coefficient
Log Reg	Logistic Regression
minmax	minmaxscaler
BBN	Bayesian belief network
ANN	Artificial neural network
ROC	Roc-auc score

CHAPTER 1

INTRODUCTION

The initial sign of credit card fraud detection can be seen in 1994. Usually fraudsters stole them or found them lost, used fake names to carry on frauds. This was the most vulnerable time due to lack of authorization. Since then merchants enabled variety of mechanism to verify cardholder.

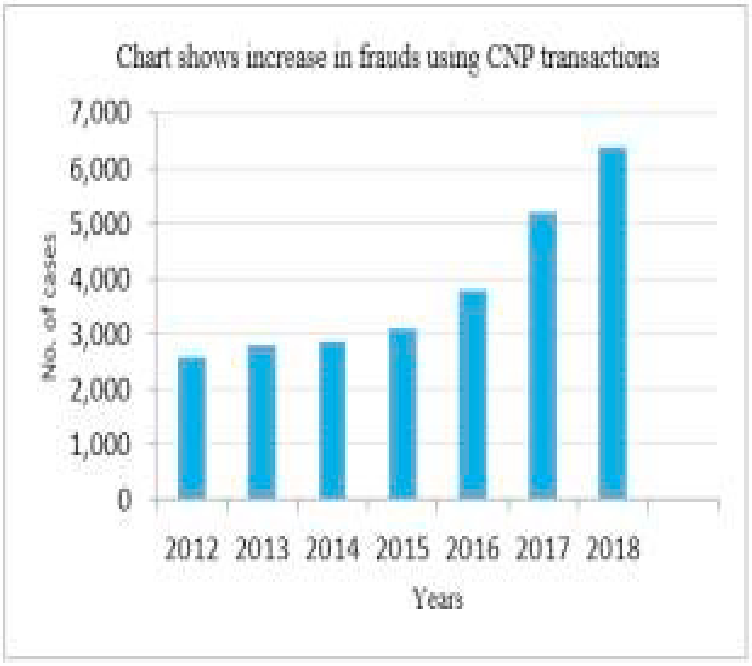


Figure 1.1: CNP Frauds Transaction

Then in 1996-97 This was the starting stage for online fraud detection now fraudsters can check status of stolen cards online. Now they have the power to attack merchant sites to acquire card details.
During 1998-99 as the internet evolved hackers used their social engineer skills, honeypots etc and other mechanism to receive order information and divert the details of it.

As the world is shifting towards automation and increase in online transactions since the introduction of ecommerce it becomes a challenging task to handle this task manually hence AI could play a key role in developing algorithms for fraud detection. Once the machine learning or deep learning models are trained on a set of input data can generalize the model to use in the real world. Once determined the model will produce an output on the basis of the feature set [3].

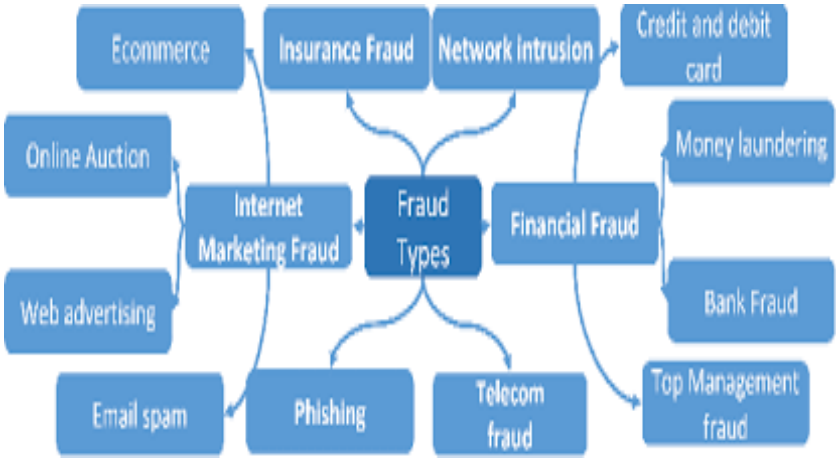


Figure 1.2: Frauds Taxonomy

1.1 Motivation

There are many problems with manual review in credit card fraud detection like it is costly, time-consuming and lead to high false positives which means denying a normal user for the transaction completely. By analyzing machine learning algorithms in this area, my main job is to reduce human time, computation and effort. Artificial Intelligence helps machines to identify and extend the importance of patterns in place of humans by visualizing and analyzing the data on the basis of dataset provided and taking appropriate action on the basis of feature set in a short duration of time.
In today's world due to increase in online transaction especially due to ecommerce, huge amounts of data is generated each day and as a result it becomes impossible to analyze through traditional methods like rule based method. AI makes machines better than humans in processing the data. Traditional approaches like rule based and traditional approaches have failed due to increase in data and more variation in types of transaction. Also one of the drawback is that due to high false positive rate it leads to loss of legitimate customers and high unbalance dataset is still a matter of concern and also a wide area of research in machine learning.

1.2 Objective

The aim of this work is to study and analyze how machine learning algorithms like boosting and ensembling along with data balancing algorithms like smote, adasyn and allknn perform on fraud detection problems on unbalance and balance datasets.

1.3 Background

There are many types of fraud and the scope of this project lies within financial fraud like mortgage loan, loan eligibility etc. The project proposes analysis of machine learning algorithms and how class balancing algorithms may affect their performance. This work majorly focuses on credit card fraud detection as fraud is a broad topic for a single machine learning to learn.
Below are details of some algorithms that have been tested:

- Logistic Regression - It's a linear algorithm that's mostly used to make a binary classification, it uses sigmoid function at it's core and outputs the probability for each class [4].

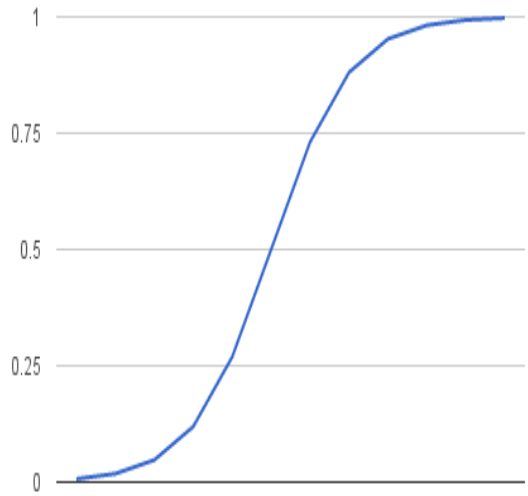


Figure 1.3: Logistic Regression

- Decision Tree - It's a powerful supervised learning algorithm that is mainly used for classification and uses tree architecture for prediction [5]. In decision tree the

data is splitted on each node or feature. Each node is the deciding splitting factor, branches represent the outcome of this node and leaf nodes represents the labels.
Entropy - It refers to randomness in given data. It lies between 0 and 1 and is given by:
H(s) = -P(+)log2(+) - -P(-)log2(-).
Now this entropy is calculated for every feature after every split and it selects the best feature on the basis of this split.
Gini Impurity - This is also somewhat similar to entropy and is used to as a criteria for splitting [6]. It is calculated as:
gini = 1 - [P(+) * P(+) + P(-) * P(-)]

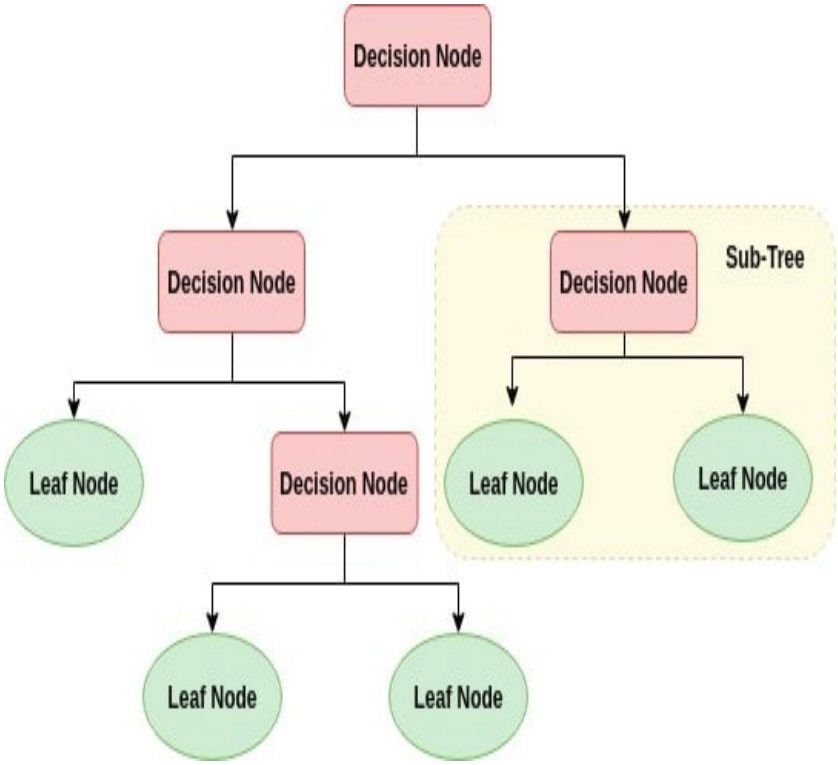


Figure 1.4: Decision Tree

- Random Forest - These are also supervised learning algorithm that uses ensembles of decision tree for prediction. The basic ides is that combining multiple learning models increases overall result [7]. In this method a sample of data is selected from original data and then a tree is built in which instead of selecting best splitting factor, random predictors are selected and from them the best splitting factor is selected.

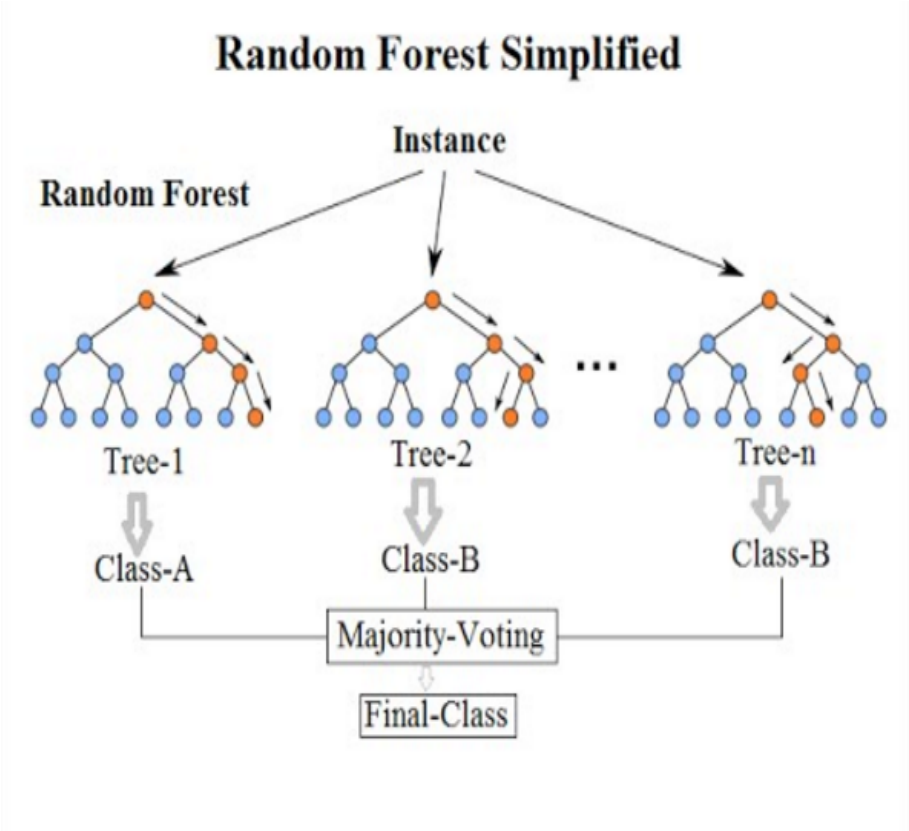


Figure 1.5: Random Forest

- Smote - SMOTE is an oversampling method which stands for synthetic minority oversampling technique which resolves class unbalancing by increasing the number of minority samples. In this method the new minority data points are created between the real minority data points. By generating these data points it helps the classifier to shift it's learning bias towards it [8].

- Boosting - These are also supervised learning algorithms that are used for classsification like adaboost [9], catboost, xgboost, and lgbm etc [10]. Most of the difference between these algorithms lies in the way they assign weights to each data point [11]. For example initially in some algorithms same weights are assigned are assigned for first classifier then in each iteration weight is increased for misclassified data points while feeding to next classifier, whereas in others the difference between prediction and ground truth is taken into account. LightGBM is also a tree based boosting and learning algorithm which grows vertically i.e, leaf-wise [12].

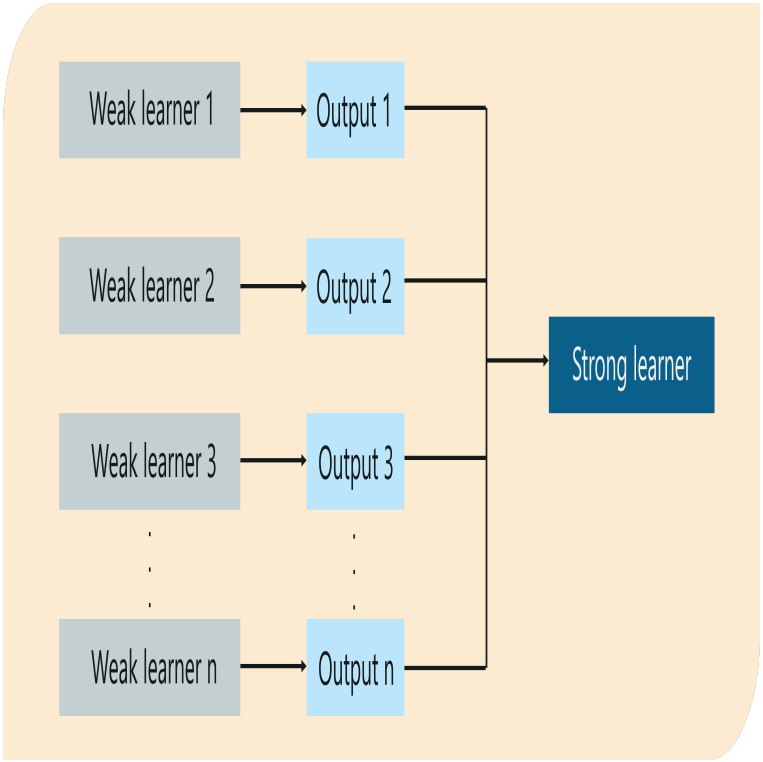


Figure 1.6: Boosting Algorithms

- Adasyn - Adasyn is also an oversampling method which stands for adaptive synthetic approach for imbalance learning. This take account for weighted distribution of data especially the minority class. It generates more data points for minority class which are harder to learn rather than minority classes that are easier to learn [13].

- Allknn - Allknn is an undersampling method which takes into account nearest neighbours. It adds only those data points which are misclassified by nearest neighbors to the majority class [14].

1.4 Deliverables

- To develop a model with high accuracy, recall, precision, f1-score, mcc and roc.
- Analyzing various machine learning algorithms like ensemblers, boosting and regression techniques etc.
- Using 20% dataset as test data in all machine learning algorithms.
- To study effect of smote, allknn and others on machine learning algorithms.

1.5 Salient Features

- Only supervised machine learning algorithms are used for analysis purpose.
- Decision trees take decisions at each step and random forest take account of various decsion trees while making a conclusion.
- Most of the algorithms used for the analysis requires pretraining on a dataset to generate an output.

CHAPTER 2

LITERATURE SURVEY

Many machine learning algorithms both supervised and unsupervised are used in real world scenarios. Supervised algorithms require accurate labelling of transactions as fraud or not in order from algorithm to work and are built to differentiate between legitimate transactions and previous known frauds. One of the issues is that the data is highly evolving and changing since a dataset may contain fraudster entry as legitimate since it fits the pattern of a legitimate user.

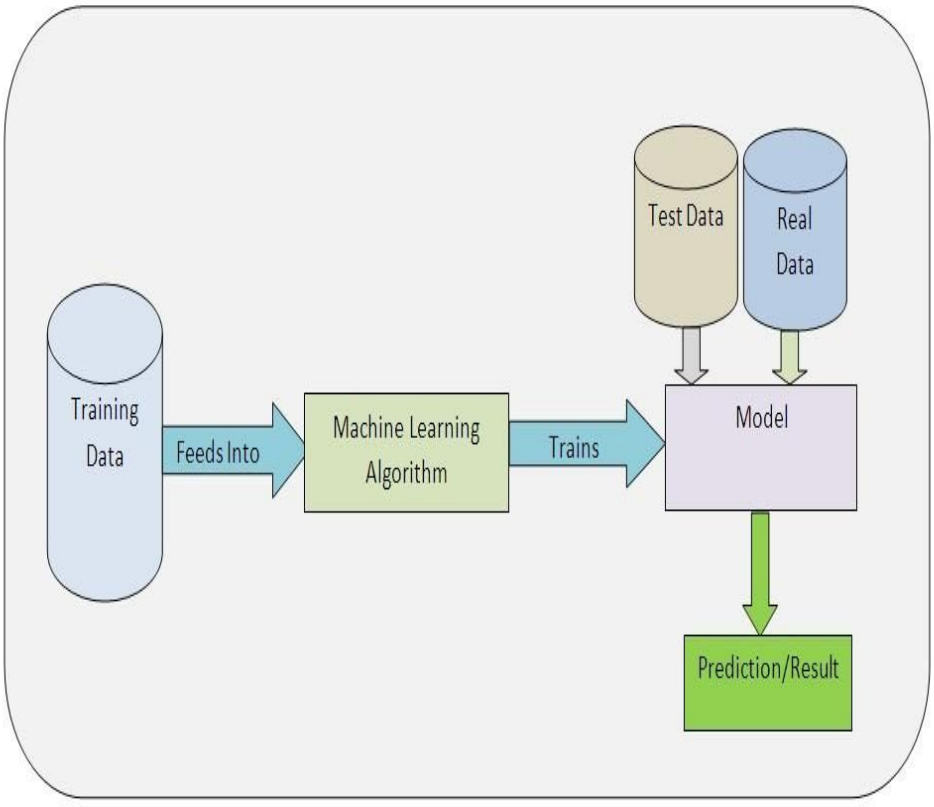


Figure 2.1: Machine learning pipeline

Cashless transactions have shown remarkable growth in recent years and have become a significant part in banking system and thus leading to increase in losses both for banks and users. Hence an efficient fraud detection system is necessary to maintain integrity of financial institutions. Large scale data mining can prove to be an efficient way to deal with it [15]. The paper describes scalable techniques that can be used to analyze massive data in an efficient, timely manner and combines various cost effective models. Hidden markov model (HMM) was also tested to develop credit card detection system, in which the model was initially trained on normal behaviour of a user and during the detection process if the probability is low then the transaction is considered as fraudulent [16]. Data unbalancing has always been a key issue in developing an efficient method, various approaches including neural network along with data mining have been studied [17]. Also the effect of BBN and ANN was studied and was found that bayesian gave better results. Big data analytics plays a key role since billions of transaction are made on daily basis, scarff integrated big data tools like kafka, spark to develop real-time fraud finder [18].

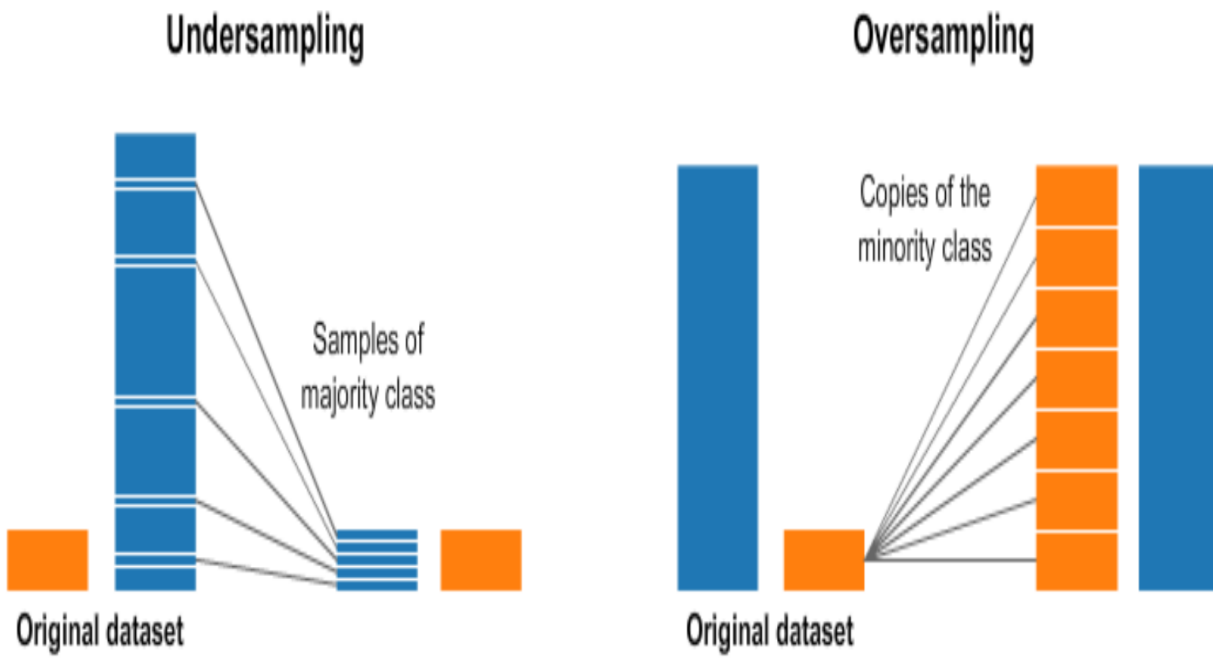


Figure 2.2: Data Undersampling and Oversampling

CHAPTER 3

SYSTEM DESIGN, PROJECT DESCRIPTION AND METHODOLOGY

3.1 Project Description

Credit cards are usually the physical payment card that are provided by customer's bank and allow to make payment in place of cash. The main advantage of credit card is that it provides time to their customer to repay in a defined duration of time. The task becomes more challenging since fraudsters use several measures to make a transaction look real making the task more difficult.

As advancement of technology leads to introduction of EMV cards which are safer than traditional ones but still vulnerable to cards not present frauds on higher rates [19]. There are various types of credit card frauds which includes:

- lost and stolen card fraud
- card not received fraud
- counterfeit card fraud
- cardholder bot present fraud
- card ID theft fraud

The task of credit card fraud detection becomes more difficult because there are more legitimate transactions compared to fraud ones making the dataset highly imbalanced for machine learning algorithms to work. The project focuses on the variation of credit card fraud detection techniques and analysis of machine learning algorithms that can be used to prevent these frauds.

CARD FRAUD TYPE ON UK-ISSUED CREDIT AND DEBIT CARDS	2012	2013	2014	2015	2016	2017	% CHANGE 16/17
Remote Purchase (CNP)	752,450	951,998	1,019,146	1,113,084	1,437,832	1,399,031	-3%
Counterfeit (skimmed/cloned)	98,555	101,109	99,279	86,021	108,597	84,861	-22%
Fraud on lost or stolen cards	113,162	138,967	133,943	143,802	231,164	350,066	51%
Card ID theft	24,287	30,718	26,542	33,566	31,756	29,139	-8%
Card non-receipt	9,053	9,125	9,302	10,719	11,377	10,905	-4%
TOTAL	997,507	1,231,917	1,288,212	1,387,192	1,820,726	1,874,002	3%

Figure 3.1: Frauds on UK issued cards in 2012-2017

3.1.1 Dataset Description

The dataset contains 284,807 transactions among which there are 492 i.e., 0.172% transactions are fraudulent transactions. It also contains transactions made by a card-holder in 2 days in month of september 2013 This dataset is highly unbalanced. Due to security reasons, most of the features in the dataset are transformed using principal component analysis (PCA). V1, V2, V3,..., V28 are PCA applied features and rest features include ‘time’, ‘amount’ and ‘class’ are non-PCA applied features.

Table 3.1: Dataset descriptions and their specifications

S.NO	Features	Specifications
1	V1,V2....V28	pca component
2	Time 3	time in sec
3	Amount	transaction amount
4	Class	0-not fraud, 1-fraud

3.2 Methodology

The study includes fraud detection problems like credit card fraud detection etc and describe how machine learning algorithms perform on them.

- The study aims to answer questions like how we can deal with unbalanced dataset in fraud detection.
- I plan to analyze how algorithms like SMOTE, ADASYN etc are useful in balancing the dataset and how they affect the final result.

- I plan to analyze how various machine learning ensembling, boosting on unbalance and balance dataset.
- The metrics used for the evaluation will be accuracy, precision, recall, f1-score, mcc and roc .

3.2.1 Metrics used

In my current progress i have used following metrices for evaluation, these parameters are used as base parameters for evaluation [20] [21]:

- Accuracy = TP + TN / TP + TN + FP + FN
- Precision = TP / TP + FP
- Recall = TP / TP + FN
- F1score = 2 * precision * recall / precision + recall
- Mcc = TP * TN - FP * FN / sqrt((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN))
- Roc = Area under curve between false positive rate and true positive rate

3.3 System Design

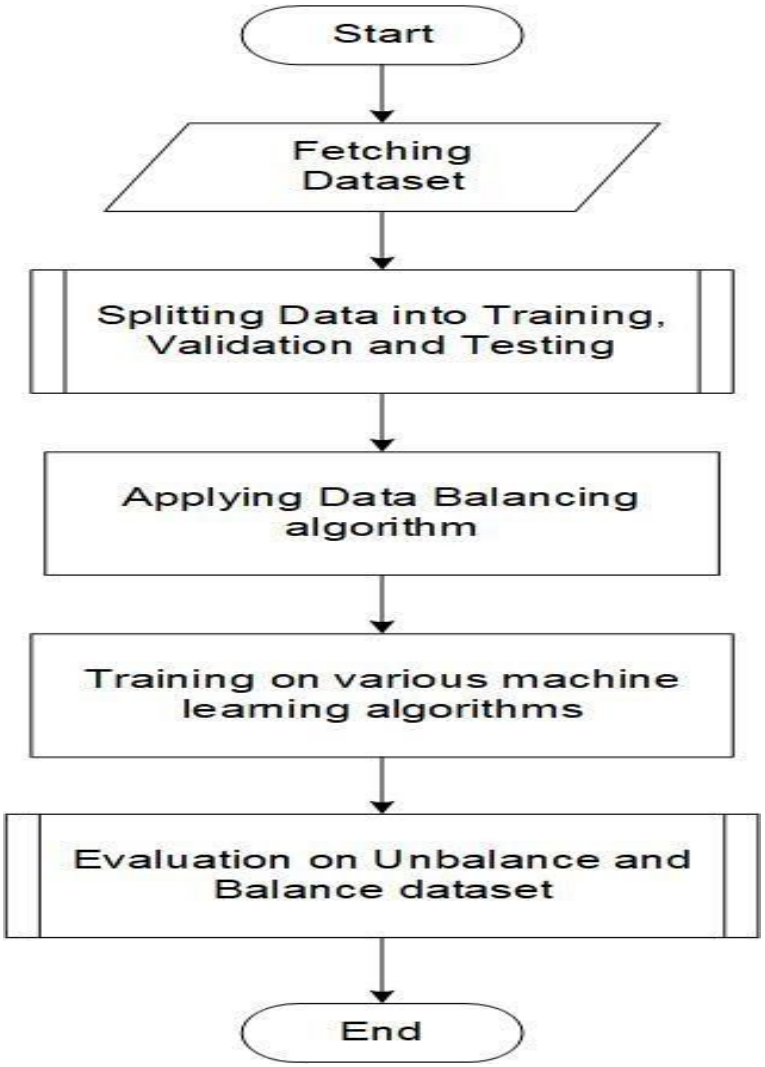


Figure 3.2: System Design

The System can be divided into five steps. It starts by fetching data, in this report data is fetched from kaggle's credit card fraud detection dataset. Then data is split between training and testing i.e, 80 percent is used for training and 20 percent is used for testing. In third step data balancing algorithms like smote, adasyn and allknn are applied on training dataset, which is then used to train various machine learning algorithms and finally accuracy, precision, recall, f1score, mcc and roc are used as metrics to evaluate machine learning algorithms.

3.4 System Architecture

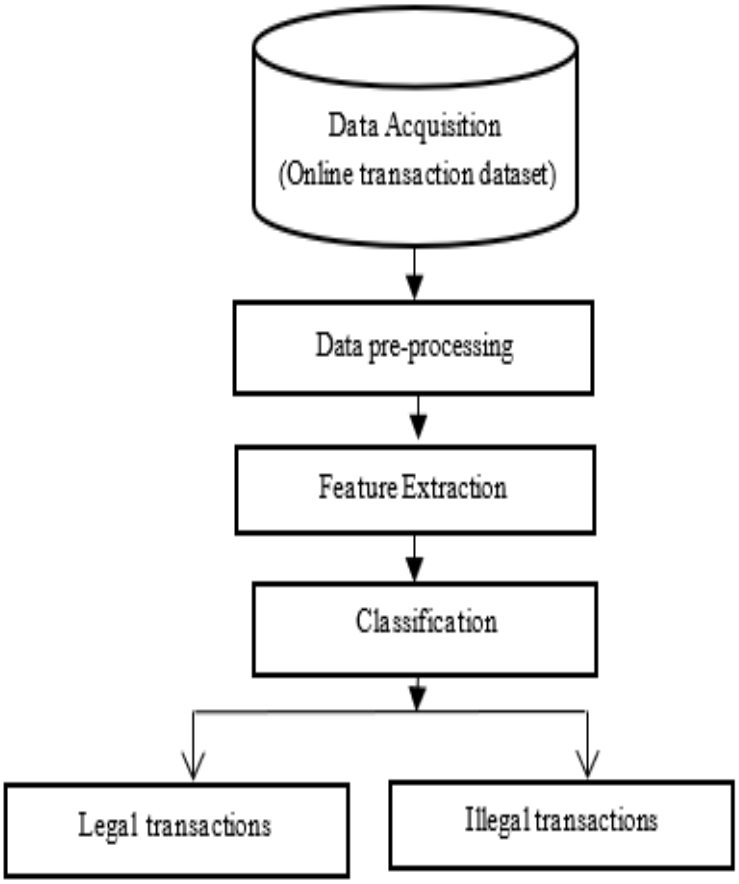


Figure 3.3: System Design

System architecture describes how the data flows into the model. Data preprocessing involves steps to convert raw data into useful data. Then an algorithm is selected which is used to extract features from the data on the basis of which classification of a given transaction is made as legal and illegal.

CHAPTER 4

IMPLEMENTATION RESULTS

Below are the result of observation. These results are based on dataset with 20% data used for testing and random state as 16 [22].

Table 4.1: Results for test cases without using any data balancing algorithm

Algorithms	Accuracy	Precision	Recall	F1score	Mcc	Roc
Logistic Regression	0.99899	0.70833	0.70103	0.70466	0.70417	0.85026
Log Reg with minmax	0.00170	0.00170	1.0	0.00339	0	0.5
Decision Tree	0.99913	0.70689	0.84536	0.76995	0.77261	0.92238
Random Forest	0.99961	0.92134	0.84536	0.88172	0.88234	0.92261
Adaboost	0.99933	0.82417	0.77319	0.79787	0.79794	0.88645
Catboost	0.99956	0.9	0.83505	0.86631	0.86670	0.91744
Xgboost	0.99961	0.93103	0.83505	0.88043	0.88154	0.91747
Lightgbm	0.99590	0.21900	0.54639	0.31268	0.34426	0.77153

The above table lists down the results of various machine learning algorithm without using any data balancing algorithm. It can be seen that boosting algorithms generally performed better than bagging in terms of accuracy. Random forest and xgboost are found to perform better in terms of precision but a sudden drop of performance can be seen in lightgbm and logistic regression with minmaxscaler. Almost same behaviour can be observed for f1score. Xgboost and random forest were found to perform best in terms of mcc and logistic regression worst.

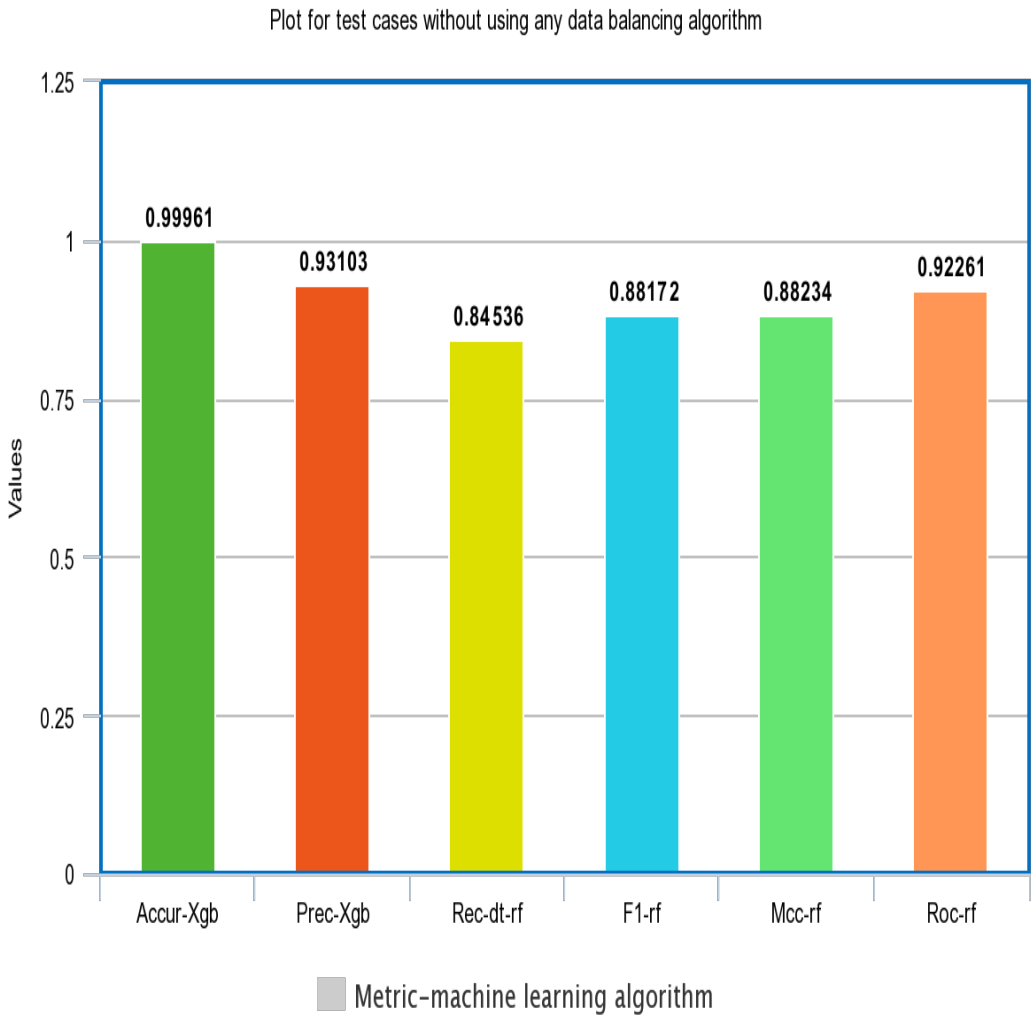


Figure 4.1: Performance of algorithms without any data balancing algorithms

Table 4.2: Results for test cases using smote as data balancing algorithm

Algorithms	Accuracy	Precision	Recall	F1score	Mcc	Roc
Log Regression	0.97738	0.06816	0.96907	0.12737	0.25390	0.97323
Log Reg with minmax	0.98476	0	0	0	-0.00483	0.49322
Decision Tree	0.99817	0.48022	0.87628	0.62043	0.64796	0.93733
Random Forest	0.99959	0.92134	0.87628	0.88172	0.88234	0.92261
Adaboost	0.98586	0.10313	0.94845	0.18604	0.31030	0.96719
Catboost	0.99943	0.79279	0.90721	0.84615	0.84780	0.91744
Xgboost	0.99961	0.93103	0.83505	0.88043	0.88154	0.91747
Lightgbm	0.99590	0.21900	0.54639	0.31268	0.34426	0.77153

The above table lists down the results of various machine learning algorithm using smote as data balancing algorithm. Performance degraded the most in logistic regression with minmaxscaler. Random forest and xgboost were found to perform

better in terms of accuracy, precision, fscore and mcc. Logistic regression registered highest recall and roc value.

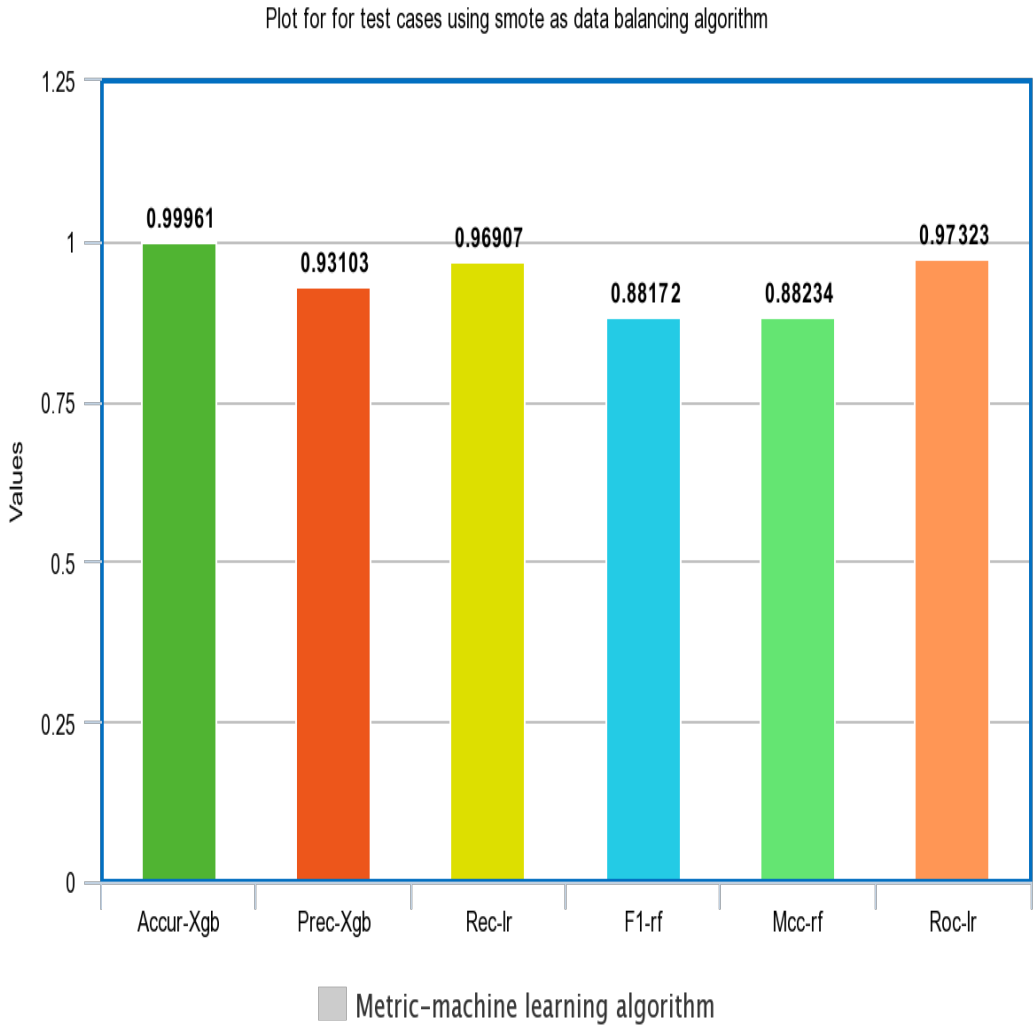


Figure 4.2: Performance of algorithms with smote as data balancing algorithm

Table 4.3: Results for test cases using adasyn as data balancing algorithm

Algorithms	Accuracy	Precision	Recall	F1score	Mcc	Roc
Logistic Regression	0.96550	0.04439	0.93814	0.08476	0.20004	0.95184
Log Reg with minmax	0.98681	0	0	0	-0.00445	0.49424
Decision Tree	0.99782	0.43005	0.85567	0.57241	0.60575	0.92686
Random Forest	0.99957	0.87628	0.87628	0.87628	0.87607	0.93803
Adaboost	0.98546	0.10054	0.94845	0.18181	0.30630	0.96699
Catboost	0.99949	0.81481	0.90721	0.85853	0.85952	0.95343
Xgboost	0.99957	0.86138	0.89690	0.87878	0.87875	0.94833
Lightgbm	0.99884	0.60992	0.88659	0.72268	0.73484	0.94281

The above table lists down the results of various machine learning algorithm using adasyn as data balancing algorithm. Like in smote here also logistic regression with minmaxscaler performed worst. Almost similar trend was observed for accuracy, precision, f1score and mcc where random forest and xgboost performed the best and adaboost performed dominated precision and roc metrics.

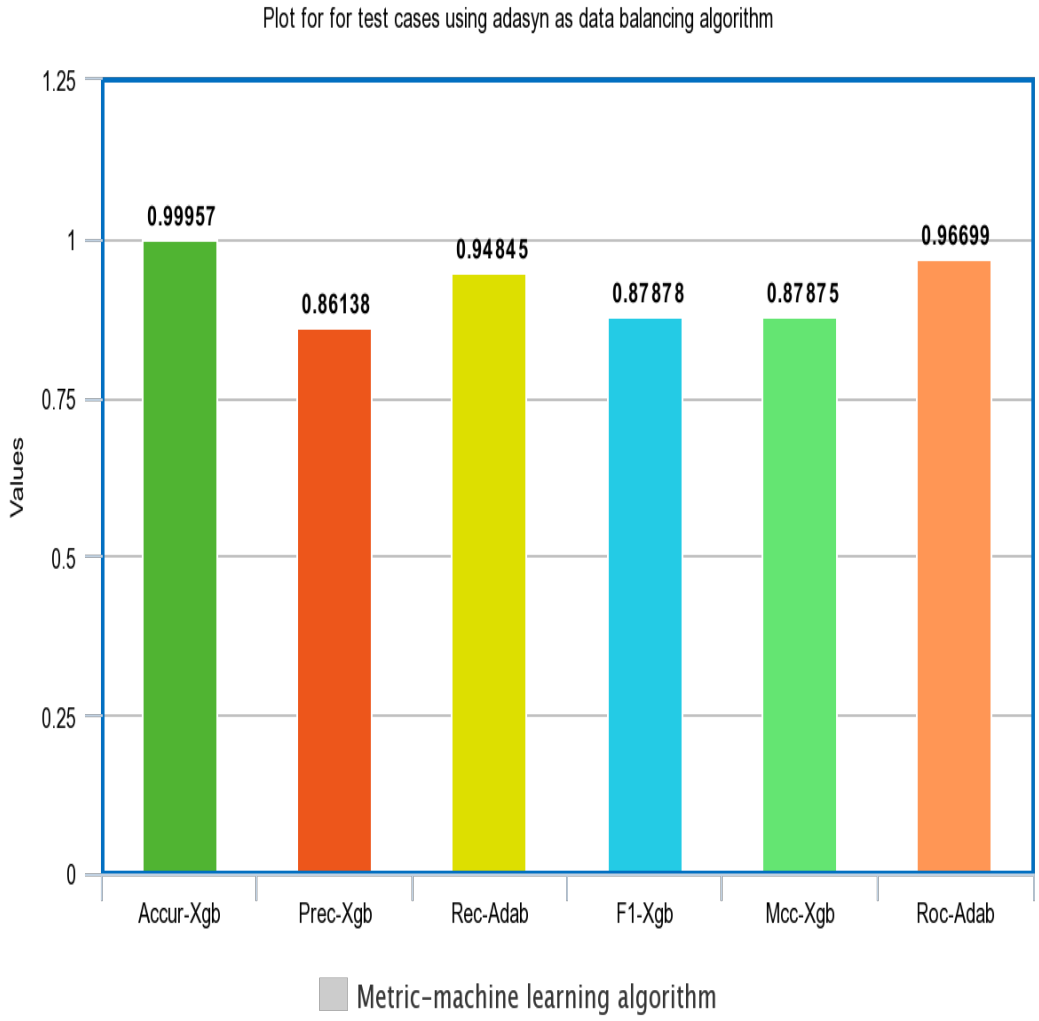


Figure 4.3: Performance of algorithms with adasyn as data balancing algorithm

The below table lists down the results of various machine learning algorithm using allknn as data balancing algorithm. It can be clearly seen that logistic regression with minmaxscaler performed worst in all metrics except recall where a perfect score of 1 is observed. Random forest and xgboost continued the trend and showed good result in terms of accuracy, precision, recall, f1score and mcc. Also decision tree outperformed all algorithms in terms of roc score.

Table 4.4: Results for test cases using allknn as data balancing algorithm

Algorithms	Accuracy	Precision	Recall	F1score	Mcc	Roc
Logistic Regression	0.99901	0.71578	0.70103	0.70833	0.70787	0.85027
Log Reg with minmax	0.00170	0.00170	1	0.00339	0	0.5
Decision Tree	0.99913	0.69354	0.88659	0.77828	0.78374	0.942964
Random Forest	0.99959	0.92045	0.83505	0.87567	0.876515	0.91746
Adaboost	0.99919	0.75757	0.77319	0.76530	0.76494	0.88638
Catboost	0.99957	0.91011	0.83505	0.87096	0.87156	0.91745
Xgboost	0.99963	0.92222	0.85567	0.88770	0.88814	0.92777
Lightgbm	0.99626	0.26033	0.64948	0.37168	0.40972	0.82316

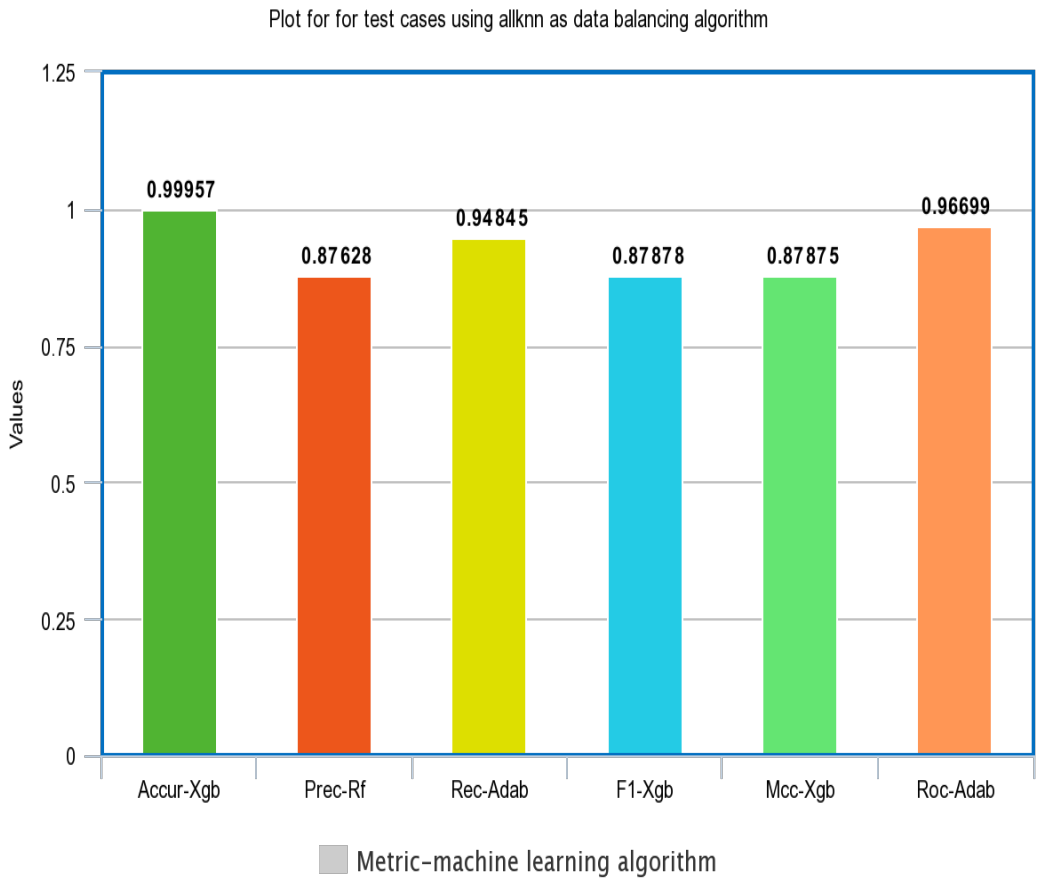


Figure 4.4: Performance of algorithms with allknn as data balancing algorithm

On analysis of behaviour of logistic regression with minmaxscaler it was observed that true positives for smote and adasyn is 0 and as a result precision, recall, and f1score is 0. True positives is 0 due to which the value of true positive * true negative vanishes to 0 and hence negative value of mcc is observed for smote and adasyn. Similarly true negative and false negative is 0 due to which mcc is 0 for simple and allknn.

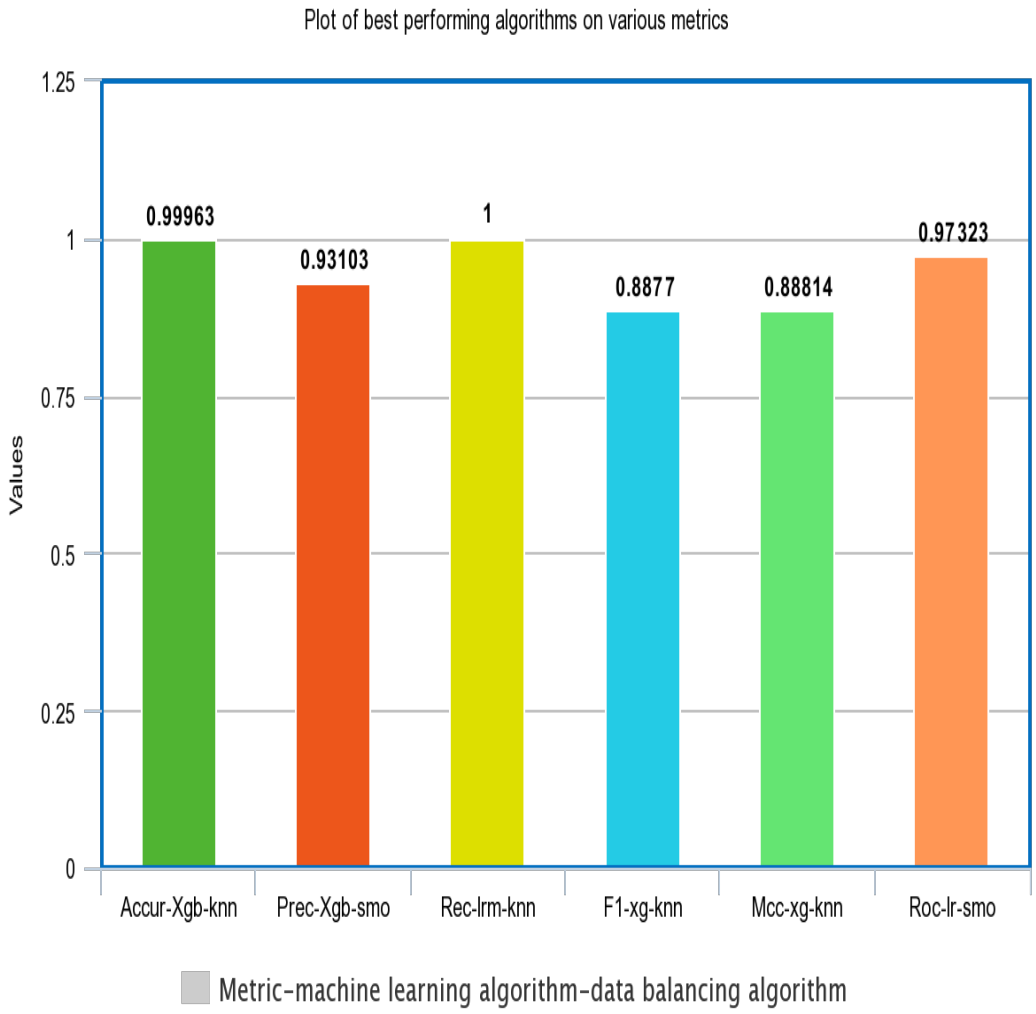


Figure 4.5: Performance of algorithms on given metrics

CHAPTER 5

CONCLUSION

The overall project discusses the problem of credit card fraud detection and how machine learning can be used to deal with it. As stated in methodology various machine learning algorithms are compared on the basis on accuracy, precision, recall, f1score, roc and mcc.
During the study while comparing various metrics for various machine learning it was found that highest accuracy was observed for xgboost using allknn, highest recall was observed for logistic regression with minmaxscaler [23] using allknn, highest precision and f1-score was observed for xgboost using allknn, highest roc and mcc score was observed for logistic regression using smote. Some results were quite an anomaly when compared to others like overall minmaxscaler doesn't improve the model performance instead it showed down tread when used with logistic regression in terms of smote and adasyn it shown negative mcc score, value of zero for precision, recall, f1score when used with smote. On further observation true positive was found to be zero for logistic regression with smote and adasyn, true negative and false negative for logistic regression with minmaxscaler and logistic regression with minmaxscaler along with allknn is found to be 0.
During analysis 'V14' is found to be the most dominating feature when smote and adasyn are used as data balancing algorithm, also both of them are oversampling method, 'V17' is found to be the most dominating feature when no data balancing algorithm is used. Most variation was observed in case of allknn as data balancing algorithm.

5.1 Future Work

- Performance improvement can be made by hyperparameter tuning techniques like grid searchCV [24], randomized searchCV, bayesian optimization [25].
- Study on comparison between machine learning techniques, neural network, hid-

den markov model [26] and bayesian belief network [27] could be done to built a better combined model.

- More data balancing techniques can be included like borderline-smote [28] and other hybrid approaches to study the effect on metrics.

REFERENCES

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

[2] S. Panigrahi, A. Kundu, S. Sural, and A. K. Majumdar, "Credit card fraud detection: A fusion approach using dempster–shafer theory and bayesian learning," *Information Fusion*, vol. 10, no. 4, pp. 354–363, 2009.

[3] P. K. Chan, W. Fan, A. L. Prodromidis, and S. J. Stolfo, "Distributed data mining in credit card fraud detection," *IEEE Intelligent Systems and Their Applications*, vol. 14, no. 6, pp. 67–74, 1999.

[4] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*. John Wiley & Sons, 2013, vol. 398.

[5] M. A. Friedl and C. E. Brodley, "Decision tree classification of land cover from remotely sensed data," *Remote sensing of environment*, vol. 61, no. 3, pp. 399–409, 1997.

[6] L. E. Raileanu and K. Stoffel, "Theoretical comparison between the gini index and information gain criteria," *Annals of Mathematics and Artificial Intelligence*, vol. 41, no. 1, pp. 77–93, 2004.

[7] A. Liaw, M. Wiener *et al.*, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[9] R. E. Schapire, "Explaining adaboost," in *Empirical inference*. Springer, 2013, pp. 37–52.

[10] T. Chen, T. He, M. Benesty, V. Khotilovich, and Y. Tang, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, pp. 1–4, 2015.

[11] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," in *Advances in neural information processing systems*, 2018, pp. 6638–6648.

[12] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in neural information processing systems*, 2017, pp. 3146–3154.

[13] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. IEEE, 2008, pp. 1322–1328.

[14] H. Dubey and V. Pudi, "Class based weighted k-nearest neighbor over imbalance dataset," in *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 2013, pp. 305–316.

[15] P. K. Chan, W. Fan, A. L. Prodromidis, and S. J. Stolfo, "Distributed data mining in credit card fraud detection," *IEEE Intelligent Systems and their Applications*, vol. 14, no. 6, pp. 67–74, 1999.

[16] A. Srivastava, A. Kundu, S. Sural, and A. Majumdar, "Credit card fraud detection using hidden markov model," *IEEE Transactions on Dependable and Secure Computing*, vol. 5, no. 1, pp. 37–48, 2008.

[17] R. Brause, T. Langsdorf, and M. Hepp, "Neural data mining for credit card fraud detection," in *Proceedings 11th International Conference on Tools with Artificial Intelligence*, 1999, pp. 103–106.

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

[19] S. Ghosh and D. L. Reilly, "Credit card fraud detection with a neural-network," in *System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on*, vol. 3. IEEE, 1994, pp. 621–630.

[20] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal classifier for imbalanced data using matthews correlation coefficient metric," *PloS one*, vol. 12, no. 6, p. e0177678, 2017.

[21] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC genomics*, vol. 21, no. 1, p. 6, 2020.

[22] A. Srivastava, A. Kundu, S. Sural, and A. Majumdar, "Credit card fraud detection using hidden markov model," *IEEE Transactions on dependable and secure computing*, vol. 5, no. 1, pp. 37–48, 2008.

[23] H. Shaheen, S. Agarwal, and P. Ranjan, "Minmaxscaler binary pso for feature selection," in *First International Conference on Sustainable Technologies for Computational Intelligence*. Springer, 2020, pp. 705–716.

[24] G. Ranjan, A. K. Verma, and S. Radhika, "K-nearest neighbors and grid search cv based real time fault monitoring system for industries," in *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*. IEEE, 2019, pp. 1–5.

[25] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, 2012, pp. 2951–2959.

[26] M. J. Beal, Z. Ghahramani, and C. E. Rasmussen, "The infinite hidden markov model," in *Advances in neural information processing systems*, 2002, pp. 577–584.

[27] J. Cheng and R. Greiner, "Learning bayesian belief network classifiers: Algorithms and system," in *Conference of the Canadian Society for Computational Studies of Intelligence*. Springer, 2001, pp. 141–151.

[28] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," in *International conference on intelligent computing*. Springer, 2005, pp. 878–887.