JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

ENHANCING CREDIT CARD FRAUD DETECTION THROUGH MACHINE LEARNING

[1]P.T.S. PRIYA,[2]Siriyala Mohan Krishna,
[1] Assistant professor, [2] MCA 2nd year,
[1]Computer Science and Engineering, [2]Master of Computer Applications,
Sanketika Vidya Parishad Engineering College, Visakhapatnam, India

ABSTRACT
Detection of credit card fraud is currently the issue that arises most frequently in the modern world. Due to the growth of e-commerce platforms as well as online transactions, this has happened. Fraudulent usage of a credit card typically occurs when the card is taken for any unauthorised use, or even when the fraudster utilises the card's information for his own objectives. Currently, there are many credit card issues that we must deal with. A technique for detecting credit card fraud [3] was devised in order to catch fraudulent actions. Machine learning [4] algorithms are the primary focus of this project. Both the Random Forest [2] and Adaboost [1] algorithms are employed. On accuracy, precision, recall, and F1-score, the two algorithms' outputs are compared. On the basis of the confusion matrix [5], the ROC curve [6] is plotted. The algorithms from Random Forest and Adaboost are compared, and the method with the highest accuracy, precision, recall, and F1-score are regarded as the optimal approach for use in fraud detection.

Keywords: Adaboost, ROC Curve, fraudulent, credit card fraud, Random Forest

I.INTRODUCTION

Credit card fraud has emerged as a growing concern across various sectors, including government offices, corporate industries, and financial institutions. This surge in fraudulent [7] activities can be attributed to the increasing reliance on the internet for transactions. However, it is important to note that credit card fraud is not limited to online transactions alone, as online and offline [8] card transactions also experience a significant rise in fraudulent incidents. Despite the utilization of data mining [10] techniques to combat this issue, the accuracy of detecting credit card frauds remains a challenge. To mitigate these losses effectively, the implementation of efficient algorithms for fraud detection has become crucial. By employing advanced algorithms, organizations can significantly reduce credit card fraud incidents and minimize associated losses. When someone else uses your credit card without your consent in your place, it is referred to as credit card fraud. Without stealing the original physical card, fraudsters can carry out any unauthorised activities by stealing the PIN or account information from the credit card. We could determine whether the new transactions are genuine or fraudulent using credit card fraud detection.

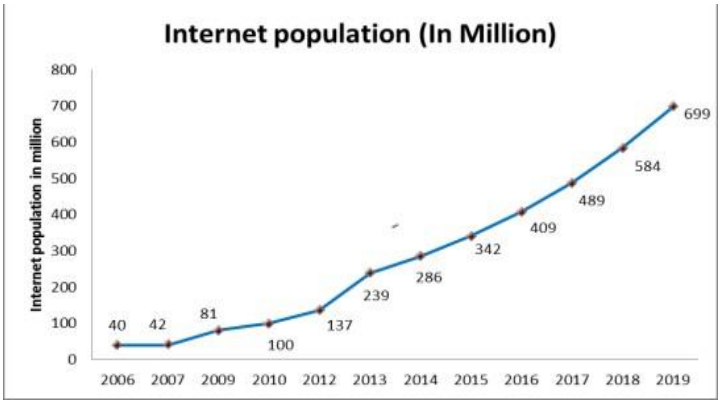


Figure1.Growth of Internet users

II.SYSTEM ANALYSIS

A lot of research is being done on new strategies for detecting credit card fraud, and neutral networks [9], data mining, and distributed data mining are of particular interest. To stop this type of credit card fraud, numerous alternative methods are employed. After conducting a literature review on the subject, we can draw the conclusion that there are many additional approaches in machine learning itself for detecting credit card fraud. The card, such as a credit card or debit card, may be used in the fraud that is performed. In this instance, the card itself serves as a source of fraud in the transaction. To gain products without paying money or to obtain an unauthorised sum may be the motivation behind the crime. Fraudsters often choose to target credit cards. The rationale is that a lot of money can be made quickly without incurring many risks, and even a crime may not be discovered for several weeks.

III.EXISTING SYSTEM

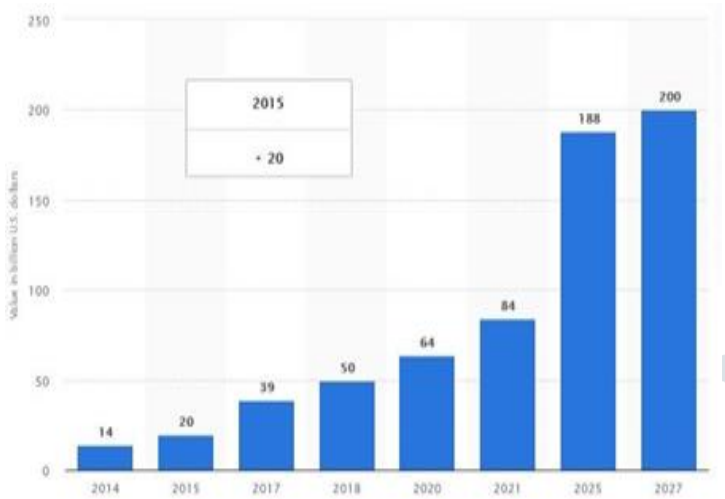


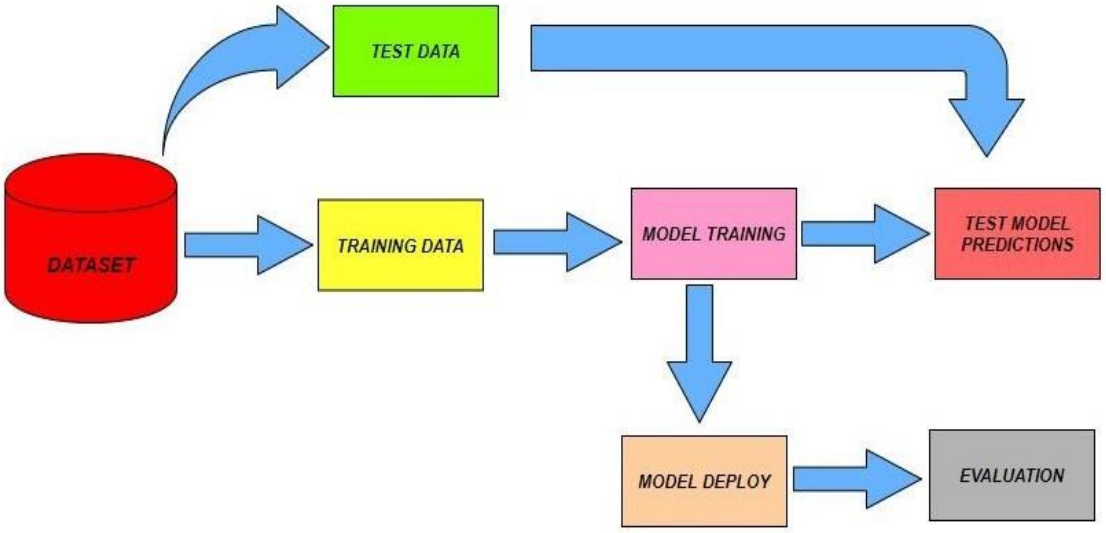
Figure.2 Growth of E-Commerce sites

Both machine learning and deep learning [11] techniques are used in the research on credit card fraud detection. We expand on the work done at two separate stages in this section: The strategies that are available to handle the imbalanced data, as well as (i) the procedures that are easily accessible for fraud detection A deep learning's unbalanced data can be handled in a variety of ways. They are (a) classification techniques [12], (b) sampling techniques, and (c) techniques-like procedures. Support vector machines (SVM), decision trees [14], logistic regression [13], gradient boosting, K-nearest neighbour, etc. are a few of the machine learning techniques used for detecting credit fraud. Using decision trees, random forests, SVM, and logistic regression, Navanushu Khare and Saad Yunus Sait presented their work in 2018. They used a dataset that was significantly skewed to operate on this kind of dataset. Accuracy, sensitivity, specificity, and precision are the main criteria for performance evaluation. According to the results, the accuracy of the Logistic Regression is 97.7%, that of the Decision Trees is 95.5%, that of the Random Forest is 98.6%, and that of the SVM classifier is 97.5%. They have come to the conclusion that the Random Forest algorithm is the best algorithm for spotting fraud because it has the highest accuracy among the other algorithms. Additionally, they came to the conclusion that the SVM algorithm has a data imbalance issue and does not perform any better in terms of identifying credit card fraud.

IV.PROPOSED SYSTEM

This paper's primary goal is to use algorithms like the Random Forest and Adaboost algorithms to categorise the transactions in the dataset that have both fraud and non-fraud transactions [15]. The algorithm that best detects credit card fraud transactions is then chosen by comparing the performance of the two algorithms. The dividing of the data, model training [16], model deployment, and the evaluation criteria are all part of the process flow [20] for the credit fraud detection problem shown in Figure 3.

Figure.3 Process Flow



The comprehensive architecture [17] diagram for the credit card fraud detection system [Figure 4] outlines a number of phases, from data collection to model deployment and analysis, depending on outcomes. The Kaggle credit card fraud dataset is used in this model, and the dataset needs to go through some pre-processing. We must now separate the data into training and testing sets in order to create the model. The Random Forest and the Adaboost models are created using the training set of data. Next, we create both models. Finally, the models' accuracy, precision, recall, and F1-score are computed. Finally, a more realistic comparison of credit card fraud transactions

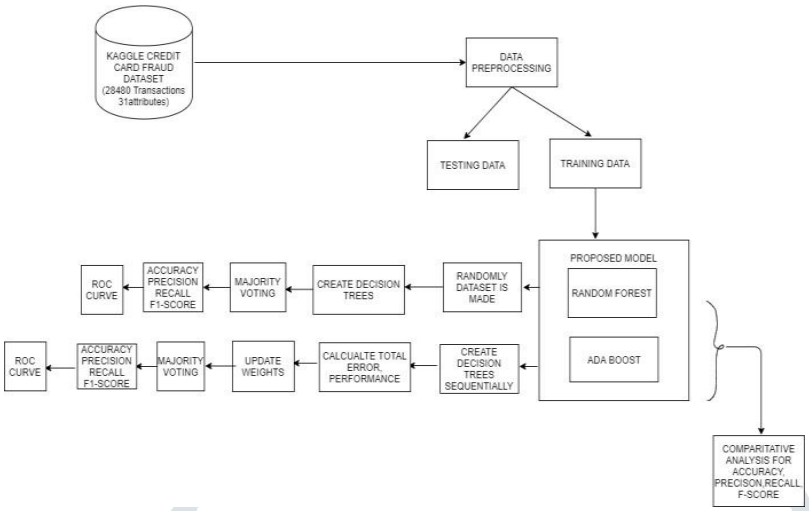


Figure.4 Architecture Diagram

V.SPECIFICATION

A. HARDWARE REQUIREMENTS (Minimum Requirement)

1.PROCESSOR: i3 5th Gen 2.2 Ghz
2. RAM: 4GB+RAM

B. SOFTWARE REQUIREMENTS

1. Code Editors: PyCharm, Data Science with Anaconda
2. Frameworks and Dependencies: Tensor flow, Keras, Open CV
3. Operating System: Windows 10
4. Domain: Python
5. Version: Python IDLE (3.8.0)

C. CODE EDITORS

PyCharm

An integrated development environment (IDE) for the Python programming language is called PyCharm. It was created by Jet Brains, a former IntelliJ subsidiary based in the Czech Republic. It supports both web development with Django and data science with Anaconda and offers code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and more.

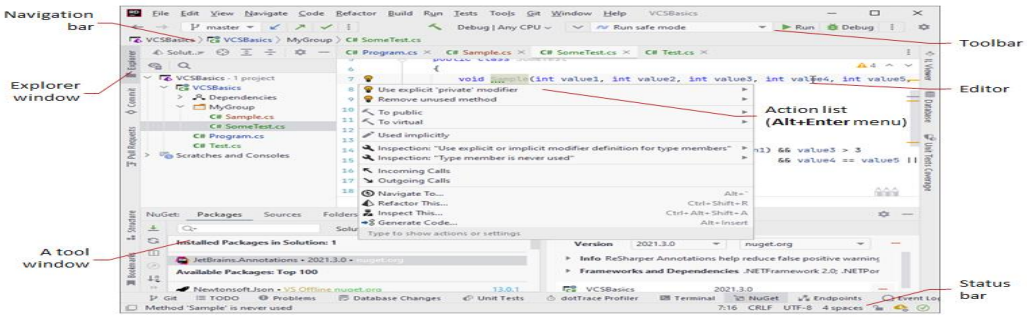


Figure5: PyCharm screen

Code conclusion, syntax and error highlighting, linter integration, and rapid fixes are all part of the coding aid and analysis services. Project and code navigation features include file structure views, specialised project views, and rapid switching between files, classes, methods, and usages. Refactoring in Python comprises renaming, extracting methods, introducing variables and constants, as well as other operations. Python has a built-in debugger. Line-by-line code coverage and integrated unit testing development in Python for Google App Engine. Integration of version control tools: change lists and merge for Mercurial, Git, Subversion, Perforce, and CVS in a single user interface support for scientific software packages like SciPy, NumPy, and Matplotlib (only the professional edition).

VI. A. Random Forest Algorithm

One of the most popular supervised learning algorithms is Random Forest [Figure 6]. Regression and classification techniques can both be applied here. However, classification issues are where this approach is most commonly applied. A forest is often composed of trees, and the Random Forest technique similarly builds decision trees on the sample data and extracts predictions from each of the sample data. In that case, the Random Forest algorithm is an ensemble approach. Due to the fact that it averages the results, this algorithm is superior to single decision trees in that it lessens overfitting.

Steps for Random Forest Algorithm

Choose some sample data at random from the trained Kaggle credit card fraud dataset. The Decision Trees that are used to categorise the cases into fraud and non-fraud cases are now formed using the sample data that was generated at random. The Decision Trees are created by splitting the nodes; the root node is the node with the most information gained, and it categorises fraud situations from non-fraud instances. After the majority vote, the decision trees may produce a value of 0, indicating that these are not fraud situations. In the end, we determine the F1 score, recall, accuracy, and precision for both fraud and non-fraud cases.

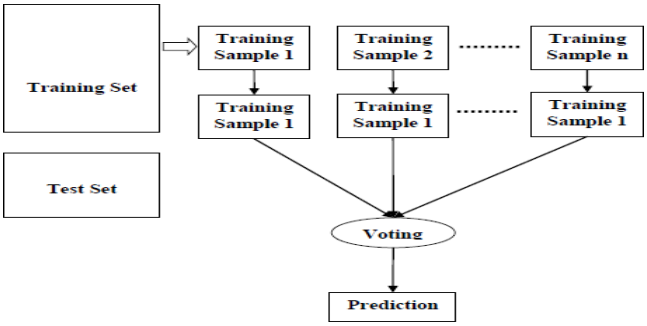
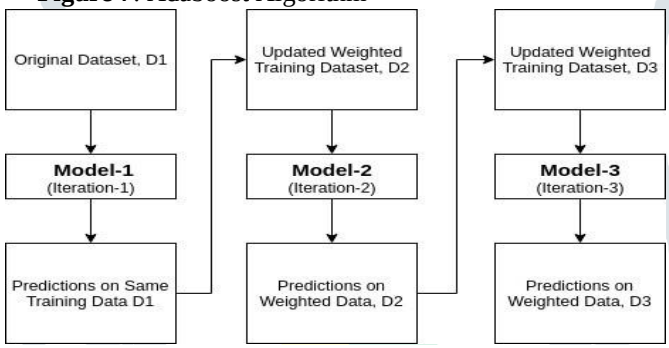


Figure.6 Random Forest Algorithm

B. Adboost Algorithm

One of the ensemble strategies is boostering. Strong classifiers are created using this algorithm from weaker ones. This can be achieved by employing a weak model in the series to develop a strong model. Initially, training data are used to build a model. Then, the first model is used to generate the second model by fixing the flaws that it included. This is a repeated process that is carried out until either the maximum number of models are added or the entire training dataset is properly predicted. One of the best boosting algorithms created for binary classification was called Adboost.

Figure 7: Adaboost Algorithm



Adaptive boosting is the abbreviation for Adaboost. It works best with reluctant students. Using the Adaboost boosting method [Figure 7], a strong classifier is created by combining several weak classifiers. Short decision trees and the Adaboost algorithm can be used together. The Adaboost is produced in such a way that the nodes are first made and the tree is made, and then the effectiveness of the tree in each of the cases is examined. Additionally, a weight is given. The training data with the highest predictive difficulty is the one that is given greater weight. The Adaboost method is an effective classifier that handles both simple and complex situations well. The fact that this method is primarily sensitive to noisy data is a drawback.

Steps for creating adboost algorithm

The credit card fraud dataset from Kaggle is used to train. Pick a few samples of the data at random. Sequentially develop decision trees for identifying fraud and non-fraud instances using the sample data generated at random. The decision trees are initially created. This can be achieved by classifying fraud and non-fraud situations and separating the nodes based on which has the largest information gain. Next, determine the error rate, performance, and updated weights for the transactions that were mistakenly labelled as fraud or non-fraud. After performing a majority vote, the decision trees may produce an output that identifies instances of non-fraud. The decision trees may produce 1, which signifies that the case involves fraud. and lastly, we determine the F1-score, recall, accuracy, and precision for both fraud and non-fraud situations.

EVALUATION AND RESULT ANALYSIS

Dataset

The information for credit card fraud statistics was obtained from a European credit card provider. The Kaggle website is where the dataset was found. The dataset contains the credit cardholders' purchases made in September 2013 (as of this writing). The transactions that were completed over a two-day period are included in the dataset. 284,807 transactions total are in the data set, 492 of which are fraudulent. Only 0.172% of all transactions are fraudulent. By using the PCA transformation, the dataset that contains the input variable is transformed into numerical values. Concealing information is achieved by doing this. It is impossible to PCA-transform the features "Time" and "Amount." The difference in seconds between a certain transaction and the first transaction is represented by the class "Time."

Evaluation Standards

We must analyse parameters like precision, recall, F1-score, and accuracy in order to compare different methods. There is also a graphic of the confusion matrix. A 2*2 matrix makes up the confusion matrix. There are four outputs in the matrix: TPR, TNR, FPR, and FNR. From the confusion matrix, metrics like sensitivity, specificity, accuracy, and error rate can be calculated. Then, we will use the finest method to identify credit card fraud.

The confusion matrix's output is: The true positive rate, which is the proportion of fraudulent transactions that the system really identifies as such. The actual negative Rate, which is the proportion of valid transactions that the system even recognises as such. False Positive Rate, which is the percentage of legal transactions that are incorrectly labelled as fraud. Transactions that are fraudulent but are mistakenly categorised as legal are known as false negative rates. By graphing the TPR against the FPR, the Receiver Operating Characteristics curve is produced. At different thresholds, this is possible. The FPR and TPR are the horizontal and vertical axes, respectively, of a graph called a ROC curve. The AUC is shown as a graph underneath the ROC curve.

Results Analysis

For both techniques, the ROC curve and the confusion matrix are presented. The dataset produces various results when used with various methods. The outcomes of applying the dataset for the random forest model are as follows:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	93825
1	0.95	0.77	0.85	162
accuracy			1.00	93987
macro avg	0.97	0.89	0.93	93987
weighted avg	1.00	1.00	1.00	93987

Figure.8 Output for Random Forest

The evaluation criteria are described in Figure 8. For non-fraud cases, precision, recall, and F1-score are the same, whereas for fraud cases, they are different.

```
Confusion Matrix on train data
[[190490      0]
 [     0    330]]

Confusion Matrix on test data
[[93818    37]
 [    7   125]]
```

Figure.9 Confusion Matrix for Random Forest

According to the confusion matrix [Figure 9], there are 190490 true positives and 0 false positives for the train data, whereas there are 330 false negatives. In the test data, there were 93818 true positives, 37 false positives, 7 true negatives, and 125 false negatives.

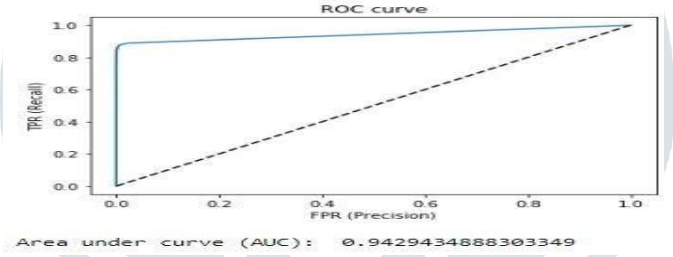


Figure.10 ROC curve for Random Forest

The dataset is now being used using the Adaboost algorithm. Similar to the Random Forest Algorithm, the results are obtained.

Figure.11 Output for Adaboost

```
Accuracy = 0.9990743400683073
              precision    recall  f1-score   support

Confusion Matrix on train data                  93825
[[190464    120]                                  162
 [    26    210]]
Confusion Matrix on test data
[[93811    65]
 [   14    97]]
```

According to the evaluation criteria [Figure 11], there are significant differences between the fraud cases and the non-fraud cases in terms of precision, recall, and F1-score.

Figure.12 Adaboost's Confusion Matrix

According to the confusion matrix [Figure 12], there are 190464 true positives and 120 false positives for the train data, while there are only 26 true negatives and 201 false negatives. In the test data, there were 93811 true positives, 65 false positives, 14 true negatives, and 97 false negatives.

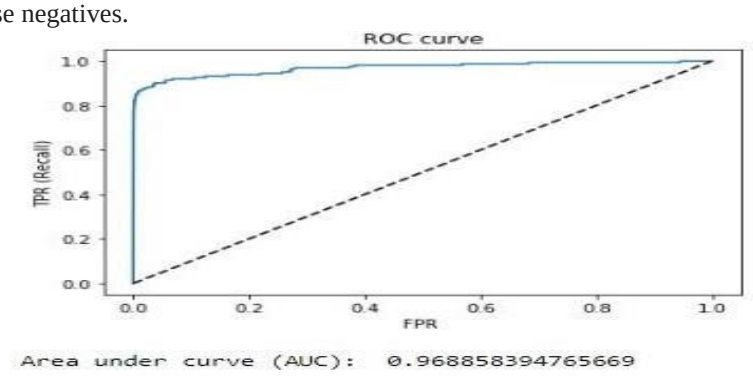


Figure.13 ROC curve for Adaboost

Now, Figure 13 ROC curve for Adaboost [18] compares the random forest with the adaboost algorithms. Although the accuracy of the two methods is the same, they have different precision, recall, and F1 score [19]. The most accurate, most reliable, and highest F1-score algorithms are random forest ones.

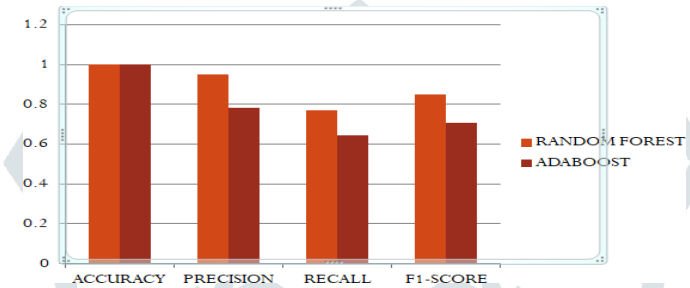


Figure.14 Comparision of Algorithms

CONCLUSION

We cannot claim that this particular algorithm completely detects fraud, despite the fact that there are various fraud detection algorithms. Our investigation leads us to the conclusion that both the Random Forest and the Adaboost algorithms have the same accuracy. When precision, recall, and the F1-score are taken into account, the Random Forest algorithm outperforms the Adaboost algorithm. Thus, we draw the conclusion that, when used to detect credit card fraud, the Random Forest method performs better than the Adaboost method

FUTURE SCOPE

It is evident from the data above that a variety of machine learning techniques are applied to identify fraud; however, we can see that the outcomes are unsatisfactory. Therefore, in order to accurately detect credit card fraud, we would like to employ deep learning techniques.

REFERENCES

[1] An article reference of Adaboost
https://www.sciencedirect.com/science/article/abs/pii/S1474034622000362
[2] An article reference of Random Forest algorithm
https://journals.sagepub.com/doi/pdf/10.1177/1536867X20909688
[3] An article reference of Credit card fraud
https://link.springer.com/article/10.1007/s44230-022-00004-0
[4] A web reference of Machine learning
https://pubs.acs.org/doi/full/10.1021/jacs.0c09105
[5] An article reference of Confusion matrix
https://iopscience.iop.org/article/10.1088/1742-6596/1229/1/012055/meta
[6] An article reference of ROC curve
https://academic.oup.com/ije/article/49/4/1397/5714095
[7] An article reference of Fraudulent
https://iopscience.iop.org/article/10.1088/1742-6596/1601/5/052016/meta
[8] An article reference of Online and Offline
https://link.springer.com/article/10.1007/s10899-022-10106-w
[9] An article reference of Neutral networks
https://link.springer.com/article/10.1007/s10915-022-01939-z
[10] An article reference of Data mining
https://pubs.aip.org/aip/acp/article-abstract/2400/1/020006/2821439/A-comprehensive-survey-of-fraud-detection-methods?redirectedFrom=fulltext
[11] A web reference of Deep learning
https://www.mdpi.com/2079-9292/11/5/756
[12] An article reference of Classification techniques
https://iopscience.iop.org/article/10.1088/1742-6596/2161/1/012072/meta
[13] A web reference of Logistic regression

https://www.hindawi.com/journals/ijd/2022/5358602/
[14] An article reference of decision trees
https://www.sciencedirect.com/science/article/pii/S2772662222000261
[15] An article reference of fraud and non-fraudulent transactions.
https://www.sciencedirect.com/science/article/pii/S2772662223000036
[16] A web reference of model training.
http://proceedings.mlr.press/v139/killamsetty21a.html
[17] An article reference of random forest architecture.
https://iopscience.iop.org/article/10.1088/1742-6596/1755/1/012039/meta
[18] An article reference of ROC Curve for adaboost
https://www.sciencedirect.com/science/article/abs/pii/S0020025521002875
[19] An article reference of F1 score
https://link.springer.com/article/10.1186/s12864-019-6413-7
[20] an article reference of process flow.
https://link.springer.com/article/10.1007/s12599-013-0250-z

XII.BIBLOGRAPHY

P.T.S. Priya working as an Assistant professor in the Department of Computer Science and Engineering, Sanketika Vidya Parishad Engineering College, Visakhapatnam Andhra Pradesh. with 6 years of experience in Master of Computer Applications (MCA), Accredited by NAAC. with her area of interests in C, Computer Organization, Software Engineering, IOT.

Siriyala Mohan Krishna is studying his 2nd year, Master of Computer Applications in Sanketika Vidya Parishad Engineering College, affiliated to Andhra University, accredited by NAAC. With his interest in python, web development and as a part of academic project, Enhancing Credit Card Fraud Detection Through Machine Learning. In the study on credit card fraud detection, machine learning and deep learning [11] approaches are both utilised. In this section, he went into greater detail on the two stages of the work: The methods available to deal with imbalanced data, as well as (i) the simple methods readily available for fraud detection A deep learning's unbalanced data can be handled in a variety of ways. They are (a) classification techniques [12], (b) sampling techniques, and (c) techniques-like procedures. As a result of a desire to comprehend. In completion of his MCA.