# Predicting the factors related to credit card fraud detection using Ensemble approach and Adaptive Synthetic Sampling

Sri HarshaVardhan Palla
School of Computer Science and Engineering
Vellore Institute of Technology
Chennai , India
Sriharshavardhan.palla2020@vitstudent.ac.in

Rukksana A
School of Computer Science and Engineering
Vellore Institute of Technology
Chennai , India
rukksana.a2020a@vitstudent.ac.in

Sacchin Ganesh Sundar
School of computer science
Vellore institute of technology
Chennai, India
sacchinsundar.g2020@vitstudent.ac.in

Shreya Shri Ragi
School of Computer Science and Engineering
Vellore Institute of Technology
Chennai, India
shreyashri.ragi2020@vitstudent.ac.in

***Abstract* — Credit card fraud detection is the technique of detecting fraudulent purchase attempts and rejecting them instead of performing the order. There are numerous methods and instruments for spotting fraud, and most businesses use a combination of several of them. Losses from credit card fraud amount to billions of dollars each year, making it a significant problem for the banking sector. Financial organisations must identify fraudulent transactions in order to stop and lessen losses. Traditional techniques of identifying fraud utilising rule-based systems have become less effective as transaction volume and complexity have increased. Payment cards are simple to use since they require a small number of transmissions to the bank to identify your account and approve the transaction. Simple numbers that must be shared with the persons you are transacting with make it very difficult to implement strict data security practises. Hence various algorithms are used in credit card fraud detection to examine transaction data, spot patterns and abnormalities, and flag questionable behaviour for more research. The effectiveness of fraud detection models has also been enhanced using machine learning techniques like random forests and support vector machine . These methods are still being developed and improved upon, and the financial sector will continue to prioritise their study**.

***Keywords***— credit card, fraud, financial sector, random forest

## I. Introduction

Credit card fraud is a major concern for both consumers and financial organisations. Online transactions are on the rise, which has raised the possibility of fraudulent activity. Essentials of Data Analysis can help identify relevant features that can be used to train machine learning models for fraud detection.The generation of predictive models based on processed data is made possible by Machine Learning algorithms and Essentials of Data Analysis, which work together to comprehend the data and identify trends. These techniques have been widely employed to detect fraudulent activity in real-time in order to address this issue.

In order to detect credit card fraud, it is necessary to examine a number of variables, including the user's transaction history, location, and purchasing habits. Algorithms for machine learning can be trained to evaluate enormous volumes of data and spot anomalies that might point to fraud. For the purpose of detecting credit card fraud, a variety of machine learning techniques can be utilised, including supervised learning techniques like logistic regression, decision trees, and neural networks, as well as unsupervised learning techniques such such as clustering and anomaly detection based on the data . We are using Voting classifier which can be a powerful tool for credit card fraud detection, allowing for more accurate and robust predictions that can better detect and prevent fraudulent transactions. Our aim of this project is to explore the data set and get to know about the key attributes , to visualize the dataset using appropriate graphs, feature selection and outlier detection and to implement machine learning algorithms to detect more accurate credit card fraud detection.

In conclusion, credit card fraud detection using machine learning algorithms is an effective way to identify and prevent fraudulent transactions. By building predictive models based on historical transaction data, financial institutions can detect and prevent fraud in real-time, reducing losses and improving customer satisfaction.

## II. Literature review

The work by the authors [1] states that the paper addresses the challenge of imbalanced classification in fraud detection, where the number of fraudulent transactions is much lower than the number of legitimate transactions. The proposed method, called Q-Credit Card Fraud Detector, uses a deep Q-learning algorithm to learn the optimal policy for fraud detection. The algorithm learns from a dataset of

historical credit card transactions and updates its policy based on the rewards received for correctly classifying fraudulent transactions. To address the issue of imbalanced classification, the Q-Credit Card Fraud Detector employs a two-stage classification approach. In the first stage, the algorithm identifies a subset of transactions that are more likely to be fraudulent using a decision threshold. In the second stage, the algorithm applies a more strict threshold to the subset of transactions identified in the first stage to improve the accuracy of fraud detection. The paper presents experimental results that show the effectiveness of the proposed method in detecting credit card fraud. The Q-Credit Card Fraud Detector outperforms several state-of-the-art machine learning models in terms of accuracy, precision, and recall. Overall, the Q-Credit Card Fraud Detector provides a promising solution to the imbalanced classification problem in credit card fraud detection, and its reinforcement learning approach can adapt to changes in fraud patterns over time.

The work by the authors [2] states that the paper addresses the challenge of classifying fraudulent transactions accurately while avoiding false positives that could inconvenience legitimate customers. The proposed method, called GAN-based credit card fraud detector, uses a GAN to generate synthetic fraudulent transactions that are similar to real fraudulent transactions. The synthetic transactions are used to train a classifier to distinguish between fraudulent and legitimate transactions. The classifier is then applied to real transaction data to detect fraud.

To ensure the synthetic transactions are representative of real fraudulent transactions, the GAN is trained using a dual loss function. The discriminator loss function ensures that the synthetic transactions are classified as fraudulent, while the generator loss function ensures that the synthetic transactions are similar to real fraudulent transactions. The paper presents experimental results that show the effectiveness of the proposed method in detecting credit card fraud. The GAN-based credit card fraud detector outperforms several state-of-the-art machine learning models in terms of accuracy, precision, and recall while maintaining a low false positive rate.

Overall, the GAN-based credit card fraud detector provides a promising solution to detecting credit card fraud while reducing the risk of inconveniencing legitimate customers with false positives. Its use of GANs to generate synthetic transactions provides a novel approach to addressing the class imbalance problem in credit card fraud detection.

The work by the authors [3] states that the proposed method uses a sparse auto encoder to compress high-dimensional transactional data into a lower-dimensional representation. The lower-dimensional representation is then used to train a GAN to generate synthetic fraudulent transactions that are similar to real fraudulent transactions. The synthetic transactions are then used to augment the training data for a classification algorithm that detects fraud.

To ensure that the synthetic transactions are representative of real fraudulent transactions, the GAN is trained using a dual loss function. The discriminator loss function ensures that the synthetic transactions are classified as fraudulent, while the generator loss function ensures that the synthetic transactions are similar to real fraudulent transactions. The paper presents experimental results that show the effectiveness of the proposed method in detecting credit card fraud. The method outperforms several state-of-the-art machine learning models in terms of accuracy, precision, and recall while maintaining a low false positive rate.

Overall, the proposed method provides a promising solution to detecting credit card fraud while reducing the risk of inconveniencing legitimate customers with false positives. Its use of a sparse auto encoder and GAN to generate synthetic transactions provides a novel approach to addressing the class imbalance problem in credit card fraud detection.

The work by the authors [4] states that the paper addresses the challenge of detecting fraud accurately while avoiding false positives that could inconvenience legitimate customers. The proposed method uses a deep neural network, specifically a multilayer perceptron (MLP), to classify transactions as either fraudulent or legitimate. The MLP is trained using a dataset of historical credit card transactions, where each transaction is represented by a set of features such as the amount, location, and time of the transaction. To address the issue of imbalanced classification, where the number of fraudulent transactions is much lower than the number of legitimate transactions, the paper employs a data augmentation technique. The technique generates synthetic fraudulent transactions by adding noise to existing fraudulent transactions, thereby increasing the number of fraudulent transactions in the training data.

The paper presents experimental results that show the effectiveness of the proposed method in detecting credit card fraud. The deep learning-based approach outperforms several state-of-the-art machine learning models in terms of accuracy, precision, and recall while maintaining a low false positive rate. Overall, the proposed method provides a promising solution to detecting credit card fraud while reducing the risk of inconveniencing legitimate customers with false positives. Its use of deep learning and data augmentation techniques provides a novel approach to addressing the class imbalance problem in credit card fraud detection.

The work by the authors [5] states that the paper addresses the challenge of detecting fraud accurately while avoiding false positives that could inconvenience legitimate customers. The proposed method uses a feedforward ANN with a single hidden layer to classify transactions as either fraudulent or legitimate. The ANN is trained using a dataset of historical credit card transactions, where each transaction is represented by a set of features such as the amount,

location, and time of the transaction. To address the issue of imbalanced classification, where the number of fraudulent transactions is much lower than the number of legitimate transactions, the paper employs a cost-sensitive learning approach. The approach assigns different misclassification costs to fraudulent and legitimate transactions, with the cost for misclassifying a fraudulent transaction being much higher than the cost for misclassifying a legitimate transaction.

The paper presents experimental results that show the effectiveness of the proposed method in detecting credit card fraud. The ANN-based approach outperforms several state-of-the-art machine learning models in terms of accuracy, precision, and recall while maintaining a low false positive rate. Overall, the proposed method provides a promising solution to detecting credit card fraud while reducing the risk of inconveniencing legitimate customers with false positives. Its use of an ANN and cost-sensitive learning approach provides a novel approach to addressing the class imbalance problem in credit card fraud detection.

The work by the authors [6] states that the research paper examines the effectiveness of different fraud detection techniques in card-not-present (CNP) transactions, which are transactions made without the physical presence of the card. The paper evaluates several machine learning-based approaches, including logistic regression, random forest, and support vector machines, for detecting CNP fraud. The approaches are evaluated using a dataset of historical CNP transactions from a large online retailer.

The paper also investigates the impact of different features on the performance of the fraud detection approaches. The features include transaction amount, country of the transaction, and device used for the transaction. The experimental results show that logistic regression and random forest approaches outperform support vector machines in detecting CNP fraud. Additionally, the transaction amount and country of the transaction are found to be the most important features for detecting CNP fraud. The paper also evaluates the cost-effectiveness of the different fraud detection approaches by considering the cost of false positives and false negatives. The results show that a combination of logistic regression and random forest approaches provides the best cost-effectiveness, with a low false positive rate and a high true positive rate. Overall, the paper provides insights into the effectiveness of different fraud detection techniques in CNP transactions and highlights the importance of considering the cost- effectiveness of these techniques. The findings can guide investment decisions for organizations looking to improve their fraud detection capabilities in CNP transactions.

The work by the authors [7] states that Class imbalance is a major issue in credit card fraud detection, as the number of fraudulent transactions is often much smaller than the number of non-fraudulent transactions. Traditional machine learning algorithms tend to perform poorly in such imbalanced datasets, as they tend to focus on the majority class, resulting in high false negative rates. The proposed framework addresses this issue by using clustering techniques to group similar transactions together, and then using a similarity-based selection method to select a representative subset of non-fraudulent transactions to balance the dataset. The authors evaluate the performance of the proposed framework on a publicly available credit card fraud dataset, comparing it to several state-of-the-art approaches. The results show that the proposed framework outperforms the other approaches, achieving high accuracy, precision, recall, and F1-score values. The framework is also shown to be effective in detecting previously unseen types of fraud, demonstrating its ability to generalize to new data.

The paper provides a detailed description of the framework, including the clustering and SBS algorithms used, and provides a thorough evaluation of its performance. The proposed framework has the potential to improve the accuracy of credit card fraud detection systems, and could be applied in other domains with imbalanced datasets. Overall, the paper provides a valuable contribution to the field of credit card fraud detection and machine learning.

The work by the authors [10] states that the paper proposes a DRL-based framework that can learn from past transactional data and detect fraudulent transactions in real-time. The proposed framework uses a neural network to map the transactional data to a lower dimensional space and uses this representation as input to the reinforcement learning agent. The agent learns to make decisions on whether a transaction is fraudulent or not, based on the reward signal received for correct decisions. The reward signal is calculated based on the losses incurred due to fraudulent transactions. The paper provides experimental results to demonstrate the effectiveness of the proposed framework in detecting payment fraud. The results show that the DRL-based approach outperforms traditional rule-based methods and other machine learning models in terms of accuracy, precision, and recall. Overall, the paper presents an innovative approach to detecting payment fraud using deep reinforcement learning, which has the potential to improve the accuracy and efficiency of fraud detection systems.

The work by the authors [8] states that this paper uses machine learning techniques to analyse payment card transaction data in real-time. The system is designed to identify fraudulent transactions and alert financial institutions, merchants, and cardholders to potential fraud. The authors describe the architecture of the system, which consists of several modules, including data pre-processing, feature extraction, fraud detection, and alert generation. The system uses a combination of unsupervised and supervised machine learning algorithms, including clustering, principal component analysis, and decision trees, to analyse transaction data and identify patterns that are indicative of fraud. The paper presents a detailed evaluation of the system's performance, using a large dataset of payment card transactions. The results show that the system is able to achieve high accuracy, precision, recall, and F1-score values, and is effective in detecting various types of fraud,

including account takeover, card-not-present fraud, and identity theft.

Overall, the paper provides a valuable contribution to the field of payment card fraud detection, demonstrating the potential of machine learning techniques to improve the accuracy and efficiency of fraud detection systems. The proposed system has the potential to help financial institutions and merchants reduce their losses due to payment card fraud, and to protect cardholders from financial harm. The paper also highlights the importance of continuous monitoring and adaptation of fraud detection systems, in order to keep up with evolving fraud patterns and tactics.

The work by the authors [9] states that The authors compare the performance of various algorithms, including decision trees, random forests, support vector machines, artificial neural networks, and convolutional neural networks, on a publicly available credit card fraud dataset. The paper provides a detailed description of the algorithms used, including their advantages and disadvantages, and presents a thorough evaluation of their performance. The results show that deep learning algorithms, particularly convolutional neural networks, outperform traditional machine learning algorithms in terms of accuracy, precision, recall, and F1-score values. The authors also evaluate the effectiveness of feature selection techniques, including principal component analysis and correlation-based feature selection, in improving the performance of the algorithms. They find that feature selection can significantly reduce the dimensionality of the dataset and improve the accuracy of the algorithms.

The paper highlights the importance of using a balanced dataset for credit card fraud detection, as class imbalance can result in high false negative rates. The authors address this issue by using a combination of oversampling and under sampling techniques to balance the dataset.

Overall, the paper provides a valuable contribution to the field of credit card fraud detection, demonstrating the potential of state-of-the-art machine learning and deep learning algorithms to improve the accuracy and efficiency of fraud detection systems. The findings of the study can be used to guide the development of more effective fraud detection systems and can help financial institutions and merchants reduce their losses due to payment card fraud .

## III. Summary of Literature Survey

Credit card fraud is a significant issue for financial institutions and individuals. Traditional methods of fraud detection are often time-consuming and not as effective as they could be. In recent years, machine learning (ML) has emerged as a promising solution for credit card fraud detection. These literature reviews provide an overview of the current research on ML in credit card fraud detection.

Many studies have shown that ML techniques are effective in detecting credit card fraud. Some of the most commonly used ML algorithms in this field include decision trees, neural networks, support vector machines, and logistic regression. Researchers have also explored the use of ensemble methods, such as random forests and boosting, to improve the accuracy of fraud detection.

One of the challenges in credit card fraud detection is dealing with imbalanced datasets, where the number of fraudulent transactions is much smaller than the number of legitimate transactions. Several studies have proposed techniques to handle imbalanced datasets, such as oversampling the minority class, undersampling the majority class, and using cost-sensitive learning.

Another area of research is the use of deep learning techniques, such as convolutional neural networks and recurrent neural networks, for credit card fraud detection. These techniques have shown promising results and can potentially improve the accuracy of fraud detection.

## IV. Proposed Methodology

### A. Proposed Methodology Diagram

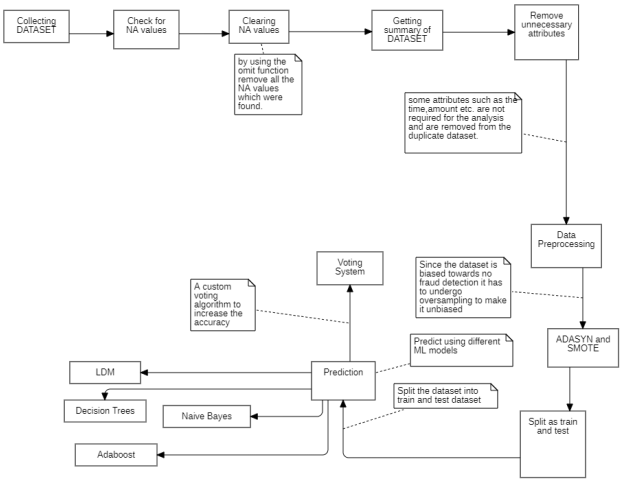


Fig 4.1 Proposed methodology

### B. Need for Proposed Methodology

The reason for doing oversampling is to remove the biasness of the dataset. The dataset we are using has 99% of non fraud transactions so part of probability the model generates that a transaction is not fraud even though its actually fraud. We are using voting system to increase the accuracy of the model.

### B. Novelty

To increase the accuracy of the dataset we have used ADASYN preprocessing with ADABOOST prediction algorithm to get an accuracy of 99%, we are also using our own custom voting algorithm to improve accuracy of the

model which gave us an accuracy of 97% while using Decision Tree, Linear Discriminant Model and Naïve Bayes.

### C. Dataset

The dataset we have used is the credit card fraud detection dataset taken from Kaggle, the dataset contains the transactions that occurred in 2 days in which nearly 0.002% of the transactions are detected as credit card frauds. The dataset contains only numeric values which are the result of PCA(principle component analysis) transformation, the original features of the dataset are hidden because of confidentiality issues. The features of the dataset include Time, Amount and features V1,V2,V3…..V28 which are obtained from PCA transformation, and finally Class which is the target variable of our analysis and class has 2 unique values 1 and 0, it takes value 1 if there is a fraud and 0 if there isn't a fraud.

### D. Algorithms

Algorithms used for Data Preprocessing

ADASYN

ADASYN (Adaptive Synthetic Sampling) is a machine learning algorithm used for imbalanced classification problems. It is an extension of the Synthetic Minority Over-sampling Technique (SMOTE) algorithm, which generates synthetic examples of the minority class to balance the class distribution.

ADASYN is designed to adaptively generate synthetic examples in regions of the feature space where the density of the minority class is low, which can improve the classification performance. It generates synthetic examples by perturbing the feature vectors of the minority class examples that are already present in the dataset.

ADASYN works by first determining the degree of imbalance in the dataset, then generating synthetic examples in the feature space based on the degree of imbalance. Specifically, it generates more synthetic examples for minority class examples that are harder to learn, and fewer synthetic examples for minority class examples that are easier to learn.

ADASYN is density-based because it generates synthetic samples in areas of the feature space where the density of the minority class is low. This is because the main goal of ADASYN is to address the issue of class imbalance, where the minority class is underrepresented and the classifier may not perform well due to lack of data to learn from.

By focusing on the areas of low density, ADASYN generates synthetic samples that are more representative of the minority class and can improve the classifier's performance. This approach is effective because it generates synthetic samples in the areas where the minority class is harder to learn, and this can increase the diversity of the minority class samples in the dataset.

The density-based approach in ADASYN is also adaptive, as it generates more synthetic samples in regions with lower densities, ensuring that the synthetic samples are generated in a way that is proportional to the local density of the minority class. This means that ADASYN can adapt to different distributions of data and is not limited to specific datasets or class distributions.

These are the parameters for the algorithm:

1) n_neighbors: This parameter specifies the number of nearest neighbors to use when generating synthetic samples. The default value is 5, but it can be adjusted based on the specific dataset.

2) sampling_strategy: This parameter specifies the desired ratio of the minority class after sampling. By default, it is set to "auto", which means that the minority class will be oversampled to have the same number of samples as the majority class. However, it can be set to a float value between 0 and 1, representing the desired ratio of minority class samples to majority class samples.

3) random_state: This parameter sets the seed for the random number generator used by the algorithm. This ensures that the same results are obtained each time the algorithm is run.

4) n_jobs: This parameter specifies the number of parallel jobs to run when fitting and transforming the data. This can speed up the process for larger datasets.

5) ratio: This parameter specifies the ratio of synthetic samples to generate for each minority class sample. The default value is 1.0, meaning that one synthetic sample is generated for each minority class sample. However, it can be adjusted based on the specific dataset and the degree of imbalance.

In density-based model with decay, the density threshold is adjusted based on a decay function that takes into account the distance between points. The density threshold decreases as the distance between points increases, making it more difficult for points that are far away from each other to be considered part of the same method.

Overall, ADASYN is a useful tool for addressing imbalanced classification problems, especially when the imbalance is severe and traditional methods such as under sampling or oversampling are not effective.

SMOTE

It creates synthetic examples of the minority class by interpolating between existing minority class samples. A new synthetic sample is then created by randomly selecting one of the k-nearest neighbors and using it to create a new sample by interpolating between the two samples. SMOTE can be applied to any classification problem where there is a class imbalance, and it has been shown to improve the performance of many classifiers. However, it can also lead to overfitting if not used carefully, and it may not work well in some cases where the minority class is highly complex or has a large variability.

Algorithms used for Prediction

DECISION TREES

Decision tree is a popular machine learning algorithm which generates a tree like structure. The tree consists of decision nodes and leaf nodes, where the decision nodes represent the features, and the leaf nodes represent the class labels or target values. The algorithm recursively partitions the data based on the features' values and selects the best feature to split the data at each node. Decision trees are easy to interpret and visualize, making them useful for exploratory data analysis and decision-making in various applications such as customer segmentation, fraud detection, and medical diagnosis. However, decision trees can be prone to overfitting, and their performance may suffer when dealing with high-dimensional data or noisy data.

NAÏVE BAYES

It is based on the Bayes' theorem, which describes the probability of an event occurring given some prior knowledge. In the context of Naive Bayes, the algorithm calculates the probability of a new data point belonging to a particular class based on the probability distribution of its features, assuming that these features are independent of each other. Despite its simplicity and the "naive" assumption of feature independence, Naive Bayes has been shown to be very effective in a wide range of classification tasks, including text classification, spam filtering, and image recognition. Moreover, it is computationally efficient and can handle high-dimensional data with ease.

ADABOOST

Adaboost, short for Adaptive Boosting, is a machine learning algorithm used for classification and regression problems. It is an ensemble method that combines multiple "weak" classifiers to form a "strong" classifier. Adaboost assigns weights to the training samples, which are then used to train each weak classifier sequentially. The final strong classifier is a weighted combination of all the weak classifiers, where the weights are determined by the accuracy of each weak classifier. Adaboost is widely used in various applications, such as face detection, object recognition, and text classification, due to its high accuracy and robustness.

LDM

The Linear Discriminant Model (LDM) is a statistical technique used for classification and prediction problems. It involves finding a linear combination of predictor variables that maximally separate two or more groups based on their class labels. This is achieved by computing the differences in means between groups and their within-group variances. LDM is particularly useful when dealing with high-dimensional datasets where the number of predictors is large relative to the sample size. It is also often used as a preprocessing step in machine learning algorithms to reduce the dimensionality of the data and improve classification performance. LDM has been widely applied in various fields, including finance, medicine, and image recognition.

CUSTOM VOTING ALGORITHM

We have taken the predictions made by using naïve bayes, LDM and decision tree algorithms to implement a voting system. This voting system will give whether a particular transaction is fraud or not using the predictions of the above 3 datasets. It generates the result based on the majority vote if more number of models say that a particular transaction is fraud the voting model will take the transaction to be fraud or vice versa.

E.Parameter setting

ADASYN:

| | |
|---|---|
| n_neighbours | 5 |
| sampling_stratergy | auto |
| random_state | 120 |
| n_jobs | 0 |
| ratio | 1 |
| λ | 0.2 |

Table 4.1

Spliting:

| | |
|---|---|
| P | 0.7 |
| times | 1 |
| List | F |

Table 4.2

Adaboost:

| | |
|---|---|
| boos | TRUE |
| mfinal | 10 |

Table 4.3

## V. Experimental Results and Discussion

Below are the results that we got on implementing the above said algorithms on credit card fraud dataset. The dataset is biased towards the non fraud class so in order to balance the dataset we have used ADASYN(adaptive synthetic sampling) as the oversampling method to make the dataset unbiased and legit for the predictions.

- We have also used SMOTE to check whether there is a significant difference between the two oversampling methods.
- The dataset before using any of the data preprocessing techniques:

| 0 | 1 |
|---|---|
| 0.998272514 | 0.001727486 |

Table 5.1

After using ADASYN:

| 0 | 1 |
|---|---|
| 0.5000149 | 0.4999851 |

Table 5.2

After using SMOTE:

| 0 | 1 |
|---|---|
| 0.5003793 | 0.4996207 |

Table 5.3

- As u can see from the above results that the difference between ADASYN and SMOTE techniques of oversampling is very minimal in a big dataset like this. Since both ADASYN and SMOTE are offering almost similar type of balancing and ADASYN is an improved version of SMOTE cause it decreasing the chance of overfitting we have continued our experimentation with the dataset we have generated using ADASYN algorithm.

Decision Tree:-

- We have implemented decision tree algorithm on the preprocessed datasets. The below are the decision tree obtained by using ADASYN as the preprocessing algorithm:-

- a decision tree node that has N total samples, and the node is split into K classes (or categories) represented by k = 1, 2, ..., K. The number of samples in each class is denoted as N_k, where k = 1, 2, ..., K.

- The Gini index (Gini impurity) of the node is given by the formula:

$$Gini(node) = 1 - \sum(N_k / N)^2$$

Equation:-5.1

where $\sum(N_k / N)^2$ is the sum of squared proportions of samples in each class.

- From the decision tree we can come to a conclusion that v14 v17 and v4 play a important role in determining whether a transaction is fraud or not. GIni index is used to find the decision tree.

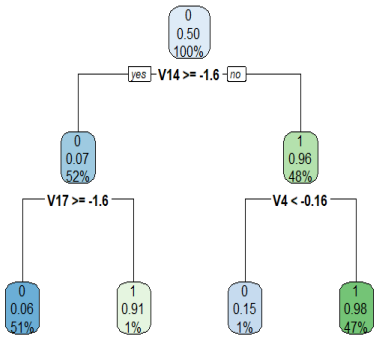


Fig 5.1 Decision Tree implemented

Naïve Bayes:-

- We have implemented naïve bayes using the e1701 package in R. The equation of naïve bayes is,

$$P(y|x) = P(x|y) * P(y) / P(x)$$

Equation:-5.2

Where:
P(y|x) is the posterior probability of the class (y) given the input variables (x).
P(x|y) is the likelihood of the input variables (x) given the class (y).
P(y) is the prior probability of the class (y).
P(x) is the marginal probability of the input variables (x).

LDM:-

- We have implemented the linear discriminant model using the mass package.

$$p(Ci|xi) = p(xi|Ci)\ p(Ci) / p(xi)$$

Equation:-5.3

Where:
p(Ci|xi) is the posterior probability of class Ci given the input feature vector xi.
p(xi|Ci) is the class-conditional probability of xi given class Ci.
p(Ci) is the prior probability of class Ci.
p(xi) is the marginal probability of xi.
The predicted class for an input feature vector is the class with the highest posterior probability.


Fig 5.2 Linear Discriminant Analysis prediction

Voting Model:-

- We have then taken Decision Tree, Naïve Bayes and LDM algorithms to implement a voting system.
- We have taken the predictions done by each model and have chosen the most common prediction of the three as the prediction of our model.
- Therefore if 2 or more than 2 of the algorithms say that for a given transaction is not fraud then the prediction of our model for that particular transaction will be not fraud.
- If 2 or more than 2 of the algorithms say that for a given transaction is fraud then the prediction of our model will also be fraud.

Accuracy of each model:-

| | |
|---|---|
| Decision Tree | 95% |
| Naïve Bayes | 92% |
| LinearDiscriminant Model | 93% |
| Voting System | 96% |

Table 5.4

- As we can see using the voting system has increased the accuracy of the model.

Adaboost:-

- We tried using adaptive boosting with adaptive synthetic sampling, this combination of algorithm has taken a lot of time to compile and predict but it has given as an accuracy of nearly 99%.

| | |
|---|---|
| Accuracy | 98% |
| Kappa | 0.9779 |

Table 5.5

## VI. Statistical Results and Discussion

We have taken the frequency distribution graphs to check which variables are similar to each other.

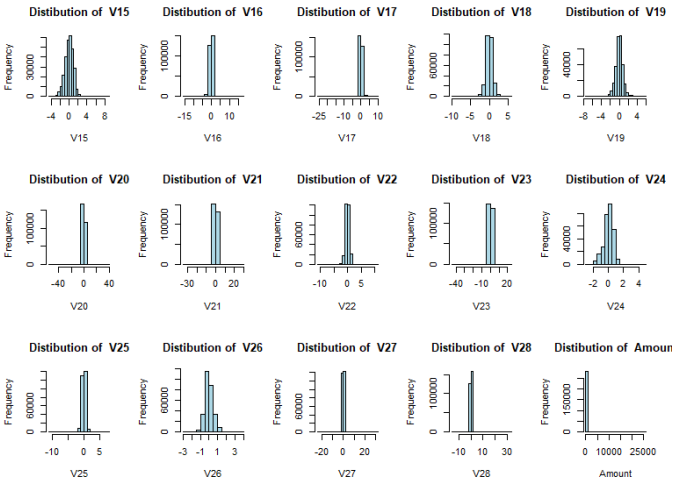


Fig 6.1

As we can see from the histograms that no 2 particular variables are similar too each other.

We have checked the dataset for class variable to see number of yes and no values in the dataset, we have taken pie chart for checking the number of yes and no values.

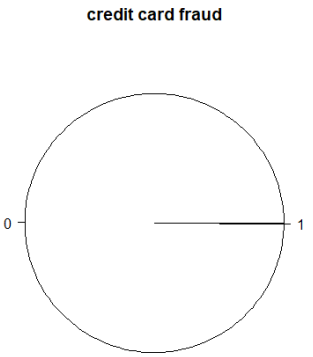


Fig 6.2 Pie chart before ADASYN

As we can see from the above pie chart the dataset consists of mostly 0 i.e, no fraud and very minimal fraud transactions. To avoid this biasedness we have done preprocessing using ADASYN the piechart after obtained after completing the preprocessing is

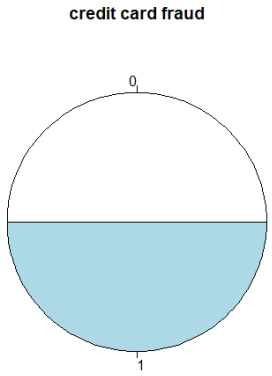


Fig 6.3 Pie chart after using ADASYN

As you can see from the pie chart the biasedness is almost removed making the dataset into a nearly equal number of fraud and non fraud transactions.

To see which variable is more correlated to the target variable we have taken the correlation plot for the dataset.

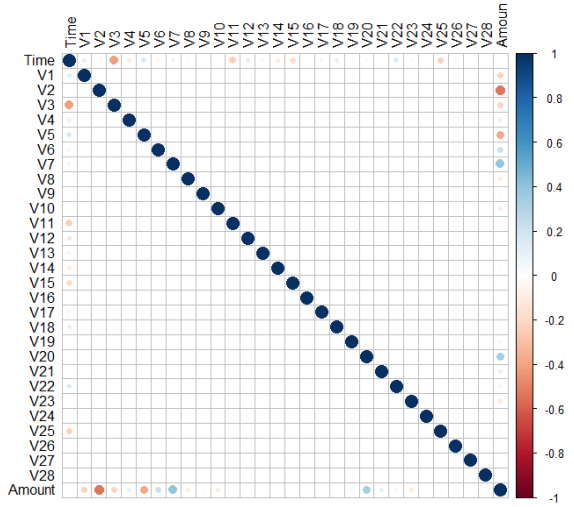


Fig 6.4 Correlation Plot

As we can see from the correlation plot there is a strong correlation between V2 and Amount.

Some of the statistical data on key attributes are:-

Amount

| | |
|---|---|
| Min. : | 0.00 |
| Median : | 22.00 |
| Mean : | 88.35 |
| 3rd Qu.: | 77.17 |
| Max. : | 25691.16 |

Table 6.1

Class

| | |
|---|---|
| 0 | 284315 |
| 1 | 492 |

Table 6.3

After preprocessing Class

| | |
|---|---|
| 0: | 284315 |
| 1: | 284298 |

Table 6.3

## VII. Conclusions

From the results we can conclude that the proposed model is giving better accuracy than normal LDM, Decision Tree and Naïve Bayes. We have also tried implementing SVM and Random Forest but the dataset is to big and the model is consuming excess time and resources which says that they are not the perfect models for big datasets like this. The ADASYN+ADABOOST combination of algorithms is giving an accuracy of 98% which is really good in comparison and it also proves better since SMOTE has a really huge chance of overfitting the data which might give more accuracy but cause metric trap.

## VIII. References

[1] Zhinin-Vera, Luis, Oscar Chang, Rafael Valencia-Ramos, Ronny Velastegui, Gissela E. Pilliza and Francisco Quinga-Socasi. “Q-Credit Card Fraud Detector for Imbalanced Classification using Reinforcement Learning.” International Journal of Advanced Computer Science and Applications 11, no. 2 (2020): 360-369.

[2] Xie, X., Wang, S., Yang, D., Zhang, Y., & Wu, J. (2020). Generative Adversarial Network-Based Credit Card Fraud Detection. In Q. Liang, X. Liu, Z. Na, W. Wang, J. Mu, & B. Zhang (Eds.), Communications, Signal Processing, and Systems. CSPS 2018. Lecture Notes in Electrical Engineering, vol 517 (pp. 122-131). Springer, Singapore. doi:10.1007/978-981-13-6508-9_122

[3] IEEE. "Credit Card Fraud Detection Using Sparse Autoencoder and Generative Adversarial Network." In 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 1-6. 2018.

[4] Shenvi, P., Samant, N., Kumar, S., and Kulkarni, V. "Credit Card Fraud Detection using Deep Learning." In 2019 IEEE 5th International Conference for Convergence in Technology (I2CT), 1-5. Bombay, India: IEEE, 2019. doi: 10.1109/I2CT45611.2019.9033906.

[5] RB, Asha, and K R, Suresh. "Credit Card Fraud Detection Using Artificial Neural Network." Global Transitions Proceedings 2 (2021): doi:10.1016/j.gltp.2021.01.006.

[6] Raza, M. Tayyab, and Kim-Kwang Raymond Choo. "Credit Card Fraud Detection in Card-not-present Transactions: Where to Invest?" Applied Sciences 11, no. 15 (n.d.): 6766. doi:10.3390/app11156766.

[7] Ahmad, H., Kasasbeh, B., Aldabaybah, B. et al. "Class Balancing Framework for Credit Card Fraud Detection Based on Clustering and Similarity-Based Selection (SBS)." International Journal of Information Technology 15, no. 1 (2023): 325-333.

[8] Seera, M., Lim, C.P., Kumar, A. *et al.* An intelligent payment card fraud detection system. *Ann Oper Res* (2021).

[9] Patel, Viral, Dhruvil Shah, and Maheshkumar H. Kolekar. "Credit Card Fraud Detection Using State-of-the-art Machine Learning and Deep Learning Algorithms." IEEE Access 10 (April 2022): 39700-39715. doi:10.1109/ACCESS.2022.3166891.

[10] S P, Maniraj, Aditya Saini, Shadab Ahmed, and Swarna Sarkar. "Credit Card Fraud Detection using Machine Learning and Data Science." International Journal of Engineering Research 8, no. 9 (2019): 126-131. doi:10.17577/IJERTV8IS090031.

[11] Fayyomi, Aisha, Derar Eleyan, and Amina Eleyan. "A Survey Paper On Credit Card Fraud Detection Techniques." International Journal of Scientific & Technology Research 10 (2021): 72-79.

[12] Benchaji, I., Douzi, S., El Ouahidi, B. et al. "Enhanced Credit Card Fraud Detection Based on Attention Mechanism and LSTM Deep Model." Journal of Big Data 8, no. 1 (2021): 151. doi: 10.1186/s40537-021-00541-8.

[13] Dornadula, Vaishnavi Nath, and Geetha, S. "Credit Card Fraud Detection using Machine Learning Algorithms." Procedia Computer Science 165 (2019): 631-641. ISSN 1877-0509. doi:10.1016/j.procs.2020.01.057.