Credit Card Fraud Detection

Haritha Madana
Computer Science Student
of PES University
Bangalore, India
harithamadana@gmail.com

Sowhith Reddy
Computer Science Student
of PES University
Bangalore, India
sowhithreddy3@gmail.com

K J Pavithra
Computer Science Student
of PES University
Bangalore, India
kjpavithra2000@gmail.com

Sreenath Saikumar
Computer Science Student
of PES University
Bangalore, India
sreenathsaikumar@gmail.com

***Abstract*—It's important for credit card and e-commerce companies to identify and prevent fraudulent credit card transactions. This issue can be tackled by Machine Learning. This project shows the modelling of a dataset consisting of transactions of different users. All transactions are modelled and the new transactions are tested based on the models developed. This is a basic example of classification. We have used algorithms like- Outlier Factor, Isolation Forest, Logistic Regression, Random Forest, Support Vector Classifier, Neural Networks and Adaboost.**

Keywords—Machine Learning, Data Science, Credit Card, Fraud, Credit Card Fraud, Neural Networks, SVC, Random Forest.

I. Introduction and background

The usage of credit cards has increased significantly over the past few years. However, a major part of credit card transactions are fraudulent and millions of money are stolen every year. It is a dark side of digital commerce becoming more prevalent day by day. The trust of customers is jeopardized. Credit card fraud leads to a loss of millions of dollars every year to customers plus the firm. Most E-commerce application transactions are through credit card and online banking- these systems are vulnerable with new techniques and attacks at an alarming rate.

Studies say that crime groups organized internationally affect non-cash payments on a global level and dominate the criminal market of credit card fraud. These incomes are afterwards invested in order to develop further fraudulent strategies, for money laundering or also to finance other criminal activities. However, there are some factors to be kept in mind- the fraudulent events are less represented than the genuine ones. Also, fraudsters may "improve" their strategies over time- which calls for continuous updation and enhancement of the current models in use.

It is essential for Credit card companies to be able to detect fraud in their transactions. Our challenge is to recognize fraudulent users in a credit card company based on various attributes and features and observing the trends in our dataset. Suspicious events are checked and can be reported.

Our aim is to make a classifier capable of detecting credit card fraudulent transactions. We apply a mixture of ML algorithms which distinguish each transaction. User behaviour is used to verify patterns that are not usual. Fraud detection in banking is one of the vital aspects nowadays since finance is a huge sector in society.

There can be two approaches to detection of credit card fraud- supervised and unsupervised ones. In the first, previous records are labeled as acceptable or not(as legal); the ML algorithm learns over this and creates a model applied to new data. The latter automatically detects patterns considered "alright" for some users. Generally, supervised approaches are better in detection of illegal transactions. Our research made us find that most data mining models for credit card fraud detection are based on Artificial Neural Networks (ANN)- in which a set of nodes process an input signal by interacting between them.

We trained the data using the following classifier algorithms-

- Local Outlier Factor
- Isolation Forest
- Logistic Regression
- Random Forest
- Support Vector Classifier
- Neural Networks
- Adaboost.

We also checked the precision, recall and F1 scores of the models.

A machine learning approach is extremely helpful to the process of detecting fraud in credit card transactions. It provides higher accuracy than its counterparts, lesser manual work, fewer false declines- and most importantly, the ability to identify new patterns and adapt to changes.

II. Previous work

A. Approaches

A variety of approaches have been proposed to solve the problem of credit fraud detection.

Credit card fraud detection using Machine Learning Techniques: A Comparative Analysis (2017) This paper investigates the performance of naïve Bayes, k-nearest neighbour and logistic regression on highly skewed

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

(towards legitimate transactions) credit card fraud data and so a hybrid technique of under-sampling negative class (legitimate) and oversampling positive class (fraud) is carried out on it. The results show that hybrid sampling greatly improves the performance of binary classification and kNN shows significant performance for all metrics in sampled datasets whereas logistic regression classifier performs better on the un-sampled dataset. Accuracy on 34:66 sampled dataset: 97%, 97%, 54% each.

Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy (2018) Focused on the challenges of the problem that occur in the real world scenario like concept drift, class imbalance and verification latency. Tied up with an industrial partner to understand the working of a real-world fraud detection system that consisted of five layers of control: the terminal, transaction-blocking rules, scoring rules, data-driven model, investigators. They focused on making an efficient data driven model with an effective learning strategy that consists in separately training a classifier on feedback and a classifier on delayed supervised instances and then combining their posteriors to find alerts. They then assess the overall fraud-detection performance of the data sets both by averaging daily performance measures (Pk, CPk, and AUC) and also by analyzing the sum of classifiers' ranks each day. They achieve an accuracy of 92.30%.

Credit Card Fraud Detection Based on Whale Algorithm Optimized BP Neural Network (2018) This paper proposes a credit card fraud detection technology based on whale algorithm optimized Back propagation neural network focusing at solving the problems of slow convergence rate, easily falls into local optimum, network problems and bad system stability caused by the back propagation neural network. The initial solution space of the problem is transformed into a search space that can be processed by the whale algorithm, the fitness function is used to evaluate the weight, and the obtained whale population optimal solution is used as the initial weight and threshold of neural network training and the training error of BP neural network is calculated, the weight and threshold are changed according to this fitness function. Achieves an accuracy of 96.40%.

Ensemble Learning for Credit Card Fraud Detection (2018) Aimed to combat the relative scarcity of fraudulent transactions by trying to either oversample, undersample, a combination of the two or use the SMOTE technique. They initially used unsupervised learning techniques: K-means clustering, DBSCAN and a mixture of Gaussians. They noticed that the fraudulent transactions were neither outliers nor were they restricted to one or more clusters and were somewhat uniformly distributed in all clusters. Finally, implemented an ensemble of Random Forests and Feed-forward Neural Networks. The final model yielded a precision of 85.85%, a recall of 86.73% and an accuracy of 99.95% but owing to the unbalanced dataset, they mentioned that accuracy cannot be considered to be a good measure of model performance.

Credit card fraud detection using machine learning with integration of contextual knowledge (2019) Proposed a multi-perspective HMM-based automated feature engineering strategy. Modeled the genuine and fraudulent behaviours according to the timing and the amount of the transactions. The HMM approach provides automated feature engineering to model temporal correlations. This feature engineering strategy is relevant for various types of classifiers (random forest, logistic regression and Adaboost) and robust to hyperparameters choices made for constructing the features. This approach achieved an accuracy of 74.54%.

B. Inferences and Problem Scope

Analysing the research papers has led us to understand that most of them focus on implementing one model for a problem whose solution depends on the size of the data and the type of sampling used. And so instead of focusing on building one solid model, we want to do a comparative analysis of different approaches, both supervised and unsupervised to understand what kind of approach suits which kind of data. The literature survey has also pointed out the significance of precision, recall and f1 score over accuracy for such imbalanced datasets.

The scope of our problem is detecting credit card frauds using an imbalanced dataset. We use the Machine Learning Group - ULB dataset.The dataset contains transactions made by credit cards in September 2013 by European cardholders. Since credit card details are confidential information, the dataset we use has the principal components of the original confidential features along with the features 'Time' and 'Amount'.

III. PROPOSED SOLUTION

We start our solution with preprocessing and EDA followed by exploration of different approaches in order to narrow down on what approach gives the best precision, recall and F1 score with the given data.

Preprocessing and EDA

Finding the number of rows and columns:

```
[ ] df.shape

    (284807, 31)
```

There are 284807 rows and 31 features

Analysing the fraud and not_fraud cases separately:

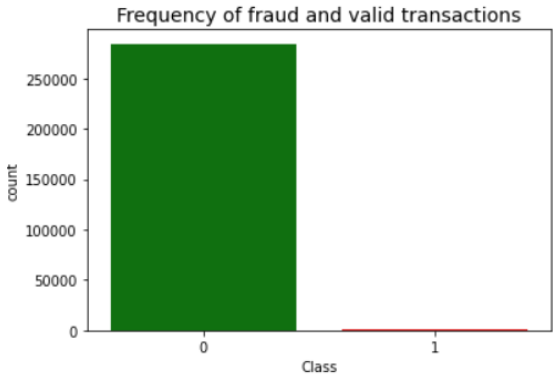
```
# analysing the fraud and not_fraud cases separately
print("Valid Transactions:")
df.Time[df.Class==0].describe()
```

```
Valid Transactions:
count    284315.000000
mean      94838.202258
std       47484.015786
min           0.000000
25%       54230.000000
50%       84711.000000
75%      139333.000000
max      172792.000000
Name: Time, dtype: float64
```

```
print("Fraudelent Transactions:")
df.Time[df.Class==1].describe()
```

```
Fraudelent Transactions:
count       492.000000
mean      80746.806911
std       47835.365138
min         406.000000
25%       41241.500000
50%       75568.500000
75%      128483.000000
max      170348.000000
Name: Time, dtype: float64
```

The number of fraud cases are very few when compared to the number of valid cases. The dataset is imbalanced. Also, the min for fraudulent transactions is 0 and for valid cases is 406. And most fraud cases' amounts are lower than that of valid transactions.We can see in the below diagram that it is a skewed distribution. (green-valid, red-fraud)



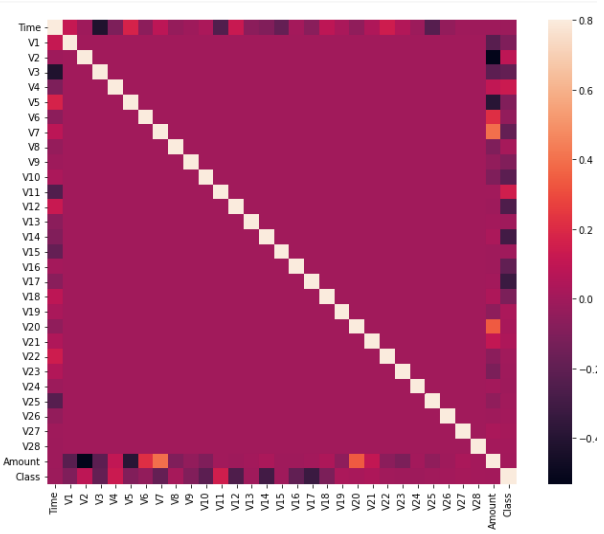
Outliers and percentage of fraud:

```
Count of outliers in the dataset= 370864
```

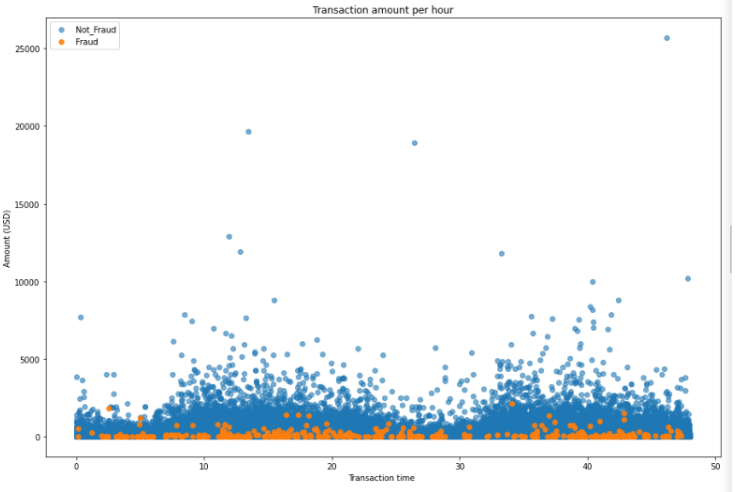
```
fraud_perc
```

```
0.1727485630620034
```

Heatmap of the correlation matrix:



A plot of transaction amount per hour and transaction time:



Here, we can see that that most of the frauds have a lower amount than valid transactions

A. Unsupervised Algorithms

We started testing our models with unsupervised algorithms. We make an assumption that frauds can be considered anomalies since the number of frauds is very less compared to the number of valid transactions. Moreover we also made the inference in EDA that fraudulent transactions have a lesser amount of money associated with them compared to valid transactions. We try out two anomaly detection algorithms- Local Outlier Factor and Isolation Forest.

- Local Outlier Factor (LOF)

  The anomaly score of every data point is called Local Outlier Factor. It calculates the local deviation of density of a given data point with respect to its neighbors which will be given in the model. Here we apply the LOF method on the original dataset without sampling. In order to do this, we set the number of neighbours as 20 and the contamination with the fraud fraction that we calculate since here we consider the frauds as

anomalies. We use sklearn to deploy this model and to print the accuracy and classification report.

- Isolation Forest (iTrees)

  Isolation Forest creates an ensemble of isolation Trees for a given data set and anomalies are the data points which have short mean path lengths on these isolation trees. It does random partitioning of features i.e. randomly chooses an attribute and then randomly selects a split value between the max and min values of that attribute in order to find the path lengths that differentiates the anomalies. Here we deploy the isolation forest using sklearn. We give the number of samples in the dataset, fraud fraction as contamination and random state as arguments to build this model. We notice that this performs better than LOF when we print the accuracy and classification report.

B. Supervised algorithms

- AdaBoost Model

  Since the dataset is very imbalanced and highly skewed towards legitimate transactions, we use a data augmentation algorithm called SMOTE (Synthetic Minority Oversampling TEchnique) to balance out the ratio of fraudulent transactions to legitimate ones. The training split (70%) is oversampled to now have 398016 entries. The Adaboost model has been implemented using the scikit-learn library with 50 decision stumps of depth 1, SAMME algorithm and a learning rate of 1. The AdaBoost model assigns weights to each classifier and readjusts weights based on whether a particular decision tree in the ensemble predicted correctly or not. These weights multiplied by the output of each decision tree gives us a final prediction.

- Logistic Regression Model

  Similar to the AdaBoost model, we use the oversampled training data to train a logistic regression model for binary classification. This model uses the simple $WX+b$ function where W is the weight assigned to a particular feature and b is the inherent bias followed by a sigmoid function with a decision boundary of 0.5.The sigmoid function returns a value between 0 and 1 for any input and essentially returns a probability that a given entry belongs to a particular class(1 or 0).It is run for 100 iterations before converging on the training data.

- Deep Neural Network

  We also implement a Deep Neural Network which is an extension of the logistic regression algorithm with each layer learning the finer details of the given data the deeper the network is. It has been implemented with the help of the Tensorflow library with 5 hidden layers of sizes 512,256,256,128 and 128 respectively. All of these layers use the 'ReLU' activation function which is essentially a max(0,x) function and a sigmoid function is used for the output layer which is of size 1 since there are 2 classes. The loss function used is binary cross entropy which is then used in backpropagation to readjust the weights if the the prediction is incorrect. The optimizer used here the adam optimizer which is a combination of the adagrad and RMSprop algorithms.

- Random Forest

  The dataset is highly unbalanced as there are 492 frauds in the dataset, even though there are 284315 no frauds in the dataset hence I used SMOTE(Synthetic Minority Oversampling Technique) works similar to k-means approach where it resamples data based on fraud cases, we split 70-30 data split for training and testing with random_state set to 42 and then we import the random forest classifier and use it on the sampled train and test and then fit onto the test dataset.

- SVM

  Since the dataset is unbalanced we use undersampling ,then we define training and testing set after applying a dimension reduction to illustrate the fact that nothing will be gained because a PCA was previously computed. We import the svm classifier and use it to train on the train data using the following parameters kernel = 'linear',chache_size=200,degree=3 , and we fit it onto the test dataset and we display confusion matrix accuracy etc.

IV. Experimental results

Finally, we look at the results we got on the models we deployed.

We noticed that the anomaly detection algorithms done on such an imbalanced dataset without sampling performed badly in comparison to the supervised algorithms. Both algorithms have an accuracy close to 99% but since they are anomaly detection algorithms we give significance to precision, recall and f1 score. We achieved only 0.05 precision, recall and f1 score with respect to fraudulent transactions using Local Outlier Factor which means we have a lot of false positives. Although the isolation forest does perform better with a precision, recall and f1 score of 0.34, it is still much lesser compared to the performance of supervised algorithms. Thus, we infer that such anomaly detection algorithms work better when sampled to balance the dataset and not otherwise. Also, since credit card details involve confidential information, there are no datasets available that can portray anomalies based on the features since features are always principal components in order to maintain confidentiality.

For the supervised algorithms, we observe that while they all have high accuracies on the test set, this is due to the fact that the testing data has not been oversampled and therefore has a very unbalanced distribution. A null classifier would have a high accuracy on the testing set regardless and therefore accuracy is of little use to us and the F1 score is of higher importance. The time

column/feature has been dropped since it introduced major inaccuracies during the training stage. The Neural Network model recorded an accuracy of 93% and an F1 score of 0.7, the Adaboost model recorded an accuracy of 95% but an F1 score of 0.1 while the logistic regression model had an accuracy of of 95% as well but an even lower F1 score of 0.098 which leads us to the conclusion that the NN model is more performant with far fewer false positives. Random Forest model records an accuracy of 99%, F1-0.847 but the precision is 84.89% and the recall is 86.764% where as in the SVM model we got an accuracy of 91.35% , F1-score of 0.03 along with precision or 0.015 however the SVM can be improved by re-balancing class weights.

Conclusions

Contribution of team members (alpha risk)

Haritha Madana - EDA, Report
K J Pavithra - EDA, Unsupervised algorithms (Local Outlier Factor, Isolation Forest), Report
Sowhith Reddy - Supervised algorithms(Random Forest ,SVM)
Sreenath Saikumar - Supervised algorithms(NN,Adaboost,LR)

References

[1] John O. Awoyemi; Adebayo O. Adetunmbi; Samuel A. Oluwadare, "Credit card fraud detection using Machine Learning Techniques: A Comparative Analysis", 2017

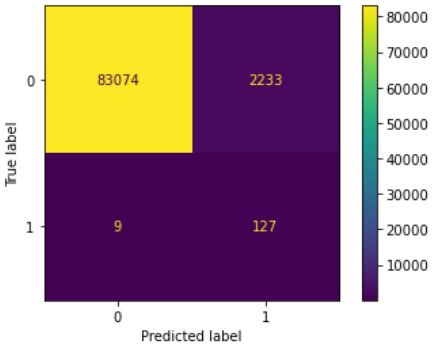
[2] Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi and Gianluca Bontempi, "Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy", 2018

[3] Ishan Sohony,Rameshwar Pratap,Ullas Nambiar, "Ensemble Learning For Credit Card Fraud Detection", 2018

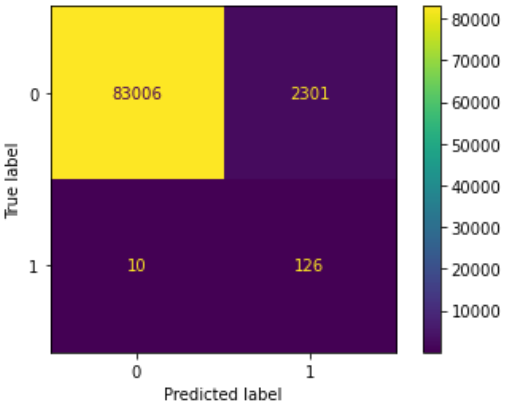
[4] Yvan Lucas, "Credit card fraud detection using machine learning with integration of contextual knowledge," 2019

[5] Chunzhi Wang,Yichao Wang,Zhiwei Ye,Lingyu Yan,Wencheng Cai,Shang Pan, "Credit Card Fraud Detection Based on Whale Algorithm Optimized BP Neural Network," 2018

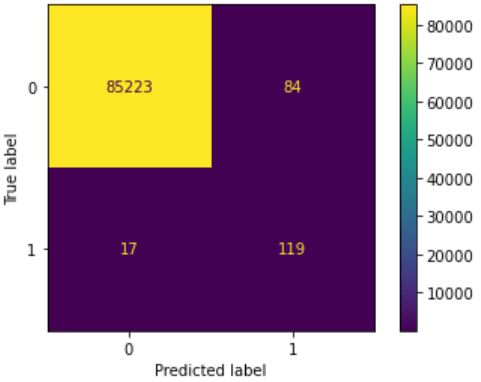
Further Visualisations



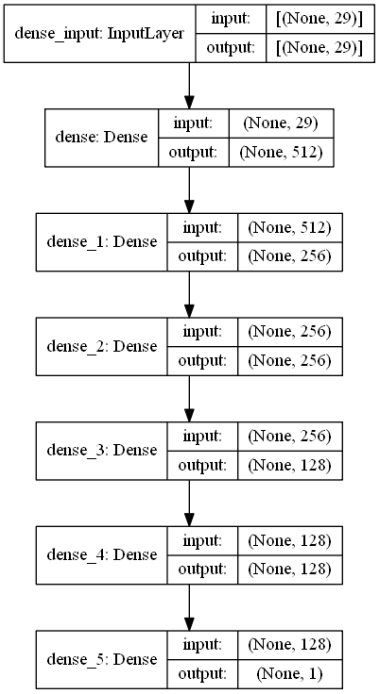
Confusion Matrix for the Adaboost Classifier



Confusion Matrix for the LR Classifier



Confusion Matrix for the Deep Neural Network



Deep Neural Network Diagram