Credit Card Fraud Detection

Rishab Kamshetty
Computer Science
California State University
Sacramento, USA

Ashutosh Deowanshi
Computer Science
California State University
Sacramento, USA

Abstract: **Nowadays when we are moving towards a cashless economy means instead of using cash for purchasing any items, we are using plastic money. One form of plastic money is Credit Card. They are very convenient to use. But with the increasing use of Credit Card, there's a new threat that arose to the financial industry which are frauds related to the credit card. Fraud transaction can be classified as those transactions which are not authenticated by the credit card holder itself. Handling credit card transactions are a very tedious task. Credit card companies need to face many challenges to tackle hackers and threats. It's very important for Credit card issuer companies to detect these kinds of frauds and don't make the credit card customer pay for it as they are not the ones who did that transaction. Tackling credit card fraud transactions is sometimes becomes very challenging because of the following reasons. Firstly because of the profiles of genuine and fraudulent changes every time. Secondly, the credit card fraud dataset is highly crooked. In this paper, credit card fraud detection is done using several machine learning models which comprises supervised, unsupervised and neural network. It's interesting to see how the dataset behaves with our data set.**

Keywords—autoencoders, neutral network, unsupervised, abnormality, transactions, outlier.

I. INTRODUCTION

With the increasing need for internet from one decade, people get more attracted to online shopping for their daily needs, this leads to an increase in doing online transactions by credit card. Offline shopping payment is too now mostly made by credit cards. Both online and offline credit card transactions lead to an increase in credit card frauds. The fraud rate has been increasing since the last decade, is also estimated to raise a lot in the following years, as people are continuously depending on e-wallets and online purchases.

A great deal of investigates has been given to recognition of external card frauds which represents a majority share of credit card frauds. Identifying fake transactions utilizing conventional strategies for manual recognition is tedious and wasteful, accordingly, the coming of huge information has made manual techniques progressively illogical. In any case, monetary establishments have centered thoughtfulness regarding later computational procedures to deal with credit card fraud issues. The report firstly features the examination of the accessible information by indicating the relations between various attributes. The later sections shortly describe literature scrutiny, data cleaning, feature selection, implementation of different classification models on the available data set and its evaluation by comparing all the models with each other.

II. BRIEF OF DATASET

The datasets contain exchanges made by credit cards in September 2013 by European cardholders. This dataset presents exchanges that happened in two days, where we have 492 frauds out of 284,807 exchanges. The dataset is exceptionally uneven, the positive class (frauds) represent 0.172% of all exchanges. It includes only numerical input variables which are the outcome of a PCA transformation. Lamentably, due to confidentiality issues, we cannot afford the real features and more background knowledge about the data. Features V1 to V28 are the principal components taken with PCA, the only features which have not been remodeled with PCA are 'Time' and 'Amount'. Feature 'Time' holds the seconds elapsed between every transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be utilized for example-dependent cost-sensitive learning. Feature 'Class' is the acknowledgment variable and it takes value 1 in case of fraud and 0 otherwise.

III. PROBLEM

Credit Card Fraud detection includes demonstrating the card transactions utilizing the transaction history dependent on the ones that ended up being a fraud. Significantly, Credit organizations can perceive fraudulent credit transactions so that clients are not charged for things that they didn't order.

Administering Imbalanced Data Set: Most of the transactions were Non-Fraud (99.83%) of the event, while Fraud transactions occur (0.17%) of the time in the data frame. There are a clear dominant part and minority class in distribution. Information is profoundly slanted. We will counter approaches to manage this issue in the notebook.

IV. DATA EXPLORATION

Firstly, the dataset was raw, that was collected from Kaggle. There might be many values that may be redundant to us and the dataset may not be clean to apply models on it. Here in data exploration we will look for such unnecessary data or values and avoid them. To start with exploration, we will check whether the dataset has any missing values in it. Because of missing values will give the wrong prediction. In our data frame, we checked for missing values and there were none.

Next, we see the distribution of each class of transactions and almost all the transactions are of an amount less than $2500. So, we plot a graph that has transactions less than $2500.

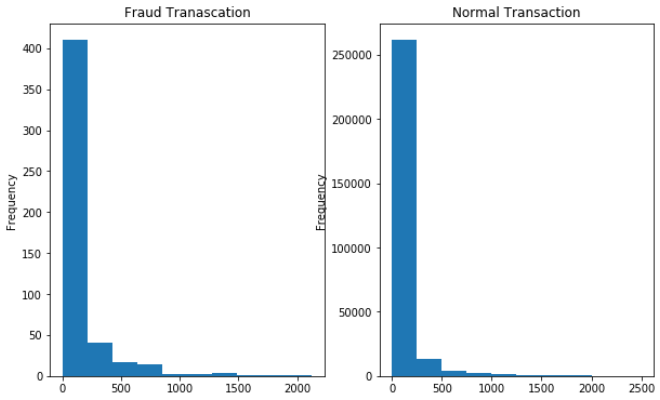

Fig. 1. Histogram of Class distribution based on amount(<2500)

To explore the data easily, we used a histogram to plot and look into each feature. The normal distribution of the feature is clearly depicted by the graph. By this, we show the range of values. As mentioned earlier our data is highly unbalanced, most of the transactions are valid and very few transactions are of class fraud. The Fig. 2 shows the comparisons between the two classes.

```
0    284315
1       492
Name: Class, dtype: int64
Text(0, 0.5, 'Frequency')
```

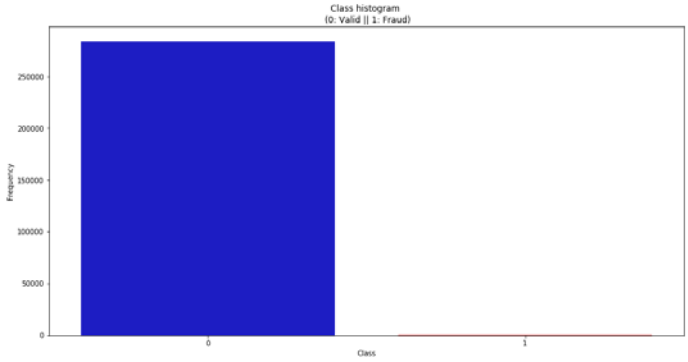

Fig. 2. Class Histogram

284315 of the transactions are of class genuine (blue bar) and only 497 of the transaction are of class fraud (red bar) which accounts for only 0.17% of the total transactions.

To check how one feature is related to another, we use a correlation matrix. By using this matrix, we came across that, features V1 to V28 are not correlated to each other. Most of the correlation is with Time, Amount, and Class. So, we only scale time and amount using scalar.

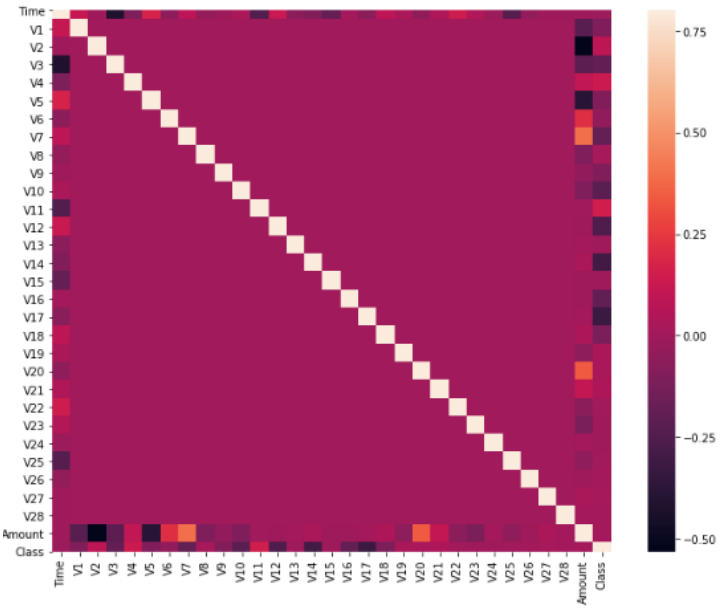

Fig. 3. Correlation matrix

V. Techniques

A. *Synthetic Minority Oversampling Technique(SMOTE)*

SMOTE creates a random synthetic point in the data to create a new sample data to increase the number of minority classes. Here we use make to use SMOTE to increase the number of transactions for fraud class.

B. *Sub-Sample(1:1 ratio)*

Sub-sample is a dataframe that has a 50/50 ratio of the classes. Sub-sample is used to overcome the problem such as overfitting and wrong correlation. Here we make use of subsample to make the transactions as a 50/50 ratio of genuine and fraud.

C. *Anomaly detection algorithms*

These algorithms will help us to detect outliers in the data. The data point which is completely different from the majority of the surrounding data points is recognized and cast out from the rest. Here we used existing Anomaly detection models like Isolation Forest, Local Outlier Factor algorithm. To compare with existing models, we built an Autoencoder neural net.

VI. Models

A. *Supervised Models(Classification methods)*

1) *Logistic Regression*

This regression model is appropriate to use when the variables are dependent. It gives a relation among binary variables and more than one nominal, interval, an ordinal variable that is independent. It has only two possible values and will be labeled as 1 and 0.

2) *Linear Discriminant Analysis*

This model is easy in the case of univariate. It reduces the dimensions. It also detects overlapping. LDA is linked to

regression analysis, ANOVA (analysis of variance) and PCA (Principal component analysis). It works when the determination done on variables that exist independent are consecutive numbers.

3) Support Vector Machine

This algorithm is a discriminative classifier that is plotted to separate a hyperplane. It categorizes the hyperplane into new examples which then divides into two parts, each of different classes. However, if the data is scattered in a nonlinear way, then SVM doesn't handle it well.

B. Supervised Models(Ensemble method)

1) Random Forest

It is an ensemble method that makes use of bagging and decision tree techniques. Instead of using individual decision trees, the random forest used to merge several trees to find the output.

2) XGBoost

This stands for Extreme Gradient Boot, essentially is an implementation of gradient boosted decision trees in consecutive order. Each variable that is independent has been specified weights, which are served to a decision tree that predicts the output by creating a non-correlated trees.

C. Unsupervised Models(Anomaly Detection)

1) Isolation Forest(existing model)

This regression model is appropriate to use when the variables are dependent. It gives a relation among binary variables and more than one nominal, interval, an ordinal variable that is independent.

2) Local Outlier Factor(existing model)

Calculates the local deviation of the density of a provided sample concerning its neighbors, and tells us that few of the data points as outlies.

3) Autoencoders Neural Network

This is a type of neural network that gets input and decodes it. The input which is dataset here is decoded to its core features and the method is reversed to reconstruct the input again. This is useful when we come across the anomalies which are like outliers. These anomalies or fraud will suffer a higher reconstruction error.

VII. EXPERIMENT

A. Splitting data

Before proceeding to apply the technique to handle imbalanced data, we split our dataset into a training set and testing set. We have split as train:80% and test:20%.

B. Supervised model using a SMOTE

After splitting the dataset, we used SMOTE from the imblearn library. Applied SMOTE on the training set. To test the dataset, we built a Random Forest model and applied it on the test set.

- Precision recall curve when we predicted the result using SMOTE on Random forest model

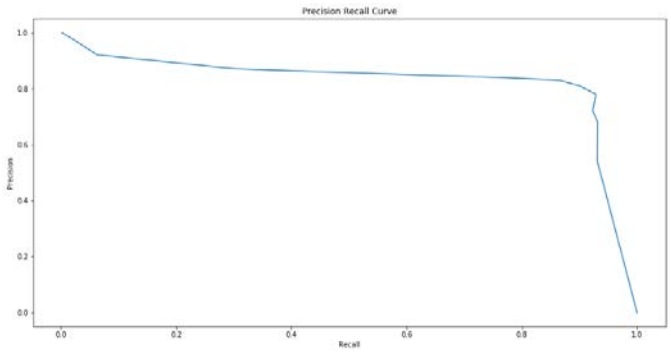

Fig. 4. Precision recall curve plot

Results of Supersvised Model Using SMOTE

- When we used SMOTE together with Random Forest, we got the best score of an F2 score of **0.82653**.
- Later, we then applied Random Forests with some Hyper Parameters without SMOTE then we got an F2 score of **0.77083**.

TABLE I. SUPERVISED MODEL USING SMOTE

Models	Our Classification results			
	Recall	***Accuracy***	***Precision***	***F2 score***
RF with SMOTE	0.81	0.9995	0.90	**0.8265**
RF without SMOTE(Hyper Parameter)	0.7708	0.9994	0.925	**0.7708**

C. Supervised model using a sub-sample data frame

On the sub-sampled data frame, we applied the 3-classification model and 2-ensembled models. These models are inspired by exciting kernel models [7]. We have implemented this to compare with our models and techniques results. The result is a confusion matrix report and Area under the curve

TABLE II. SUPERVISED MODEL USING SUBSAMPLE TECHNIQUE

Models	Existing Classification results		
	Model Recall (in %)	***AUPRC (in %)***	***Precision (in %)***
LR	92.222	96.715	94.318
LDA	85.555	92.593	**95.061**
SVM	72.222	94.359	91.549
RF	91.111	**98.416**	94.252
XGB	**94.444**	**98.417**	94.444

Results of Supervised Models with Sub-Sample

- LDA: Lowest AUPRC (92.593%)
- Standard SVM: Lowest recall (72.222%)
- Standard RF and Standard XGBoost: Both high AUPRC (98.41%) and Recall (94.44%)
- To improve the accuracies of the supervised models we applied tuning to SVM and XGBoost model. Tuning the model gives us better results.
    - For SVM we explored the two of the parameters: C and kernel.
    - For XGBoost, we applied an estimator of 60 and max depth of 2

TABLE III. SUPERVISED MODEL WITH TUNING

Models	**Our Classification results**	
	Recall (in %)	***AUPRC (in %)***
Tunned SVM	85.899	**98.630**
Tunned XGB	91.12	**98.694**

- Tuned XGBoost: Best AUPRC (98.630%) with a high Recall (91.1%)
- Tuned SVM: Very high AUPRC (98.694%) but lowest Recall (85.89%)

D. *Unsupervised model*

Firstly, we applied the existing models, those are Isolation forest and local outlier detection model referred from the 2016 kaggle notebook [4].

TABLE IV. UNSUPERVISED MODELS(EXISTING MODELS)

Models	**Existing models Results**
	F1 Score
Isolation Forest	0.9013
Local Outlier Factor	0.8988

To improve upon the existing model, we implemented an autoencoder using Keras to build a neural network. Autoencoders will help us to detect fraud transactions easily. As the fraud transactions are anomalies will face higher reconstruction errors.

We built the autoencoders model using four fully connected layers with different numbers of neurons. Two of the layers are used by decoders and the other two by encoders. We also used regularization L1. Once the model was built, we applied to the training set. Fig.5 below shows the loss model. Orange plot displays the test set and blue illustrates train set.

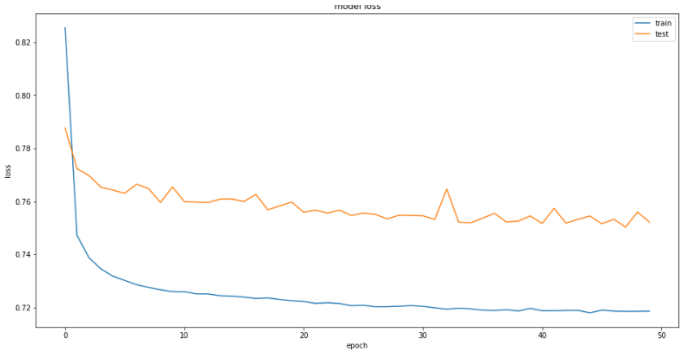

Fig. 5. Model loss

Our model will be separately testing on fraud transactions and genuine transactions. The same model is applied to the fraud transactions it had reproduced different results with more error. Fig.6 below shows the reconstruction error, from the figure, we can see the transaction of fraud (orange points) have faced an higher reconstruction error

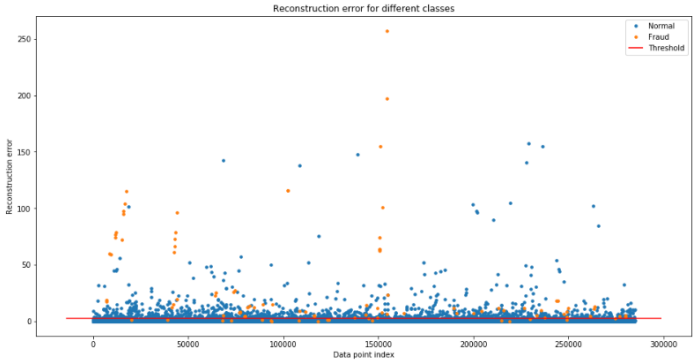

Fig. 6. Classification Report Of Autoencoders Forest

Initially, we trained the model with 100 epochs and a batch size of 128. From around 35 epoch the accuracy stayed the same to around ~0.7543, whereas the final accuracy was AUPRC 88.032. Later, we also tried with 50 epochs and a batch size of 32, to which we got the highest accuracy on test dataset

Experimenting Our Model(epoch)	**Our Results**
	Accuracy
Autoencoder(100 and batch size 128)	88.11
Autoencoder(50 and batch size 32)	90.008

To evaluate the results, we used a classification report for the 50 epoch model which gave an improved F1 score of **0.96** and AUPRC of **0.90087.**

	precision	recall	f1-score	support
0	1.00	0.97	0.98	56864
1	0.04	0.84	0.08	98
accuracy			0.96	56962
macro avg	0.52	0.90	0.53	56962
weighted avg	1.00	0.96	0.98	56962

Fig. 7. Classifcation Report Of Autoencoders Forest

TABLE V. AUTOENCODER NEURAL NETWORK

Models	Our Results
	F1-Score
Autoencoder NN	**0.96**

VIII. CONCLUSION

In this paper, we discussed a different approach that is being applied to check credit card fraud, how credit card fraud influences the financial institution as well as solicitor and purchaser, fraud detection procedure used by VISA and MasterCard. Achieving SMOTE on our imbalanced dataset supported us with the imbalance of our labels (more no fraud than fraud transactions). We randomly picked half of the dataset as valid transactions, the other share being frauds. We were conscious of the influence of the Recall, to identify fraudulent transactions and reduce False Negative rate. The area under the Precision-Recall curve, to avoid the False Positive rate to skyrocket. The neural network is the newest method that is being used in several areas due to its persuasive abilities of knowledge and predicting. In this, we try to use this ability of neural networks in the area of credit card fraud detection. We've created a very modest Autoencoder that can replace what nonfraudulent transactions look like. We gave a lot of one-class samples (normal transactions) to a model and it learned (somewhat) how to specify whether new examples belong to that same class.

IX. FUTURE WORKS

While solving the credit card fraud detection problem, we got to see that there's a need for a very huge amount of data in which the transactions are captured and this should bring up with some pattern of the purchases in which way the customer made the transactions. As we saw in all of our models including Neural Network, they need to train effectively on those data but the difficulty arises at the beginning stages when there's a very less or not at all transactions has been occurred, in this case, we need to figure how we are going to train our model which such less amount of data captured because we know in order to make Neural Network work effectively to predicts we must have some pattern available from the dataset from which Neural Network can get trained and must predict effectively. That's why we must think to design some model that can restrict the fraud without being any actual transaction happened in the past.

REFERENCES

[1] WorldPay. (2015, Nov). Global payments report preview: your definitive guide to the world of online payments. Retrieved September 28, 2016. http://offers.worldpayglobal.com/rs/850-JOA856/images/GlobalPaymentsReportNov2015.pdf

[2] The Nilson Report. (2015). Global fraud losses reach $16.31 Billion. Edition: July 2015, Issue 1068.

[3] Credit Card transactions from Kaggle: https://www.kaggle.com/mlg-ulb/creditcardfraud

[4] Pavan Sanagapati, "Anomaly Detection - Credit Card Fraud Analysis" https://www.kaggle.com/pavansanagapati/anomaly-detection-credit-card-fraud-analysis

[5] M. Shell. (2015, Aug.) The IEEEtran.cls package. [Online]. Available: http://www.ctan.org/pkg/ieeetran

[6] (2015, Jul.) IEEEtran homepage. [Online]. Available: http://www.michaelshell.org/tex/ieeetran/

[7] Janio Martinez, "Credit Fraud || Dealing with Imbalanced Datasets" https://www.kaggle.com/janiobachmann/credit-fraud-dealing-with-imbalanced-datasets

[8] Credit Card transactions from Kaggle: https://www.kaggle.com/mlg-ulb/creditcardfraud

[9] Y. Sahin and E. Duman, "Detecting Credit Card Fraud by Decision Trees and Support Vector Machines", International Multiconference of Engineers and computer scientists March, 2011.

[10] (2015, Jul.) The IEEE website. [Online]. Available: http://www.ieee.org/

[11] S. Benson Edwin Raj, A. Annie Portia "Analysis on Credit Card Fraud Detection Methods".

[12] IEEE-International Conference on Computer, Communication and Electrical Technology; (2011). (152- 156).

[13] Siddhartha Bhattacharyya, Sanjeev Jha, Kurian Tharakunnel, J. Christopher Westland, "Data mining for credit card fraud: A comparative study", Decision Support Systems 50 pp. 602–613,2011.

[14] Tao Guo, Gui-Yang Li "Neural Data Mining For Credit Card Fraud Detection". IEEE, Proceedings of the Seventh International Conference on Machine Learning and Cybernetics; (2008). (3630-3634).

[15] M N Pathak "FC network with Tensorflow" https://www.kaggle.com/mnpathak1/fraud-detection-analysis-with-nn

[16] Dataman "Feature Engineering for Credit Card Fraud Detection" https://towardsdatascience.com/how-to-create-good-features-in-fraud-detection-de6562f249ef

[17] Hung, W. N. N., Song, X., Aboulhamid, E. M., & Driscoll, M. A. (2002). BDD minimization by scatter search. IEEE Transactions on Computer-Aided Design on Integrated Circuits and Systems, 21(8), 974–979.