Credit Card Fraud Detection Using Interpretable Black Box Model

Ajinkya Pawale (ajpawale@iu.edu), Abhishek Sharma (absharm@iu.edu)
Indiana University Bloomington

Abstract

In credit card fraud detection problem, we are trying to tackle two problems- one is to eradicate the data imbalance between fraud and genuine transactions and the other is to provide interpretation of a black box model's output to predict the possibility of fraud transactions. To tackle the data imbalance, two feature engineering approaches were tried namely SMOTE and random under sampling. Results of SMOTE were better as compared to under sampling and hence it was used to balance the fraud transaction data with the genuine transaction data. Various Machine Learning models namely SVM, Decision Tree, Logistic Regression and KNN were used to build a prediction system. Along with these ML algorithms, a Deep Neural Network (DNN) was also trained on the data and used for prediction. From the precision-recall curve and ROC-AUC curve comparison we found that Deep Neural Network performed better as compared to the other Machine Learning models. Since DNN is a black box system it becomes important to understand the explanibility of the system as we need to know why the decision was made along with the actual decision. To attain that, we have two white box explainers LIME and SHAP added to the top of our DNN model to better understand the outputs of our prediction. Prediction analyses by two explainers are presented, providing a clear picture of how each attribute of an interest instance contributes to the final model output.

Introduction

Card fraud over the next decade will cost the industry a collective $408.50 billion in losses globally, according to an annual report from the industry research firm Nilson Report. The industry will lose an estimated $49.32 billion to fraud by 2030, when total payment card usage is forecast to reach $79.14 trillion. The numbers predicted are huge and it becomes extremely important to devise methods to prevent card fraud. To combat this problem, banks and internet payment firms use anti-fraud software to identify fraudulent transactions. The most widely used algorithms for fraud detection are Machine Learning and Deep Learning based. [1] states that many of the credit card fraud detection techniques used ML algorithms. However, many of them yielded low classification accuracy, False Positive and Negative rates. This is likely because, the techniques were not combined with good and effective feature selection and parameter optimization techniques. It states that future work should consider focusing on constructing accurate classification models based on hybrid structure containing ML and NI-techniques (Genetic Algorithms and Artificial Immune Systems).

Credit card fraud detection is not a conventional classification task because fraudulent transactions are extremely rare. The dataset is heavily skewed, which leads models to perform badly when uncommon cases are encountered. Thus, dealing with imbalanced data is a major task in these kinds of problems. [2] [3] [4] are ML based methods consisting of SVM, Decision Trees, Ensembling methods which work towards the classification without dealing with the data imbalance. On the skewed data, [5] applies a hybrid strategy of under-sampling and oversampling. The results suggest that naive bayes, k-nearest neighbor, and logistic regression classifiers have the best accuracy. The main goal of [6] is to use the Near-Miss Under-sampling Method to balance the dataset. Decision Tree, with 97 percent accuracy, is the best algorithm discovered in this research, compared to Naive Bayes, which has 90 percent accuracy.

In terms of DL-based techniques, there have been attempts at high-quality data augmentation. [7] states that in comparison to the other GANs, the Wasserstein-GAN is more stable in training and produces more realistic fraudulent transactions. Apart from that feature-based techniques have also been used for classification. [8] uses spatio-temporal attention based neural network and [9] uses OCAN that consists of LSTM- Autoencoder and complementary GAN for fraud detection. Few research like [10] regard this problem as anomaly detection, which naturally avoids the imbalance problem and has more space for development.

The model explainability [11], a hotspot in the ML field, is another important problem for DL-based credit card fraud detection systems. Since DL based black box systems lack explainability, it becomes important to add interpretability to the model so that it can be used to explain a reason why a transaction was flagged as fraudulent for official court reports. [12] suggests using an adversarial trained anomaly detection model to solve the problem of credit card fraud detection. The GAN network basically creates samples of the fraud cases to augment the dataset with more fraud transactions. Then the cases from the GAN along with the original data are passed to a classifier which is multi-layer perceptron for detecting weather the transaction is fraud or genuine. This fraud detection model's input-output

relationships are investigated using LIME, and individual fraud transaction records are checked for local explanations.

Our approach tries to use feature engineering approach to tackle the data imbalance. [7] gives the AUC, AUPRC, f1 score of SMOTE along with the WCGAN, the performances were comparable. Thus, we tried to implement SMOTE instead of a black box system to implement the data augmentation. Adding another black box model and trying to explain that will make things complicated and can be avoided. Thus, we try to tackle that problem and use SMOTE for data imbalance. SMOTE uses KNN to generate samples of the class that is less in number. Further, our approach uses an ANN model to classify the examples and then explanations are provided using LIME and SHAP. LIME focuses more on local explanation for individual records that need to be looked at and SHAP focuses on global explanations to understand how the model behaves overall. Both these explainers provide explanations in terms of the feature value thresholds so that it becomes easy to interpret which feature was responsible for a particular output.

Methodology

A. Data

This dataset consists of the data from a mobile payment application. It has information about the sender and the recipient of a particular transaction including Sender and Receiver ID, Sender and Receiver's Old and New Balance (before and after the transaction), amount of the transaction and the type of transaction. Apart from these features, there is another column which classifies the transaction as Fraud or Not Fraud and a column which keeps track of the time step of the transactions. Out of these, the receiver's and sender's id, and the Transaction Type are text type, and all the other columns are numeric type.

1) Cleansing: Several data cleansing methods were performed on the dataset and as a result, a few extra columns were added. They are as follows:
mean_amount: the mean amount of all the transactions; *count_orig*: total count of a sender in the dataset; *count_dest*: total count of the receiver in the dataset; *step_dest_count*: total count of the receiver in a particular times frame

These columns proved to be helpful in analyzing the pattern in the data and thus provide useful explanations as to whether a transaction is fraud or not.

2) Sampling: After the data cleansing step, to create a balance in this highly unbalanced dataset three different approaches were followed: Random Under-Sampling, Random Over-Sampling and SMOTE. *Under-Sampling*: It reduces the number of instances of the class that is higher in number to match the number of instances of the class lower in number. *Over-Sampling*: It increases the number of instances of the class that is lower in number to match the number of instances of the class higher in number. *SMOTE*: It is a technique which works by increasing the number of instances of the smaller class and decreasing the number of instances of the higher class till the dataset attains a balance.

3) Scaling: After the sampling process, it was decided to scale and test the dataset to remove the effect of the difference between the magnitude of values of different features. However, upon testing for the explanations, sampling rendered different features to be of less advantage since now their values were more or less similar to each other and as a result the explanations did not properly account for the effect of the features individually. Thus, the idea of scaling the dataset was dropped and the explanations were provided on the unscaled dataset itself. After the scaling process, the feature transaction_type that had textual values was transformed into numeric values to maintain consistency across the dataset. For this, the number of occurrences of each transation_type throughout the dataset was counted and the textual values were replaced with their count values. Since the count values were large integers, they were indirectly affecting the explanations because of their large magnitudes. To resolve this issue, the large numeric values were replaced with simple integers ranging from 1 to 5.

After obtaining a balanced dataset, we performed the classification on Logistic Regression. On training the model and testing it on the test data, it was concluded that the SMOTE technique provided the best results among all the three techniques.

B. Analysis

After all the data transformation steps mentioned above, the model is trained on 4 different Machine Learning algorithms and 1 Deep Neural Network. The aim here is to understand the performance of various algorithms on the dataset and compare their results with each other. The 4 different Machine Learning models used are Logistic Regression, Linear SVC, K-Nearest Neighbors and Decision Tree Classifier.
The logistic regression model is built with the penalty as L2. The value of inverse regularization strength is 1. The tolerance for stopping criteria is set to 1e-4. The solver used for the optimization is lbfgs and maximum number of iterations taken for the solver to converge is 100.

The Linear SVC model is built with the penalty as L2. The loss function considered is squared hinge. The tolerance for stopping criteria is set to 1e-4. The value of inverse regularization strength is 1. Maximum number of iterations taken for the solver to converge is 1000.

The K-Nearest Neighbors model is built with number of neighbors as 5. Weights are used as uniform distribution. The power parameter for the Minkowski metric is set to 2. The Leaf size passed to KDTree is set to 30.

Model	Precision-Recall AUC	ROC AUC	Accuracy
Logistic Regression	0.9482	0.8873	0.8554
SVM	0.9694	0.9510	0.8785
KNN	0.9984	0.9880	0.9960
Decision Tree	0.9987	0.9954	**0.9984**
Deep Neural Network	**0.9996**	**0.9996**	0.9923

Table 1. Credit Card Fraud Detection evaluation metrics on various mode

The Decision Tree model is built with the Gini as the criteria for the split of the nodes. The minimum number of leaf samples is set to 1. The minimum number of samples required to split an internal node is set to 2.

The Deep Neural Architecture used consists of 3 dense layers with the last one as the output layer. The first dense layer consists of 60 neurons along with ReLU as the activation function and HeNormal as the kernel initializer. The second dense layer consists of 30 neurons along with ReLU as the activation function and HeNormal as the kernel initializer. Finally, the last layer is a single neuron node with the activation as sigmoid to give us the output probabilities. The optimizer used is Adam with the default learning rate. The loss metric used in this case is the binary cross-entropy. The model is trained for 40 epochs and results are evaluated on the test set.

Looking at the performance of different models in table 1 we can see that Deep Neural Network performs much better as compared to other models. The highest score in each category is highlighted in bold. Although, Decision Tree has a better accuracy, DNN is performing better in terms of ROC-AUC curve and Precision-Recall Curve. A false negative occurs when a fraudulent transaction is not detected as fraud and passes via the fraud detection system. Heavy losses are incurred if the fraud cases are not caught. Since, in the case of fraud detection false negatives need to be avoided as much as possible it is recommended to consider Precision-Recall curve in terms of performance evaluation.

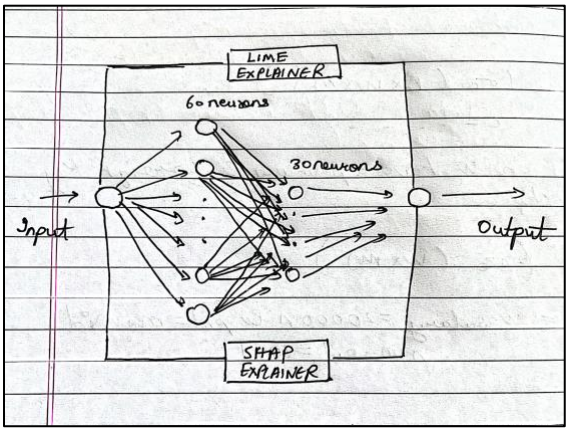

Figure 1. Structure of our method

The proposed system is made up of two components: (1) fraud detection model, and (2) model explainers. The former is the DNN architecture as explained above since it performs better as compared to other models. The model explainers give users a clear picture of how input characteristics influence network output. LIME was chosen to examine a single transaction of interest and SHAP was chosen to understand the single transaction of interest along with the overall global explanation in terms of feature importance. Figure 1 depicts a high-level picture of the planned structure.

LIME is a model-agnostic technique that just requires a classifier or regressor as the target model. As a result, the DNN classification explainers can be directly trained using samples taken from the dataset's distribution. LIME works as follows:

1) It creates a new dataset around the observations by sampling from the distribution learned from the training data
2) Then it calculates the distance between original observation and above sampled observations
3) Uses a model to predict probabilities on new points
4) Choose top k features which best describe the complex model outcome from the sampled data
5) Finally, a linear model is fit on the data which consists of k dimensions weighted by similarity
6) These weights are used as explanation of the result under consideration

Instead of giving a global model interpretation, LIME advocates focusing on understanding locally. We can zoom in on a piece of data in a model to see which features influenced the model to reach a particular result. It gives the contribution of each feature towards the results. LIME is unconcerned about the machine learning/deep learning model you're using. It will consider it as a black box and concentrate on deciphering the local outcome.

SHAP is also a model-agnostic technique that just requires a classifier or regressor as the target model. As a result, the DNN classification explainers can be directly trained using samples taken from the dataset's distribution. SHAP's purpose is to compute the contribution of each feature to the prediction of an instance x to explain it. Shapley values are computed using the SHAP explanation technique, which is based on coalitional game theory. A

data instance's feature values operate as coalition members. Shapley values provide us how to allocate the prediction among the features in a fair manner.

The Shapley value explanation is depicted as an additive feature attribution approach, a linear model, which is one of SHAP's innovations. LIME and Shapley values are linked in this perspective. SHAP provides both local explanations and global explanations. In local interpretability each observation gets its own set of SHAP values and in global interpretability collective SHAP values can be used to illustrate how much each predictor contributes to the target variable, either positively or negatively.

Results

After choosing the best technique for the data sampling, the dataset was used to train the deep learning model and the final classification decisions were explained by using LIME and SHAP.

Below are some examples of different transactions that were explained:

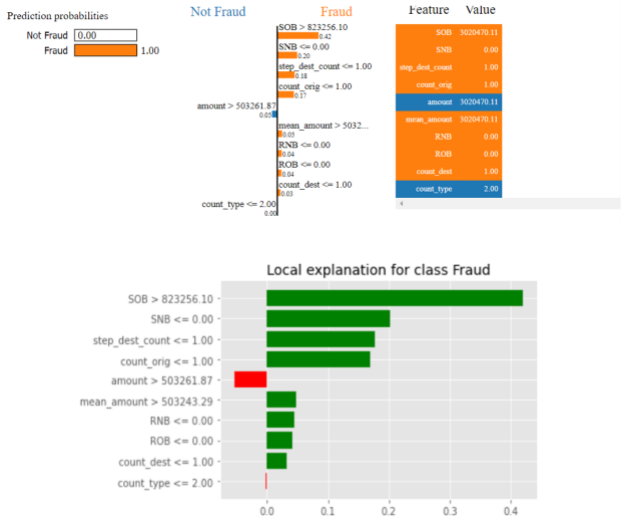

Figure 2. LIME explanations for fraud class

The first instance tested belongs to fraud category. As shown in figure 2, the most important feature turns out to be the Sender's old balance, followed by Sender's new balance. Analyzing the LIME chart, it is evident that after the transaction, Sender's whole account was left empty, and all the account balance was transferred. However, the receiver's new balance remained Zero, which shows that probably, the money did not reach the intended receiver. Also, the whole balance was transferred within the same time step, which is the third most important feature affecting the classification decision. The amount of the transaction, which was greater than 503261, did point towards a possibility of the transaction being Not Fraud, however the other features were prominently pointing towards it being Fraud. Based on these inputs, the transaction was finally classified as a fraud transaction. Similar effect of the features is also shown in the local explanation graph.

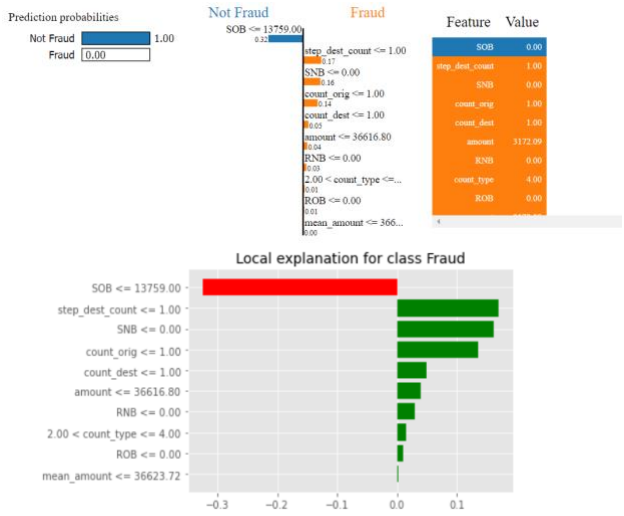

Figure 3. LIME explanations for not fraud class

In the figure 3 above, the transaction has been classified as a non-Fraud transaction. In this, the amount to be transferred was 3172 and the sender's old balance is Zero. Also, the receiver's new and old balance is also Zero. This suggests that perhaps the transaction got cancelled due to insufficient funds because of which the receiver did not receive any money. Since it seems to be a genuine case of a failed transaction, it is classified as Not Fraud. This becomes evident on analyzing the effects of most important features of the transaction, i.e., Sender's old balance which affects the decision by the largest magnitude. The other features point towards a possibility of the transaction being fraud, but their effect is less compared to that of the sender's old balance which results in the transaction being classified as Not Fraud.

SHAP uses the magnitude of a feature value to determine whether it affects the classification positively or negatively. It shows the direction in which a particular feature drives the overall decision. The feature which leads to Not Fraud is shown in Blue and the feature which leads towards Fraud are shown in Pink.

The transaction in figure 4 falls under a fraud category. According to the graph, Receiver's old and new balances are the top two contributing factors towards the Fraud classification followed by amount and Sender's new balance where the new balance increased significantly after the transaction. Also, Sender's new balance became Zero indicating towards a possible Fraud. The other features don't have a significant impact towards the classification decision. Although Sender's old balance contributes towards the transaction being Not Fraud, the cumulative

effect of all the features in Pink eventually lead to the transaction being classified as a fraud transaction.

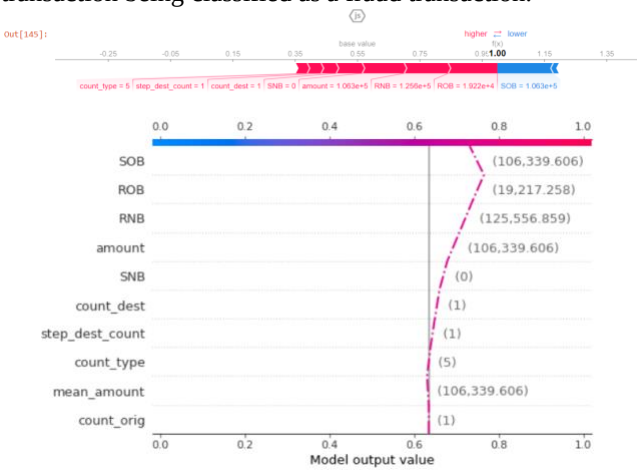

Figure 4. SHAP explanations for fraud class

In the figure 4 summary plot, blue color refers to lower magnitude of the feature values, whereas pink color indicates higher values. Towards the left side of the mean value (middle line) is the negative impact of a feature, i.e., it will drive the decision towards a transaction being Not Fraud, and the right side indicates a positive impact. Looking at the graph above, when Sender Old Balance is low (blue color), it has a negative impact on the classification (towards left of the middle line), whereas high value has a positive impact. On the other hand, Sender's New balance follows an opposite pattern, where a higher value has a negative impact, and a lower value has a positive impact. This can also be verified intuitively. If after a transaction Sender's new balance gets very low, it indicates towards a possibility of a fraud transaction. A similar pattern is shown by Receiver's old and new balances.

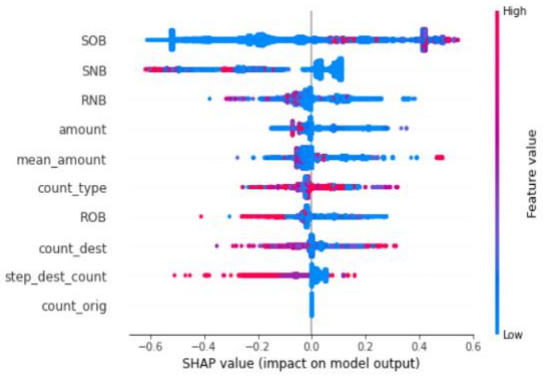

Figure 5. Global Explanation using SHAP summary plot

A higher value of Receiver's new balance has a negative impact, whereas a lower value has a positive impact, since that could indicate that the intended receiver of the transaction didn't receive the funds, thus indicating towards a possible fraud. The amount of the transaction also follows a specific pattern throughout the dataset. According to the summary plot, the chances of a transaction being a fraud increase when the amount value is low as compared to when the amount is high. Other features like the step_dest_count, which counts the number of times a receiver was transferred money in each time step, and count_dest and count_orig contribute towards a Non fraud transaction when their values are high.

Having observed the different explanation strategies followed by LIME and SHAP, it can be well established that LIME provides better Local explanations as compared to those provided by SHAP. LIME provides an in-depth analysis of the value ranges of individual features which contribute to the classification decision, thus enabling us to deduce better explanations as to why a particular transaction instance was classified as either Fraud or Not Fraud.
On the other hand, SHAP provides a better overall analysis of the dataset through its summary plot which enables us to identify the most important features contributing towards the classification decisions. Thus, it can be established that SHAP is better suited for providing global explanations as compared to LIME.

Conclusion

This work proposes to understand a black box model using local and global explanations along with a reasonable amount of transparency for a credit card fraud detection system. The goals have been achieved by applying LIME and SHAP white box explainers to a Deep Neural Network architecture.

Following are the strengths, limitations, and future work of our proposed approach:

A. Strengths

1) Model Independence: One of the obvious strengths of using LIME and SHAP for explanation purposes is that it can be used to explain any kind of model, be it an already interpretable white box model like logistic regression, decision tree etc. or a black box model like a deep neural network.

2) Granularity: LIME was able to provide explanations with a fine granularity. Along with identifying the most important features impacting the classification decisions, it was also able to provide a specific range of the feature values in which they either affected the decisions positively or negatively.

3) Easy to understand explanations: Both LIME and SHAP provide explanations that could be easily interpreted and inferred by the human user, thus increasing the level of trust between the model and the end user.

B. Limitations

1) It is difficult to use LIME with one hot encoded data. Since each datapoint is manipulated and perturbed in order to create an approximate model, manipulating one hot encoded data lead to generation of meaningless datapoints which in turn lead to unexpected and unsatisfying explanations.

2) SHAP explanations depend on calculating Shapley values for all data points. This process is computationally expensive, especially in the case of large datasets. To handle this problem, we have to rely on sub sampling the dataset into smaller datasets, which may affect the accuracy of the explanations.

3) LIME works better with models that predict the probabilities of decisions of classification problems. Since models like SVM do not provide output probabilities, using LIME with them may result in some bias in the explanations.

C. Future Work

1) Implementation of BRCG: Boolean Rule Column Generation explainer implements a directly interpretable supervised learning method for binary classification that learns a Boolean rule in disjunctive normal form (DNF) or conjunctive normal form (CNF) using column generation (CG).

2) Larger datasets can be used for future experiments, which would hopefully provide an even better performance and accuracy.

3) Different Black Box models: Apart from Deep Neural Network, other black box models like 1-D CNN can be used to provide better accuracy.

References

[1] Adewumi, A.O., Akinyelu, A.A. A survey of machine-learning and nature-inspired based credit card fraud detection techniques. *Int J Syst Assur Eng Manag* **8,** 937–953 (2017).

[2] Y. G. S ̧ahin, E. Duman, "Detecting Credit Card Fraud by Decision Trees and Support Vector Machines," IMECS 2011 - International MultiConference of Engineers and Computer Scientists 2011, vol. 1, pp. 442-447, Mar. 2011.

[3] M. Zareapoor, P. Shamsolmoali, "Application of credit card fraud detection: Based on bagging ensemble classifier," Procedia computer science, vol. 48, pp. 679-685, 2015.

[4] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim and A. K. Nandi, "Credit Card Fraud Detection Using AdaBoost and Majority Voting," IEEE Access, vol. 6, pp. 14277-14284, 2018.

[5] J. O. Awoyemi, A. O. Adetunmbi and S. A. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis," 2017 International Conference on Computing Networking and Informatics (ICCNI), 2017, pp. 1-9, doi: 10.1109/ICCNI.2017.8123782.

[6] Kajal et al., "Credit Card Fraud Detection using Imbalance Resampling Method with Feature Selection", International Journal of Advanced Trends in Computer Science and Engineering, 10(3), May - June 2021, 2061 – 2071

[7] H. Ba, "Improving Detection of Credit Card Fraudulent Transactions using Generative Adversarial Networks," arXiv:1907.03355 [cs.LG], Jul. 2019.

[8] D.Cheng,S.Xiang,C.Shang,Y.Zhang,F.YangandL.Zhang"Spatio-temporal attention-based neural network for credit card fraud detection," In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 01, pp. 362-369, Apr. 2020.

[9] Y. Ki, J. W. Yoon "Pd-fds: Purchase density based online credit card fraud detection system," In KDD 2017 Workshop on Anomaly Detection in Finance, PMLR, pp. 76-84, Jan. 2018.

[10] P. Zheng, S. Yuan, X. Wu, J. Li, and A. Lu, "One-Class Adversarial Nets for Fraud Detection", AAAI, vol. 33, no. 01, pp. 1286-1293, Jul. 2019.

[11] A. Barredo Arrieta et al., "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities, and challenges toward responsible AI," Information Fusion, vol. 58, pp. 82–115, 2020.

[12] Wu, Tungyu, and Youting Wang. "Locally Interpretable One-Class Anomaly Detection for Credit Card Fraud Detection." arXiv preprint arXiv:2108.02501 (2021).

Contribution of Group Members:

Abhishek: Worked on Data Cleaning, ML models and LIME explainer. Results, Dataset and Conclusion from the paper.

Ajinkya: Worked on Data Cleaning, DNN model and SHAP explainer Abstract, Introduction, Methodology from the paper.