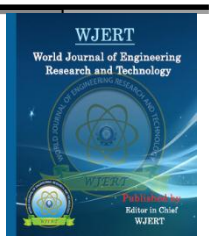
wjert, 2022, Vol. 8, Issue 5, 155-169. Review Article ISSN 2454-695X

World Journal of Engineering Research and Technology

WJERT

www.wjert.org

SJIF Impact Factor: 5.924



A COMPARATIVE ANALYSIS OF CREDIT CARD FRAUD DETECTION USING DIFFERENT MACHINE LEARNING TECHNIQUES WITH LIME–A HUMAN EXPLAINABLE AI

Muntasir Hasan Kanchan and Muhammad Masud Tarek*

Department of Computer Science and Engineering State University of Bangladesh Dhaka, Bangladesh.

Article Received on 14/03/2022 **Article Revised on 04/04/2022** **Article Accepted on 24/04/2022**

***Corresponding Author**
Muhammad Masud Tarek
Department of Computer Science and Engineering State University of Bangladesh Dhaka, Bangladesh.

ABSTRACT

Detecting fraud in credit card purchases is perhaps one of the better testbeds for computational intelligence algorithms. Indeed, there are a variety of significant problems in this issue: definition drift (evolving consumer preferences and shifting tactics over time), class imbalance (actual transactions far beyond fraud), and latency verification (only a limited number of transactions are tracked in good time by the investigators). Accurate identification and avoidance of fraud are essential to protect financial institutions and individuals. The credit card fraud monitoring system was used to track fraudulent practices that was implemented. In this work, we use human explainable AI technique Local Interpretable Model-agnostic Explanations (LIME) to model the sequence of transactions in managing credit card transactions and show how it can be used to detect fraud. Decision Tree, K-Nearest Neighbors (KNN), Random Forest, Support Vector Machine (SVM), and XGBoost, with three performance measures (accuracy, f1-score, and confusion matrix) required to demonstrate the classification prediction's effectiveness. As a result, it is vital to evaluate if a model generates a specific prediction. We eventually train an interpretable model called LIME for the sample based on its neighbors, this cardholder's activity patterns, and the associated cross features. Compared to the five classifiers, KNN gives better results from accuracy and f1-score to identify fraud.

KEYWORDS: Credit card fraud, e-commerce security, fraudulent activities, machine learning, LIME.

INTRODUCTION

Increase of e-commerce activities and online payments credit card fraud is also a rising concern in the modern era. With the evolved e-banking system, any flaws in these operations have escalated fraudulent transactions. Fraud can be avoided in the first place by taking preventative and detection measures. Fraudsters are thwarted by prevention, which acts as a security barrier. Sailusha et al. argued that, despite the use of various data mining techniques, the results are not very effective in detecting fraud cases (Sailusha et al., 2020). The only way to combat these dangers is to use successful fraud detection algorithms. Detection system can alert as soon as an irregular transaction occurs. Machine Learning techniques are being utilized to develop mathematical algorithms that can classify non-legitimate transactions based on their amount and length. Internal and external fraud are two types of card fraud. Internal fraud occurs when a bank employee poses as a customer with a fraudulent identity. Outer fraud is defined as the use of a stolen credit card by criminals to make money.

The technique of distinguishing transactions for credit card purposes is known as fraud detection. According to Maes et al. there are two types of transactions: legitimate and fraudulent (Maes et al., 2002). Typical fraud detection systems include an academic degree system that is automated and a manual methodology. The automated system is based on fraud detection rules. All new transactions are analyzed and a false score is assigned. Stolfo et al. developed manual cost based modeling which can be used for intrusion detection (Stolfo et al., 2000). Credit card fraud detection uses a dataset to classify fraud. Due to databases that are deeply imbalanced and distorted the judgment is extremely challenging. The challenging job is to collect datasets. Financial datasets are not simply skewed, but they're also not always complete. Consumers whose data is in the hands of dataset suppliers are conscious of their privacy and security concerns. Thus, in a number of datasets, only one or two alphabetical attributes are used in numerical tables. Another issue that arises frequently during the credit card detection procedure is that non-legal records are complicated, making fraudulent transactions appear to be legitimate. Around the same time, it is difficult to locate datasets for credit card purchases. Not all of these methods provide real-time monitoring, but they increase the rate of false alarms. However, the client profile is seldom used.

The major goal of our work is to propose a novel model for detecting credit card fraud, based on the LIME (Local Interpretable Model-agnostic Explanations) method, which describes the accuracy that we have obtained and tailors the proposal to real-time transactions and reduces false alarm rates. The Paradigm Description in the area of Artificial Intelligence is very recent and difficult. Understanding a model will give one a deeper view into how the projections come in. In the world of e-commerce, this intuition tends to recognize the key points of fraud. This technique also aims to improve current algorithms for greater accuracy in the future.

We presented the LIME model, which assesses many machine learning models and determines whether the explanations can be utilized to choose a model, emulating the circumstance where a person must discern between two opposing fraud cards. The purpose of this experiment is to see if a customer can use the validation collection's descriptions of fraud incidents to categorize a better classifier.

The major issue with the credit card fraud identification system is to detect fraud in a broad data collection where the rate of legitimate transactions is more important than the rate of fraud, which could be negligible.

To overcome this problem, we suggested an approach where our findings are validated by LIME which proves the explainable result that the fraudulent is identified correctly. After classification via different machine learning methods, LIME helps to assess the correctness of the detected result by comparing the output with visual observation. Local fidelity does not equal global fidelity in our function, as in the case of LIME: qualities that are important globally cannot be crucial locally, and vice versa. Because of this, only a few variables can be directly linked to local (individual) prediction, whereas the model has hundreds of variables globally.

LITERATURE REVIEW

The most up-to-date program to detect credit card fraud employing a variety of research approaches and fraud detection approaches, with a special focus on neural networks, data processing, and data mining distribution. There are a variety of alternative ways to detect credit card fraud. Following the completion of the literature review, it is possible to infer that there are alternative approaches for identifying credit card fraud in Machine Learning. Saputra and Suharjito used machine learning in e-commerce (Saputra and Suharjito, 2019), and Roy et al. worked on Deep Learning algorithms for detecting fraud payment transactions (Roy et al.,

2018). Fang et al. proposed Light Gradient Boosting Machine model claiming it gives better result than random forest in detecting credit card theft (Fang et al., 2019).

Jain et al. compared among SVM, ANN, Bayesian networks, Hidden Markov Model, KNN, Fuzzy Logic approach, and Decision Trees methods for detecting frauds (Jain et al., 2019). KNN, Decision Trees, and SVM are found to have a medium level of accuracy. The Fuzzy and Logistic Regression had the lowest precision of any other algorithms. Logistic regression, SVM, and decision trees all provide a high level of detection with an intermediate rate. ANN and Naive Bayesian Networks performed in all cases but they are expensive to practice. Again, they do not give consistent outcomes. With one type of dataset, they produce better outcomes, but with another type of dataset, they produce worse results. For limited datasets KNN and SVM produce impressive results. Fuzzy logic systems have decent consistency for raw and non-sampled data.

Ghosh and Reily implemented fraud detection using neural network (Ghosh and Reily, 1994). They trained the neural network on different types of frauds like lost card, mail-order fraud, missing cards, stolen cards etc. and set up a surveillance system in a bank. Their system produced significantly lower false positive detection. Parallel granular neural networks (PGNNs) were used by Syeda et al. on 24-CPUs for faster extraction of card fraudulent (Syeda et al., 2002). A large number of Visa Card transactions were used as training data for preprocess before applying for fraud detection. Aleskerov et al. introduced CARDWATCH with a GUI to a variety of e-commerce sites for card fraud detection. It is an auto-associative neural network learning model with very successful detection rate (Aleskerov et al., 1997). Because of biasness of typical training data set, misdetection rate is normally very high. So, Kim et al. took fraud density of real-world transaction as confidence value (Kim et al., 2002). They compared the effectiveness of their result on real data. Synthetic minority oversampling technique (SMOTE) is used for class imbalance problems while Whale optimization algorithm (WOA) is used for solving complex optimization problems. Sahayasakila et al. combined both techniques to optimized detection of both fraud and non-fraud transactions (Sahayasakila et al., 2019). Khare and Sait took significantly skewed dataset, preprocessed it and applied different machine learning techniques (Khare and Sait, 2018). They made a comparison and concluded that Random Forest outperforms other methods in detecting fraud cases.

Proposed Methodology

The key goal of this research is to identify fraud and non-fraud credit card purchases by classifying data using Decision Tree, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest, and XGBoost algorithms. The proposed model shown in Fig. 1 needs data prepossessing (model training), testing data and data classification model execution to detect fraud. Then LIME comes in action to evaluate the correctness of the detected result via visual observation.

The fraud dataset for the Kaggle credit card (link: https://www.kaggle.com/mlg-ulb/creditcardfraud) is preprocessed to use in this model. To create the model, we have to separate the dataset as training data and testing data. We split the data into 80:20 where we train 80% of data and test 20% of the data using 'split' method in Python. After the data is divided, we also quantify the percentage of fraud cases in the total transactions reported on the train data. We build a training set that helps the algorithms to attain those attributes (Table I). We apply the above-mentioned classification algorithms on the training dataset and calculate accuracy, f1-score and confusion matrix of the models.

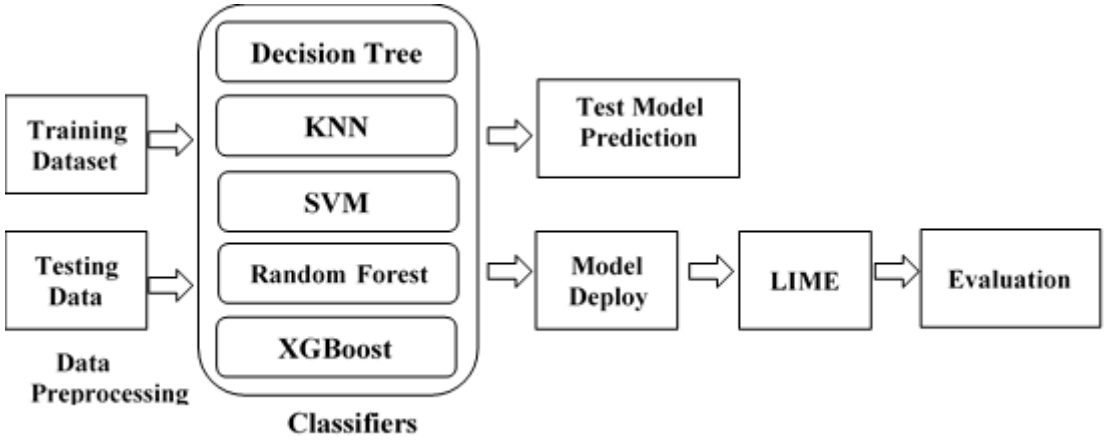


Fig 1: Flow chart of the proposed methodology.

Table I: Case Count in Data.

Category	Case Count
Total Number of Cases	284807
Number of Non-Fraud Cases	284315
Number of Fraud Cases	492
Percentage of Fraud Cases	0.1727%

Dataset Description

The data provided by Kaggle for academic purpose are used in this research. The dataset included here are the two days transactions done by the European card holders. It has information of 284,807 purchases of which 492 (0.1727%) are scams. Hence, the dataset is highly imbalanced. Due to privacy concern, instead of actual features, 28 feature variables are provided as the principal components obtained with Principal Component Analysis (PCA). Only 'Time' and 'Amount' have the actual values. The feature 'Time' means the number of seconds between first transaction and purchase transaction. Another relevant 'Class' is a Boolean feature indicating 'Fraud' (1) or 'Non-fraud' (0) transactions as shown in Table II. Different statistical values of the given dataset are shown in Table III.

Table II: Features with potential values.

Variable	Full-form	Variable type	Potential value
Time	Differences between current transaction and the first transaction in seconds	Independent	84, 236
V1-V28	Values obtained with PCA of aqctual data to protect user privacy	Independent	0.2376089398, 1.5487178465
Amount	Transaction amount	Independent	378.66, 231.71
Class	Fraudulent transactions, Non-fraudulent transactions	Dependent	Non-fraud (0), Fraud (1)

We designed five different classification models: Decision Tree, KNN, Random Forest, SVM, and XGBoost using scikit-learn bundle and XGBoost kit in Python. While there are additional models that can be used to address classification problems, these are the most prevalent ones. After classification design, the accuracy and f1-score are calculated. Then, in the case of concept implementation, we use LIME to describe the outcome we have obtained. Finally, the payment card theft purchases with LIME were analyzed more specifically.

Table III: Case Amount Statistics in Data.

Case Criteria	Count	Mean	Standard Deviation	Max
Non Fraud Case	284315.00	88.29	250.10	25691.16
Fraud Case	492.00	122.21	256.68	2125.87

Implementation of Different Classifier Algorithm of Machine Learning

Decision Tree Implementation

Decision tree is a system consists of root and nodes where an internal node indicates an attribute demand, each branch is indicated by the trial outcome and leaf node is indicated by the class as mentioned by Gaikwad et al. (Gaikwad et al., 2014). Entropy of decision tree tests

the impurity of the result class in a subset with the attributes of Ps in any dataset as shown in the equation 1:

$$H(p_1, p_2, \ldots, p_s) = \sum_{i=1}^{s} \left(p_i \log \left(\frac{1}{p_i} \right) \right) \quad (1)$$

Information gain is the difference between entire dataset entropy and the splitting attribute entropy as shown in the equation 2:

$$Gain(D, S) = H(D) - \sum_{t=1}^{s} p(D_i) H(D_i) \quad (2)$$

In Python, the ‘DecisionTreeClassifier' technique is used to develop the model. The maximum depth of the tree is set 4 and the 'criterion' is set as 'entropy' that is the closest to ‘max_depth,' but specifies when to prevent splitting the tree. Finally, we have entered, processed and stored the expected values in the ‘tree yhat' variable.

- ***KNN Implementation***

According to Malini and Pushpa, controlled learning methodologies have proven that KNN performs very well in credit card fraud detection programs (Malini and Pushpa, 2017). In KNN, we used a Python library named 'KNeighborsClassifier' with 'n neighbors' set to '5.' The value could be chosen arbitrarily. Finally, data are fitted and the expected values are stored in the 'knn yhat' vector.

- ***SVM Implementation***

SVM is a classifier that combines kernel techniques and maximal boundary classifiers. Demla and Aggarwal used SVM fraud detection which is focused on Vapnik's mathematical reasoning fundamental concept (Demla and Aggarwal, 2016). It has been successfully introduced to multiple real-world concerns such as face identification, intrusion detection, handwriting recognition, knowledge retrieval. We used the 'SVC' algorithm and default ‘rbf’ kernel to design the SVM model. We stored the expected values in the ‘svm yhat'.

- ***Random Forest Implementation***

Kumar et al. showed that Random Forest Algorithm (RFA) often offers greater performances compare to many other schemes and is the most widely used algorithm for decision making (Kumar et al., 2019). Here, we used the ‘RandomForestClassifier' algorithm to create the

Random Forest Model, and we set the ‘max_depth' to 4 just like we did with the decision tree model. Finally, stored the expected values in the ‘rf yhat'.

- ***XGBoost Implementation***

The XGBoost is a learning classifier with two positive aspects of individual learning units which render engineering feature redundant. In XGBoost, gradiant gi and hessian hi independently construct a booster tree to handle class imbalance (Meng et al., 2020). According to Ribeiro et al., the objective equation of regularization for the training features and the goal, the tree set with the number of trees K is given as (Ribeiro et al., 2016):

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \qquad f_k \in \mathcal{F} \tag{3}$$

Where f is the practical field and F is the collection of potential classification and regression trees (CART). Optimized regularized target equation is,

$$O(\theta) = \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{4}$$

Considering the additive tree boosting preparation, the optimized objective function is defined as:

$$O = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i(t)\right) + \sum_{i=1}^{t} \Omega(f_i) \tag{5}$$

Here, we used the ‘XGBClassifier' algorithm to create the XGBoost Model, and we set the ‘max_depth' to 4 just like we did with the decision tree model. Finally, stored the expected values in the ‘xgb yhat'.

LIME Implementation

Ribeiro et. al. introduced LIME (Local Interpretable Model-agnostic Explanations), which is an approach that can describe each classifier's predictions by approximating them to a locally interpretable model (Ribeiro et. al., 2016). It is a consistent model agnostic explicator and a system for choosing a representative collection of interpretations (SP-LIME) to ensure that the model acts consistently when replicating human reasoning.

The representative collection will have an intuitive global interpretation of the model. Let the model is $f : R^d \rightarrow R$, where $f(x)$ is the classification probability. Letting $\Omega(g)$ as the depth of a tree or a non-zero small number. LIME describes the forecast in such a way that even non-experts can compare and develop an untrustworthy model by function engineering as given below:

$$\xi(x) = \text{argmin\_(g} \in \text{G)}\ \mathcal{L}(f, g, \pi_x) + \Omega(g) \quad (6)$$

Where, $L(f, g, \pi_x)$ is a measurement of how untrustworthy g in predicting f in the locality defined by π_x. In order to ensure both interpretability and local fidelity, $L(f, g, \pi_x)$ should be minimized and $\Omega(g)$ should be a low value. We train all five classifiers using two important features (time and amount). For each prediction on the test set, using LIME, explanation is generated. The explanation is used to verify whether the model used for prediction is good to use with the given features.

Experimental Result Analysis

- ***Performance Evaluation***

To evaluate the implemented models, we use the evaluation metrics supplied by the scikit-learn package of Python. The accuracy score, F-score and the Confusion-metrix are the main assessment components for this purpose. Finally, we apply LIME in model deployment for validating the predicted outcomes.

- ***Accuracy-Score***

The Accuracy-score or precision score is calculated by dividing the number of true predictions (both positive and negative) by the total number of predictions generated by the model as stated in the equation 7. In Python we use 'accuracy_score' method for the calculation.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

- ***F1-Score***

F1 or F-score score is the harmonic mean between precision and recall. It is calculated using the given equation 8. The F1 score can be readily determined in Python using the scikit-learn package's 'f1_score' method.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

The Table IV depicts the results of our five models that predict the credit card fraud detection. Here KNN gives the highest percentage of detection among the others on basis of accuracy and f1-score.

Table IV: Results of Five Classifiers to detct Credit Card Fraud.

Model	Accuracy-Score	F1- Score
Decision Tree	99.93%	81.05%
KNN	99.95%	85.71%
SVM	99.93%	77.71%
Random Forest	99.92%	77.27%
XGBoost	99.94%	84.21%

- ***Confusion Matrix***

Confusion or Error Matrix is a two-by-two matrix that devides all the outcome into four parts (TP, FP, FN, TN). We fitted the all outcomes in the confusion matrix after implementing five classifier algorithms on the given credit card dataset. The figure 2, 3, 4, 5, 6 depict the confusion matrix as heatmaps for five classifier algorithms. TPR, TNR, FPR, and FNR are calculated to extract sensitivity, specificities, coherence and error rate from the matrix.

For example, the KNN model's confusion matrix in Fig 5, the first row contains total 56861 non-fraud transactions of which 56854 cases were truly predicted as non-fraud transactions (labelled as 0) and 7 cases were falsely predicted as fraud transactions (labelled as 1). Next row contains total 101 truly fraud transactions of which 81 cases were rightly identified as fraud (labelled as 1) where is 20 fraudulent cases could not be detected rightly (labelled as 0). Comparing with the predictions of other classifiers we found KNN model gave slightly better prediction accuracy for the given training and test data set.

- ***LIME Evaluation***

Finally we tried to validate the predictions of classifier algorithms using LIME (Ribeiro et al., 2016). We first selected V8 and V16 features arbitrarily to be “untrustworthy”. Then we perform a black box testing by removing those features and checking whether the predictions are changed or not. If the predictions are changed, the features are labelled as really “untrustworthy” otherwise as “trustworthy”. In the Figure 7 and 8, we see the representation of V8 and V16 feature evaluation using LIME. Here orange color code indicates fraud case and blue color indicates non-fraud case. In Figure 8, the feature value of V16 (-0.26) indicates fraudulent case which is shown by orange color and the feature value of V8 (0.63) indicates non-fraud case which is shown by blue color by LIME.

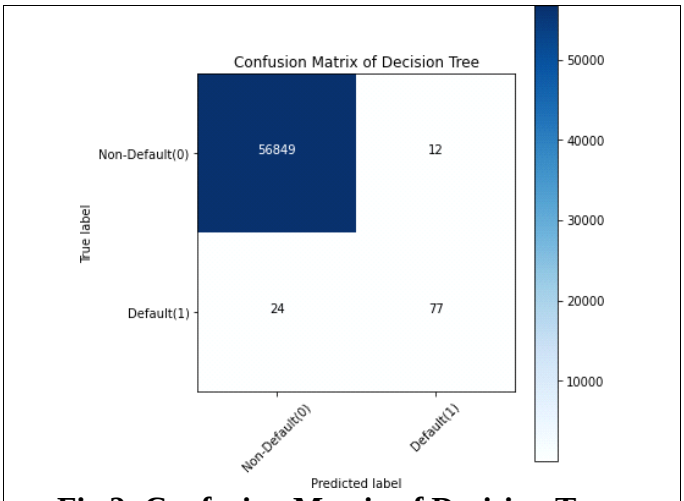


Fig 2: Confusion Matrix of Decision Tree.

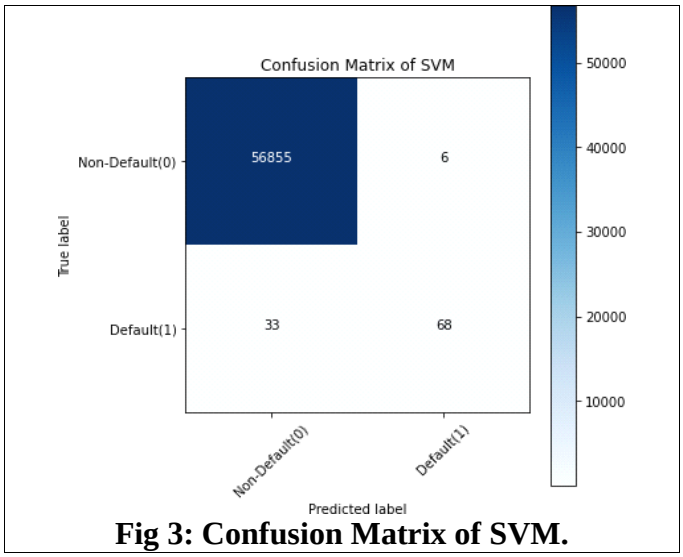


Fig 3: Confusion Matrix of SVM.

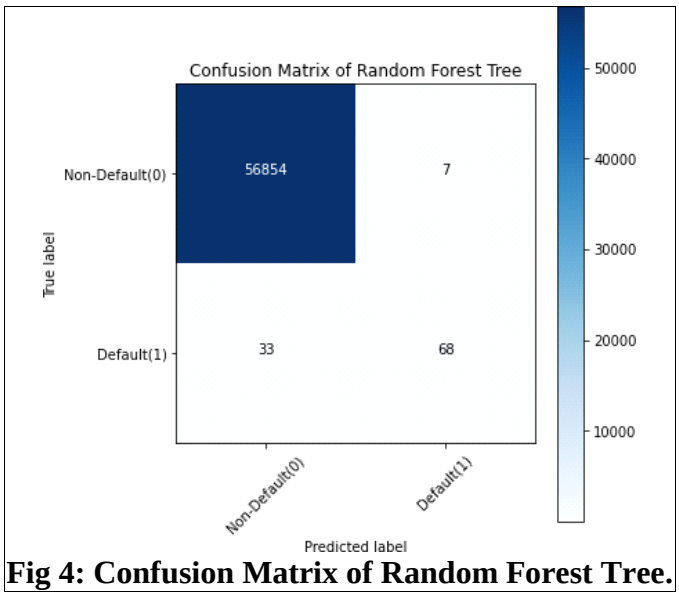


Fig 4: Confusion Matrix of Random Forest Tree.

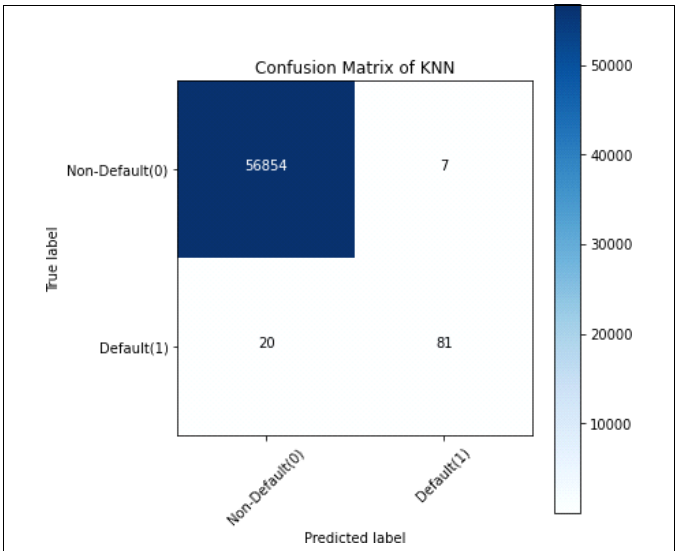


Fig 5: Confusion Matrix of KNN.

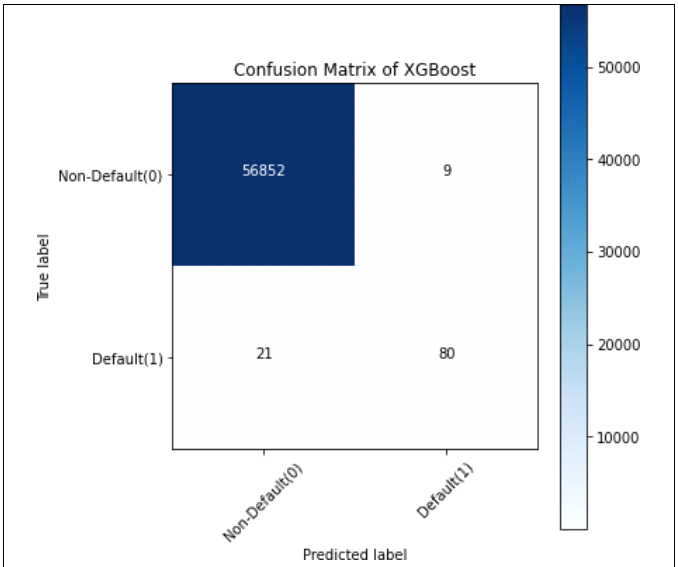
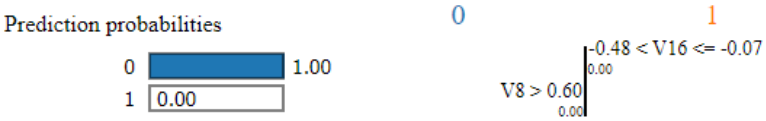
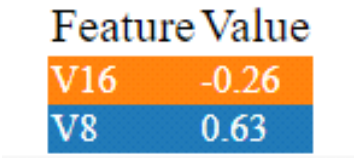


Fig 6: Confusion Matrix of XGBoost.

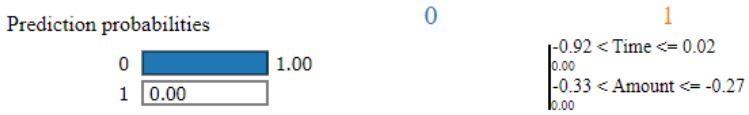


Prediction of V8 and V16 Feature

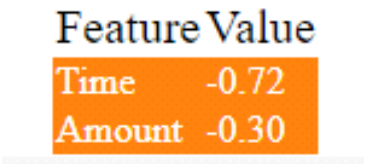


V8 and V16 Feature

While we choose the features as untrustworthy arbitrarily, LIME is useful in determining confidence in individual forecasts. We test the loyalty of descriptions on classifiers which can be understood on their own. Thus, we consider the Time and Amount feature (Figure 9 and 10), LIME can make these feature human explainable and the feature values indicate whether the detected credit card fraudulent with five classifiers tend to be accurate.



Prediction of Time and Amount Feature



Time and Amount Feature

CONCLUSION

The purpose of this work is to offer a concept for detecting credit card fraud. This paradigm provides a major contribution relative to the classic model suggested in the literature as our primary emphasis is on interpreting the whole classification process for a clearer intuition about how the model really operates. Five types of machine learning models have been used to test their output in a data set containing real world transaction data. We found KNN has a 99.95% accuracy rate and f1 score of 85.71% while XGBoost has an accuracy rate of 99.94% and f1 score of 84.21%. We choose them not only for their accuracy and f1 score, but also for their market relevance. We want to demonstrate how classical methods can be used to detect fraudulent transactions along with the extension of deep learning techniques. Both KNN and XGBoost are precise and economical. In credit card fraud detection, feature reduction helps us to achieve remarkable outcomes but we may have to concentrate on high recall value. We can incorporate LIME with five typical machine learning models to interpret the prediction in human explainable way. Next we like to focus on addressing the detection problem with a broad variety of functions and will put it into line with the state-of-the-art SHAP network in our future work.

REFERENCES

1. Aleskerov E, Freisleben B and Rao R B "Cardwatch: A Neural Network Based Database Mining System for Credit Card Fraud Detection", Proc. IEEE/IAFE: Computational Intelligence for Financial Eng., 1997; 220-226.
2. Demla N and Aggarwal A "Credit card fraud detection using svm and reduction of false alarms", International Journal of Innovations in Engineering and Technology (IJIET), 2016; 7(2). ISSN: 2319-1058, 176-182.
3. Fang Y, Zhang Y and Huang C, "Credit Card Fraud Detection Based on Machine Learning", Computers, Materials & Continua CMC, 2019; 61(1): 185-195.
4. Gaikwad J R, Deshmane A B, Somavanshi H V, Patil S V and Badgujar R A "Credit Card Fraud Detection using Decision Tree Induction Algorithm", International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, 2014; 4(6).
5. Ghosh S and Reilly D L "Credit Card Fraud Detection with a Neural- Network", Proc. 27th Hawaii Int'l Conf. System Sciences: Information Systems: Decision Support and Knowledge-Based Systems, 1994; 3: 621-630.
6. Jain Y, Tiwari N, Dubey S and Jain S "A comparative analysis of various credit card fraud detection techniques", International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, 2019; 7: 5S2: 402-407.
7. Khare N and Sait S Y "Credit Card Fraud Detection Using Machine Learning Models and Collating Machine Learning Models", International Journal of Pure and Applied Mathematics. 2018; 118: 20. 825-838 ISSN: 1314-3395.
8. Kim M and Kim T "A neural classifier with fraud density map for effective credit card fraud detection." In proc. Int'l Conf. Intelligent Data Eng. and Automated Learning, 2002; 378-383.
9. Kumar M S , Soundarya V, Kavitha S, Keerthika E S and Aswini E "Credit Card Fraud Detection Using Random Forest Algorithm", 3rd International Conference on Computing and Communications Technologies (ICCCT), Chennai, India, 2019; 149-153. doi: 10.1109/ICCCT2.2019.8824930.
10. Maes S, Tuyls K, Vanschoenwinkel B and Manderick B "Credit card fraud detection using Bayesian and neural networks", Proceedings of the 1st international naiso congress on neuro fuzzy technologies, 2002.
11. Malini N and Pushpa M "Analysis on credit card fraud identification techniques based on KNN and outlier detection", Third International Conference on Advances in Electrical,

Electronics, Information, Communication and Bio-Informatics (AEEICB), Chennai, 2017; 255-258. doi: 10.1109/AEEICB.2017.7972424.

12. Meng C, Zhou L and Liu B "A Case Study in Credit Fraud Detection With SMOTE and XGBoost", In Journal of Physics: Conference Series, 2020; 1601(5): 052016, IOP Publishing, doi:10.1088/1742-6596/1601/5/052016.
13. Sahayasakila V, Monisha D K, Aishwarya and Yasaswi S V "Credit Card Fraud Detection System using Smote Technique and Whale Optimization Algorithm", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958, 2019; 8(5).
14. Sailusha R, Gnaneswar V, Ramesh R and Rao G R "Credit Card Fraud Detection Using Machine Learning," 4th International Conference on Intelligent Computing and Control Systems (ICICCS), 2020; 1264-1270. doi: 10.1109/ICICCS48265.2020.9121114.
15. Saputra A S and Suharjito S "Fraud Detection using Machine Learning in e-Commerce" International Journal of Advanced Computer Science and Applications (IJACSA), 2019; 10(9): 332-339.
16. Stolfo S, Lee W, Prodromidis A and Chan P K "Cost-based modeling for fraud and intrusion detection: results from the JAM project", Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00, IEEE. 2000; 2: 130-144. DOI: 10.1109/DISCEX.2000.821515.
17. Syeda M, Zhang Y and Pan Y, "Parallel Granular Networks for Fast Credit Card Fraud Detection", Proc. IEEE Int'l Conf. Fuzzy Systems, 2002; 572-577.
18. Ribeiro M T, Singh S and Guestrin C "Why should I trust you? Explaining the predictions of any classifier", In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge Discovery and Data Mining, 2016; 1135-1144. DOI: http://dx.doi.org/10.1145/2939672.2939778.
19. Roy A, Sun J, Mahoney R, Alonzi L, Adams S and Beling P "Deep learning detecting fraud in credit card transactions", Systems and Information Engineering Design Symposium (SIEDS), IEEE, 2018.