A Systematic Method for the Explanation of Credit Card Fraud Detection

Khan Md. Hasib
ID-20266015
Department of CSE
Brac University
khan.md.hasib@g.bracu.ac.bd

Rashik Hasnat
ID-20266010
Department of CSE
Brac University
rashik.hasnat@g.bracu.ac.bd

S M Monwar Adeeb
ID-20266014
Department of CSE
Brac University
sm.monwar.adeeb@g.bracu.ac.bd

***Abstract*—The use of credit cards has grown significantly, owing to a rapid advance in electronic commerce technologies. As the most common payment system for online and daily transactions becomes the credit card, there are still growing fraud cases. 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 article, we model the sequence of transactions in handling credit card transactions using five traditional machine learning models with LIME methodology and demonstrate how it can be used to detect fraud. Five different types of classification models namely Decision Tree, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest, and XGBoost are used and the measurement of results by two-performance measures (accuracy and f1-score) is necessary to show the effectiveness of the classification prediction. Therefore, it is important to consider whether a model makes a particular prediction. Thus, we then train an interpretable LIME model 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.**

***Index Terms*—credit card fraud, e-commerce security, fraudulent activities, machine learning, LIME**

I. Introduction

Credit card fraud is a growing concern in the world today with growing fraud in government departments, the private sector, the banking industry and many other organisations. In today's environment, high internet reliance is the explanation for a rise in the incidence of credit card fraud transactions. However, fraud has escalated not only online, but also offline transactions. Although data mining methods are being used, the result is not very successful in detecting these credit card frauds. The only way to mitigate these threats is to predict fraud by using successful algorithms, which is a promising way to eradicate credit card fraud. The sky is big as the market moves to e-commerce credit card payments. Any vulnerabilities in these processes have increased fraudulent transactions with the developed e-banking system. Fraud can be avoided in the first place by measures of prevention and identification. Prevention prevents attacks by fraudsters by serving as a barrier of security. Detection occurs as evasion fails. Detection often allows it possible to detect and alert as soon as the unusual transaction takes place. Machine Learning approaches are used to establish mathematical methodologies that can classify non-legitimate transactions on the basis of the sum and duration of such transactions. Card fraud can be defined as internal and foreign fraud. If a bank worker is a customer who has a false identity, which is an internal fraud. Outer fraud can be referred to as a stolen credit card used by fraudsters to earn money.

Fraud detection is the process of discriminating transactions for credit card purposes. Two forms of transactions may be legitimate or fraudulent [1]. Typical fraud detection systems provide an automatic academic degree system and manual methodology. Automated method depends on fraud identification guidelines. It analyzes all new transactions and assigns false score. Fraud analysts produce manual technique [2]. Credit card fraud monitoring attempts to classify fraud dependent on a dataset. 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 only distorted, but are not necessarily loaded with data throughout - column. Dataset providers are aware of the privacy and protection condition of consumers whose data they have. Thus, in a number of datasets, only one or two alphabetical attributes are used in numerical tables. Another similar issue that happens much of the time as the credit card detection process has taken place, non-legitimate records are complex, fraudulent transactions appear to look like legitimate transactions. 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 goal of this paper is to suggest a new model for the identification of credit card fraud, using the LIME approach that describes the accuracy that we have achieved is accurate and the proposal is tailored to the real-time transaction and to reduce the rate of false alarm. 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 recognise the key points of fraud. This technique also aims to improve current algorithms for greater accuracy in the future.

The remaining parts of the paper will compose of the following sections:

- A quick overview of the various detection of credit cards is discussed in Section II.
- We have discussed about the Dataset, our Proposed Model and the details of algorithms in the Methodology segment of section III.
- In Section IV, descriptions of our approach are included and the consequence discussion of the nature of the classification model is covered.
- Section V would include the conditions for conclusion and there is an acknowledgement in section VI.

II. Related Work

Latest software to identify credit card fraud using a broad range of research methods and different fraud identification techniques with a particular focus in neural networks, data processing, and data mining distribution. There are many other forms to detect such credit card fraud. Upon completion of the literature survey, different methods of detecting credit card fraud may be concluded that there are other approaches to detect credit card fraud in Machine Learning itself. Credit card fraud detection research utilizes Machine Learning [3], [4] and Deep Learning [5] algorithms. In this portion, we are developing work on two different points: I methods that are readily available for fraud. Detection and (ii) strategies for handling imbalanced data. Any methods available to treat imbalanced data [6]. (a) sorting approaches (b) sampling methods (c) associated strategies. Here are some of the algorithms used to identify credit theft including support vector machine (SVM), decision trees, logistic regression, gradient boosting, K-nearest neighbor, etc.

In 2019, Yashvi Jain, NamrataTiwari, Shripriya Dubey, Sarika Jain researched numerous techniques[7] for detecting credit card fraud, e.g. support vector machines (SVM), artificial neural networks (ANN), Bayesian networks, Hidden Markov Model, K-Nearest neighbors (KNN) Fuzzy Logic method and Decision Trees. They find that the algorithms k-nearest neighbor, decision trees, and SVM have medium-level accuracy. The Reasoning of Fuzzy and Logistic regression provides the lowest precision of any other algorithms. Neural networks, naive bays, ambiguous structures, and KNN provide a high degree of incarceration. Logistic regression, SVM, decision trees provide a strong degree of detection and the rate is at the medium range. There are two algorithms, namely ANN and Naïve Bayesian Networks, which perform better for all parameters. They're really costly to practice. There is a big downside to these algorithms. The downside is that these algorithms do not generate the same outcome with both forms of algorithms. For the surroundings. They offer better results with one form of dataset and worse results with another type of dataset. Algorithms such as KNN and SVM provide impressive outcomes for limited datasets and algorithms such as logistic regression. Fuzzy logic systems have decent consistency for raw and non-sampled data.

Again, Ghosh and Reilly [8] have suggested the detection of credit card theft through a neural network. They also set up a monitoring system, which is trained on a wide range of credit card purchases. These transactions include, for example, fraud cases involving missing cards, stolen cards, application fraud, fraudulent fraud, mail-order fraud, and non-received (NRI) fraud. Syeda et al. [9] have recently used parallel granular neural networks (PGNNs) to increase the pace of data mining and information exploration in the identification of credit card fraud. For this reason, a full structure has been placed in motion. Aleskerov et al.[10] are introducing CARDWATCH, a database mining method used for the analysis of credit card fraud. The framework, based on a neural learning module, offers an interface to a broad range of commercial databases. The distorted distribution of data and the combination of valid and illegitimate purchases have been described as the two major factors for the difficulty of the identification of credit card fraud [11]. Based on this finding, the fraud density of actual transaction data is used as a trust value and the weighted fraud score is created to minimize the amount of errors observed. In 2019, Sahayasakila V, D.Kavya Monisha, Aishwarya, Sikhakolli Venkatavisalakshiswshai Yasaswi clarified Twain's essential algorithmic techniques [12], Whale Optimization Techniques (WOA) and SMOTE (Synthetic Minority Oversampling Techniques). They are primarily and the goal was to increase the speed of consolidation and to address the issue of data imbalance. The dilemma of class imbalance is solved by the SMOTE technique and the WOA technique. The algorithm also increases the tempo, durability and performance of the method.

Khare et al.[13] clarified the work on decision tree, random forest, SVM, and logistic regression. They took the heavily distorted dataset and focused with this sort of dataset. Performance measurement is based on precision, flexibility, consistency and accuracy. The findings show that the precision for the Logistic Regression is 97.7 per cent, for the Decision Trees is 95.5 per cent, for the Random Forest is 98.6 per cent, for the SVM classification is 97.5 per cent. They concluded that the Random Forest algorithm has the best accuracy of the others. Algorithms which is known to be the best algorithm to identify fraud. They also concluded that the SVM algorithm has a data imbalance problem and does not help in improved identification of credit card fraud. The key problems described for the credit card fraud identification system are set out below:

- Detecting fraud in a broad data collection where the rate of legitimate transactions is more important than the rate of fraud, which could be negligible.
- In order to eliminate false alarms.
- To understand and dynamically update user actions.
- To increase the accuracy of identification.

To overcome these problems, we suggested an approach where our findings are validated by LIME that proves the

explainable result that the credit card fraud part is correctly identified. We can create five different classification types, Decision Tree, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest, and XGBoost. After classification, LIME arrives to assess in this part as the model is implemented and the explanation for individual predictions should be at least geographically accurate, i.e. it must contribute to how the model behaves in the vicinity of the forecast individual observation. In our function, in the case of LIME, local fidelity does not equal global fidelity: features that are significant globally cannot be essential 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.

III. Proposed Methodology

The key aim of this paper is to differentiate transactions that include both fraud and non-fraud transactions in datasets utilizing algorithms such as Decision Tree, K-Nearest Neighbors (KNN), Vector Machine Help (SVM), Random Forest, and XGBoost. All three algorithms are then related to the algorithm that better identifies credit card fraud purchases. The proposal model for the issue of identification of credit fraud as seen in Figure 1. Requires data splitting, model training, model execution, then comes the LIME approach to explore the technique before the assessment.

The fraud dataset for the Kaggle credit card is used in this model and the dataset is used for pre-processing. Now, in order to prepare the model, we need to isolate the data from the training data and the test data. We use training data to plan Decision Tree, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest, and XGBoost. And we plan the three of them. The accuracy and f1-score of the bot models are eventually calculated.

Again, we preprocessed dataset by testing and training data. We split the data into 80:20 where we train 80% of data and test 20% of the data. After the data is divided, we also quantify the percentage of fraud cases in the total transactions reported on the train data. Build a training set that helps the algorithms to attain those attributes. From table I, we see the overview of case amount in data.

TABLE I
Overview of Case Count in Data

Number & Percentage	Case Count
Total Number of Cases	284807
Number of Non-Fraud Cases	284315
Number of Fraud Cases	492
Percentage of Fraud Cases	0.17%

The independent (X) and the dependent variables are described (Y). For the variables defined, we can split the knowledge into a training set and test set that can then be modelled and tested. Community variables are associated with predictors, but it is obvious that the relationship between them is very small. So their similarity depends on two points. Next, predictors are the key component since they have been assembled by PCA. Second, our class variable has some association, and may be blurred due to the disparity of class variables, which is immense. We can conveniently divide the data using the 'train test split' algorithm in python. In table II, we can visualise the case amount statistics of our data and this are the known transaction identifiers.

TABLE II
Overview of Case Amount Statistics in Data

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

Then, we constructed five separate styles of classification models, namely Decision Tree, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest, and XGBoost. While there are several other models that we can use, these are the most common models used to solve classification problems. All of these models can be easily designed using the algorithms given by the scikit-learn bundle. We can just use the xgboost kit for the XGBoost model. Afterwards, we are designing all the classification models. Finally, the accuracy and the f1-score are determined for both versions. 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.

A. Decision Tree Implementation

A structure with the root node, leaf node, and expanding. The decision tree is a system. Any internal node indicates an attribute demand, each branch is indicated by the trial outcome and each leaf node is indicated by the class. The larger tree node[14] is the root node. The next decision tree is to buy a defining device which indicates whether a company consumer is willing to buy a machine. Any inner node is defined in the attribute examination. Each leaf's node represents the class. Figure 2. Represents the decision tree form. Basically, what algorithm does is break the data into two or more sets. Splitting is achieved with the most important characteristics to render classes as distinct as possible using information gain and entropy. Entropy tests the impurity of the result class in a subset with the attributes of ps in any D dataset as shown in the formula:

$$H(p1, p2, \ldots ps) = \sum_{i=1}^{s} \left(p_i \log \left(\frac{1}{p_i} \right) \right) \tag{1}$$

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

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

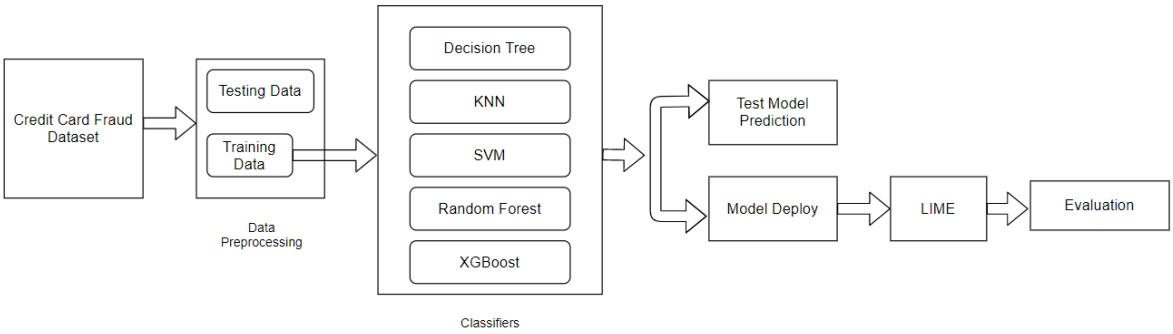

Fig. 1. Proposed Methodology.

We used the 'DecisionTreeClassifier' algorithm to construct the model. Within the algorithm, we've listed 'max depth' to be '4,' which implies that we're enabling the tree to break four times and the 'criterion' to be 'entropy,' which is most close to 'max depth,' but specifies when to avoid splitting the tree. Finally, we have entered and processed the expected values in the 'tree yhat' variable.

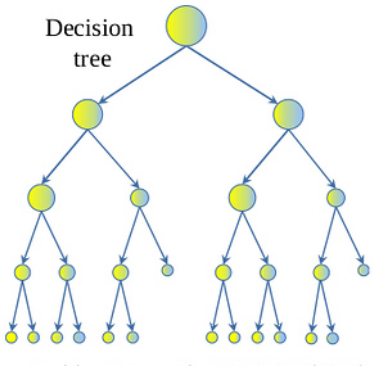

Fig. 2. Structure of Decision Tree

B. KNN Implementation

K-nearest neighbor algorithm is commonly used in detection systems. KNN has also been shown to perform exceptionally well in credit card fraud identification programs utilizing controlled learning strategies [15]. Figure 3. represents the structure of KNN. The new instance question will be listed according to the KNN type in this process. The effects of KNN rely on the following three factors:

- The distance metric used to be determined by the closest neighbours.
- The gap law that is used for the grouping of the Knearest neighbor.
- The amount of neighbors deemed to have classified the latest survey.

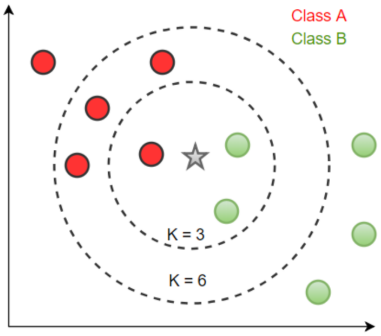

Fig. 3. Structure of KNN

In the case of KNN, we developed the model using the 'KNeighborsClassifier' algorithm and mentioned 'n neighbors' to be '5.' The 'n neighbors' value is picked arbitrarily, but can be chosen optimistically by iterating a number of values, accompanied by fitting and storing the expected values into the 'knn yhat' vector.

C. SVM Implementation

The Support Vector Machine (SVM) is a versatile machine learning method focused on the firm statistical and mathematical foundations of generalization and optimization theory.

It provides a comprehensive methodology for many areas of data mining, including grouping, regression and outlier identification. SVM is based on Vapnik's mathematical learning principle and lies at the intersection of kernel approaches with maximal margin classifiers [16]. Help vector machines have been successfully introduced to multiple real-world concerns such as face identification, intrusion detection, handwriting recognition, knowledge retrieval, and others. In a linearly separable case, there is one or more hyperplanes capable of distinguishing the two types of training data with 100% accuracy normally. Figure 4. depicts the structure of SVM.

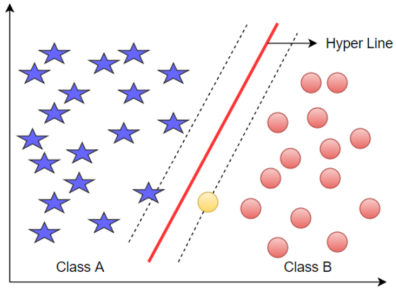

Fig. 4. Structure of SVM

We designed the support vector machine model using the 'SVC' algorithm, and we didn't mention anything within the algorithm since we managed to use the default kernel that is the 'rbf' kernel. After that, we stored the expected values in the 'svm yhat' after fitting the construct.

D. Random Forest Implementation

Random Forest is also referred to as RFA and serves for sorting, regression and other RFA tasks. Several decision-making trees are being installed. This Random Forest Algorithm is developed on directed learning and is based on supervised learning that can be used for classification or regression purposes in the most significant advantage of this algorithm. Random Forest Algorithm offers you greater performance as opposed to all other current schemes and is the most widely used algorithm [17]. Figure 45 represents the structure of Random Forest. Usage of the Random Forest in this article, credit card fraud identification algorithm will give you an accuracy of between 90 and 95 %.

For Random Forest Model, we developed using the 'RandomForestClassifier' algorithm, and we listed 'max depth' to be 4 just like how we built the decision tree model. In the end, suit and store the values in the 'rf yhat.' Note that the key distinction between the decision tree and the random forest is that the decision tree utilizes the whole dataset to create a single model, whereas the random forest uses randomly chosen features to construct different models. That's why a random forest model is used against a decision tree.

E. XGBoost

The XGBoost is a learning algorithm with two positive aspects of individual learning units which render engineering feature redundant. Next, decision trees are invariant to

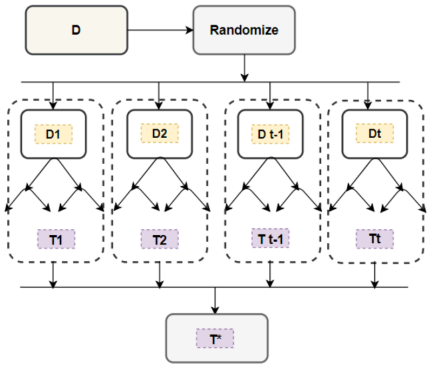

Fig. 5. Structure of Random Forest

monotonous attribute transformations (e.g. scaling or polynomial transformations). Secondly, the relations between roles may be inherently caught and modulated [18]. We may not then need to manually construct functional interactions. The XGBoost model is developed using an additive tree boosting approach with two derivatives. In other terms, gradient gi and hessian hi independently construct a booster tree to handle class imbalance. According to Chen and Guestrin, the objective equation of regularization for the training features and the goal, the tree set with the number of trees K is given as:

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

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

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

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

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

The final model is the XGBoost model. We designed the model using the XGBClassifier algorithm given by the xgboost bundle. We stated the 'max depth' to be 4 and finally equipped and stored the expected values in the 'xgb yhat.'

F. LIME Implementation

LIME, an algorithm which can truly describe each classifier's forecasts by approximating them to an interpretable model locally. Some classifiers use representations that are not at all intuitive to users. Lime explains these classifiers in terms of interpretable representations (words), even if this is not the representation actually used by the classifier. Furthermore, Lime takes human limitations into account: i.e. the explanations are not too long. 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 being explained be denoted:

$$f : \mathbb{R}^d \rightarrow \mathbb{R} \tag{6}$$

In classification, f(x) is the probability (or binary indicator) that x belongs to a given class LIME describes the forecast in such a way that even non-experts can compare and develop an untrustworthy model by function engineering. The explanation given by LIME is as follows:

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \mathcal{L}\left(f, g, \pi_x\right) + \Omega(g) \tag{7}$$

Here we focus on sparse linear models as explanations, and on performing the search using perturbations. We test the fidelity of classification descriptions that can be represented in their own right (Decision Tree, K-Nearest Neighbors (KNN), Help Vector Machine, Random Forest and XGBoost). We train all five classifiers in particular to ensure that the maximum of features they use is 2 in our dataset for each case and we therefore know the most significant features these models consider relevant. For each prediction on the test set, we generate explanations and compute the fraction of these main features that are recovered by the explanations using consfusion matrix. Through LIME, we try to explain the model and determine whether the explanations can be used for model choosing, simulating the situation when a person has to distinguish between two opposing fraud card. The purpose of this experiment is to determine whether a consumer can classify a better classifier based on the descriptions of fraud instances from the validation collection.

IV. Results and Experiments

A. Dataset

The data collection shall be taken from the European Credit Card Company with respect to theft through credit card information. The Kaggle takes the data package. The data collection includes the payments made in September 2013 by the credit card operators. The data collection consists of transactions done within two days. The set of data contains 284,807 purchases, 492 of which are scams. This is just 0.172% of the purchases. The input variable data collection is converted to the PCA transition numerical values. It is for secrecy considerations. It is not possible to convert 'Time' and 'Amount' attributes into PCA. The 'Time' class represents the gap between the sale and the first transaction. The class 'Amount' represents the cash and the dealing. Another relevant 'Class' feature reveals whether the transaction is dishonest or not. Number 1 indicates that is a con and 0 implies purchases without theft.

B. Evaluation Criteria

In this process we are going to evaluate our built models using the evaluation metrics provided by the scikit-learn package. Our main objective in this process is to find the best model for our given case.The calculation measures we will use are the score metric, the score calculating f1 and eventually the matrix of uncertainty. We need to test parameters such as accuracy and f1 score in order to equate different algorithms. Then we apply LIME in model deployment for evaluate the accuracy of our work.

1) Accuracy: Accuracy score is one of the most common assessment criteria commonly used to assess classification models. The precision score is determined simply by calculating the amount of accurate predictions made by the model by the total number of predictions made by the model (can be multiplied by 100 to transform the result into a percentage). Generally speaking, it can be expressed as:

$$\text{Accuracy } = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

To do it in python, we can use the 'accuracy_score' method provided by the scikit-learn package.

2) F1-Score: The F1 or F-score score is one of the most common assessment metrics used to test classification models. It can be clearly described as the harmonic mean of precision and recall of the model. It is determined by dividing the output by the precision of the model and by extracting the amount obtained by applying the precision of the model and by retrieving and eventually multiplying the result by 2. It can be expressed as follows:

$$\text{F1 Score } = \frac{2 \times (\text{ Precision x Recall })}{\text{Precision } + \text{ Recall}} \tag{9}$$

The F1 score can be calculated easily in python using the 'f1_score' method provided by the scikit-learn package.

The Table III depicts the result of our five model that predicts the credit card fraud detection. From the table, we analysis that Random Forest gives the highest percentage of detection among the others on basis of accuracy and f1-score.

TABLE III
RESULTS OF OUR FIVE CLASSIFIERS TO DETECT CREDIT FRAUD

Model	Accuracy	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%

3) Confusion Matrix: The matrix of confusion is therefore drawn up. The instability matrix is a matrix of 2*2. The matrix contains four outputs, TPR, TNR, FPR, FNR. In the ambiguity matrix it is possible to extract parameters such as sensitivity, specificities, coherence and error rate. And we fit the credit card to detect the fraud. The performance of the uncertainty matrix is as follows:

- True Positive Amount, which can be described as the amount of fraudulent transactions that are even identified as fraudulent by the method.

$$TPR - \frac{TP}{TP + FN} \tag{10}$$

- True Negative Rate, which can be described as the number of valid transactions that are even classified as legitimate by the method.

$$TNR - \frac{TN}{TN + FP} \tag{11}$$

- False positive rate, which can be described as a number of legitimate transactions that are falsely counted as fraud.

$$FPR - \frac{FP}{FP + TN} \tag{12}$$

- False Negative Rate is characterized as transactions that are fraud but are falsely labeled as legitimate.

$$FNR - \frac{FN}{FN + TP} \tag{13}$$

The object of the data variance classifier should be to decrease the true positive, true negative, false positive and false negative percentage or to maximize the value of confusion matrix.

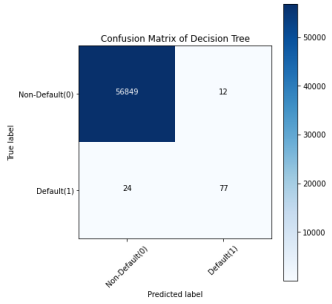

Fig. 6. Confusion Matrix of Decision Tree

Usually, a confusion matrix is a visualization of a classification algorithm that indicates how accurately the model estimated the effects as opposed to the initial ones. Typically, the estimated effects are stored in a vector that is then translated into a correlation table. Using the correlation table, the uncertainty matrix is plotted as a heatmap. Even though there are a range of built-in approaches to visualize an uncertainty matrix, we're going to describe and visualize it from scratch for improved comprehension. Figure 6,7,8,9 & 10 depicts the representation of our five model confusion matrix.

Take the confusion matrix of the XGBoost model as an example. Take a peek at the first row. The first row is for transactions whose real significance for fraud in the test range is 0. As you can measure, the fraud meaning of 56861 is 0. And out of these 56861 non-fraud transactions, 56854 of them were correctly predicted by the classifier as 0 and 7 as 1. This implies that for 56854 non-fraud transactions, the real churn

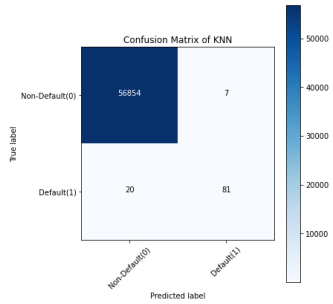

Fig. 7. Confusion Matrix of KNN

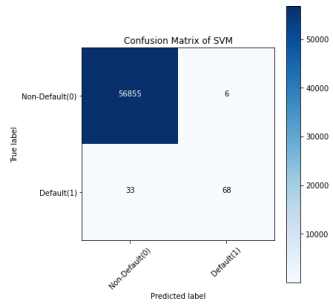

Fig. 8. Confusion Matrix of SVM

value was 0 in the test set, and the classifier also correctly estimated 0 in the test set. We may assume that our model has categorized non-fraud transactions very well.

Let's have a peek at the second row. It seems like there were 101 transactions with a fraud value of 1. The classifier correctly estimated that 79 of them were 1, and 22 of them were incorrectly 0. Error of the model may be assumed to be improperly expected values. Thus, when contrasting the confusion matrix of both models, it can be found that the K-Nearest Neighbors model has done a very successful job of classifying fraud transactions from non-fraud transactions accompanied by the XGBoost model. So we can assume that the most suitable model that can be utilized in our situation is the K-Nearest Neighbors model.

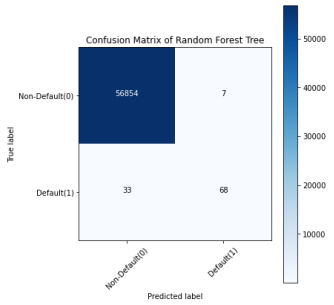

Fig. 9. Confusion Matrix of Random Forest

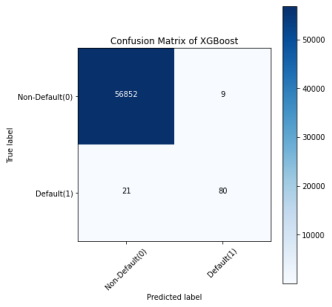

Fig. 10. Confusion Matrix of XGBoost

C. LIME Evaluation

In the final consumer trial we analyze whether reasons for model selection need to be included, simulating the circumstance where an individual wants to assess the validity data between two separate, similarly exact models [19]. We first select v17 amd number features arbitrarily to be "untrustworthy" so as to simulate belief in individual predictions and believe that users know this and do not want to confide in this function. We therefore build oracle "trustworthiness" by labeling "unreliable" black-box predictors when the forecast changes when "reliable" functions are taken out of the case. In the Figure 11 and 12, we see the representation of v17 and Amount feature evaluation using LIME.

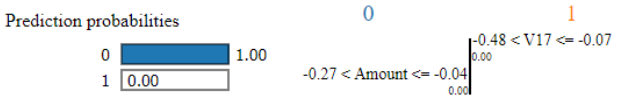

Fig. 11. Prediction of v17 and Amount Feature

Prediction is mistrusted here if any inaccurate aspects are included in the interpretation, since these approaches do not have an indication of the importance of each function to the prediction. Thus, with each test set forecast, we will determine if the simulated consumer trusts each interpretation process and equate it to the trustworthiness oracle.

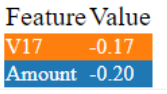

Fig. 12. v17 and Amount Feature

While we artificially pick which features are untrustworthy, these findings suggest that LIME is helpful in determining confidence in individual predictions. We test the fidelity of descriptions on classifiers that can be understood on their own. Thus we consider the Time and Amount feature here, LIME can make this feature human explainable and the feature values tell that it is detect the credit card fraudulent with all our five classifier tends to accurate.

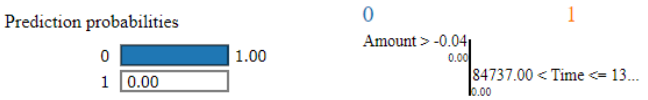

Fig. 13. Prediction of Time and Amount Feature

In the Figure 13 and 14, we see the representation of Time and Amount feature evaluation using LIME which is human explainable.

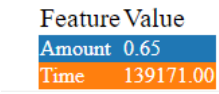

Fig. 14. Time and Amount Feature

V. Conclusion

The present paper presents a conceptual method for the detection of 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. While in KNN accuracy, Random Forest has achieved good performance, but we have based not just on accuracy, but also on market importance. Because deep learning has broadened the possibility of detecting fraudulent transactions, we have attempted to show how classical algorithms can be identified. Comparison between these algorithms helped us infer that Random Forest and XGB are both precise and economical. Problems like this, with the removal feature that helps us to achieve remarkable outcomes, we may concentrate on high recall value. Our future work will focus on addressing this issue with a broad variety of functions and will put it into line with the state-of-the-art SHAP network. Again, will apply deep learning model too as a classification step whether to see the results of the currently used dataset.

VI. Acknowledgement

We would like to thank **Md. Golam Rabiul Alam Sir** for his time, generosity and critical insights into this project.

References

[1] Maes, Sam, et al. "Credit card fraud detection using Bayesian andneural networks." Proceedings of the 1st international naiso congresson neuro fuzzy technologies. 2002.

[2] tolfo, Salvatore J., et al. "Cost-based modeling for fraud and intrusiondetection: Results from the JAM project." Proceedings DARPAInformation Survivability Conference and Exposition. DISCEX'00. Vol.2. IEEE, 2000.

[3] Adi Saputra1, Suharjito2L: Fraud Detection using Machine Learning in e-Commerce, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 10, No. 9, 2019.

[4] Dart Consulting,Growth Of Internet Users In India And Impact On Country's Economy: https://www.dartconsulting.co.in/market news/growth-of-internet-users-in-india-and-impact-on-countryseconomy

[5] Roy, Abhimanyu, et al:Deep learning detecting fraud in credit card transactions, 2018 Systems and Information Engineering Design Symposium (SIEDS), IEEE, 2018.

[6] Yong Fang1, Yunyun Zhang2 and Cheng Huang1, Credit Card Fraud Detection Based on Machine Learning, Computers, Materials & Continua CMC, vol.61, no.1, pp.185-195, 2019.

[7] Yashvi Jain, NamrataTiwari, ShripriyaDubey,Sarika Jain:A Comparative Analysis of Various Credit Card Fraud Detection Techniques, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7 Issue-5S2, January 2019.

[8] S. Ghosh and D.L. Reilly, "Credit Card Fraud Detection with a Neural-Network," Proc. 27th Hawaii Int'l Conf. System Sciences: Information Systems: Decision Support and Knowledge-Based Systems, vol. 3, pp. 621-630, 1994.

[9] M. Syeda, Y.Q. Zhang, and Y. Pan, "Parallel Granular Networks for Fast Credit Card Fraud Detection," Proc. IEEE Int'l Conf. Fuzzy Systems, pp. 572-577, 2002.

[10] E. Aleskerov, B. Freisleben, and B. Rao, "CARDWATCH: A Neural Network Based Database Mining System for Credit Card Fraud Detection," Proc. IEEE/IAFE: Computational Intelligence for Financial Eng., pp. 220-226, 1997.

[11] M.J. Kim and T.S. Kim, "A Neural Classifier with Fraud Density Map for Effective Credit Card Fraud Detection," Proc. Int'l Conf. Intelligent Data Eng. and Automated Learning, pp. 378-383, 2002.

[12] Sahayasakila.V, D. Kavya Monisha, Aishwarya, Sikhakolli VenkatavisalakshiseshsaiYasaswi: Credit Card Fraud Detection System using Smote Technique and Whale Optimization Algorithm,International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958, Volume-8 Issue-5, June 2019.

[13] Navanshu Khare ,Saad Yunus Sait: Credit Card Fraud Detection Using Machine Learning Models and Collating Machine Learning Models, International Journal of Pure and Applied Mathematics Volume 118 No. 20 2018, 825-838 ISSN: 1314-3395.

[14] Gaikwad, J. R., Deshmane, A. B., Somavanshi, H. V., Patil, S. V., & Badgujar, R. A. (2014). Credit Card Fraud Detection using Decision Tree Induction Algorithm. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 4(6).

[15] N. Malini and M. Pushpa, "Analysis on credit card fraud identification techniques based on KNN and outlier detection," 2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), Chennai, 2017, pp. 255-258, doi: 10.1109/AEEICB.2017.7972424.

[16] Demla, N., & Aggarwal, A. (2016). Credit card fraud detection using svm and reduction of false alarms. International Journal of Innovations in Engineering and Technology (IJIET), 7(2), 176-182.

[17] M. S. Kumar, V. Soundarya, S. Kavitha, E. S. Keerthika and E. Aswini, "Credit Card Fraud Detection Using Random Forest Algorithm," 2019 3rd International Conference on Computing and Communications Technologies (ICCCT), Chennai, India, 2019, pp. 149-153, doi: 10.1109/ICCCT2.2019.8824930.

[18] Meng, C., Zhou, L., & Liu, B. (2020, August). A Case Study in Credit Fraud Detection With SMOTE and XGBoost. In Journal of Physics: Conference Series (Vol. 1601, No. 5, p. 052016). IOP Publishing.

[19] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). " 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 (pp. 1135-1144).