DOI: https://doi.org/10.33103/uot.ijccce.21.3.4

An Efficient Feature Engineering Method for Fraud Detection in E-commerce

Suha M. Najem[1], Suhad M. Kadhem[2]
[1]Computer Science Dept., University of Technology, Baghdad, Iraq.
[2]Computer Science Dept., University of Technology, Baghdad, Iraq.
[1]Cs.19.46@grad.uotechnology.edu.iq, [2]110102@uotechnology.edu.iq

Abstract: *With the speedy expansion of e-commerce, credit cards have also become rising vogue, and that makes online transactions sleek and suitable. In conjunction with rising in online transactions, credit card fraud also increasing, which contributes to losses incurred yearly. As a result, many deep and machine learning methods are produced to fix such as problems like Logistic Regression (LR), Support Vector Machine (SVM), Naïve Bayes (NB), and other algorithms, but the current models are still not accurate. Moreover, sometimes the used datasets still need further preprocessing, since that has been approved the important role of feature engineering in performance optimization. In this paper, effective feature engineering and feature selection methods have been produced for preprocessing the raw dataset, which was transformed with Exploratory Data Analysis (EDA). Then LightGBM, XGboost, and Random forest classifiers are used for fraud detection. Experiments show that the LightGBM and XGboost models achieved the best accuracy with 100% after applying further preprocessing on the dataset.*

Index Terms— *Fraud detection, Feature engineering, Machine Learning, LightGBM, Random forest, XGboost*

I. INTRODUCTION

The use of electronic credit cards increased by the expansion of technology of communications, which in turn contribute to the prosperity of e-commerce companies[1]. The flexibility of the Internet has expanded the capability of fraudsters to attack companies. Fraud and Security have direct effects on merchants and financial institutions' monetary losses. Any data breaches that happened with financial companies or merchants, will most lead to a temporary loss in their stock price and sales if they are common.[2]

Generally, fraud detection system used for recognizing the transactions if it is fraudulent or legitimate [3]. Many institutions spent many funds to prevent fraudulent transactions by developing a powerful detecting system. The fraud detection algorithm is the most important part of such systems [4]. Many challenges appear when building any Fraud Detection System (FDS), one of which is an imbalance in datasets that is the number of non-fraud transactions outperforms the fraud transactions[5]. Another major challenge is how to generate new successful features of credit card transactions for machine learning from raw datasets [6]

For a long time, many machine learning algorithms have been proposed to detect Fraudulent transactions and behavior, every one of them has its advantages and disadvantages or limitation. When the Fraudsters always develop their behaviors to look legitimate, fraud detection systems need to be updated to catch up with such activities.

Received 24/4/2021; Accepted 18l7l2021

 ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.21.3.4

In this paper, the raw dataset is transformed by Exploratory Data Analysis (EDA), preprocessed by using series of preprocessing methods, and data imbalance is solved by using the (SMOTE) algorithm to optimize the performance of machine learning models[7]. The dataset was trained and tested using Light Gradient Boosting Machine (LightGBM), Extreme Gradient Boosting (XGboost), and Random Forest (RF) classifiers.

The rest of the paper is arranged as follows: section 2 reviews some related works. Section 3 shows the proposed model with the parameters. Section 4 explains the dataset and feature engineering. In section 5 evaluation metrics and experiments of LightGBM and other machine learning models are discussed. Finally, section 6 shows the conclusion.

II. RELATED WORKS

Fraud detection models are increasing significantly with the rising of Fraudster methods, which caused losses every day. There are many challenges when design machine learning models, the dataset is one of these challenges. Choosing, analyzing, and preprocessing the suitable dataset is very important. Dataset imbalance is one of the problems facing designers when implementing Fraud detection systems. Many studies have discussed some solutions for solving these problems.

One of the important algorithms used in the process of data imbalance is Synthetic Minority Over Sampling Technique (SMOTE), in [8] confirmed that the SMOTE optimizing the data classification performance. The classifiers and their results that are used in this paper are Artificial Neural Network (ANN) with an accuracy of 96%, Random Forest (RF) and Naive Bayes (NB) with 95%, and DT accuracy was 91%.

In [9] Standardization is the only technique used for preprocessing the dataset, then a hybrid method consists of SMOTE algorithm to achieve data balancing followed by the XGboost algorithm. They approved that solving the data imbalance problem increases the performance of the proposed model. The XGboost algorithm with SMOTE achieved an AUC score of (0.98).

Several resampling techniques are used to solve data imbalance problems in [10] like Tomek Links Removal, random under-sampling, (SMOTE) algorithm, random oversampling, and combination from SMOTE and Tomek Links Removal. Then some machine learning models have used Fraud detection. The combination of SMOTE and Tomek Links Removal approved its efficiency in the performance of the models, the results for the models were evaluated by several metrics one of them is ROC. Logistic Regression achieved a ROC score with (0.973), Random Forest (0.977), and XGboost score with (0.980).

Some solutions have been mentioned in [11] to solve the class imbalance problem, the first one is used to balance the classes as a preprocessing step like under sampling and oversampling techniques, the second is used through the classification algorithm like One-Class Classification (OCC) and Cost Sensitive (CS). Eight machine learning algorithms were used in the study and comparison has been done among them, but each one of the algorithms still has advantages and disadvantages. The best accuracy results were achieved by C5.0, SVM, and ANN with a score (96%).

Received 24/4/2021; Accepted 18l7l2021

 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.21.3.4

An advanced feature engineering method depending on Homogeneity-Oriented Behavior Analysis (HOBA) was proposed in [12], it is used to create new features from the raw dataset then a deep learning approach is applied to optimize the performance of Fraud detection systems. Six machine learning models were used in their experiment, the highest accuracy score was for Recurrent Neural Network (RNN) with (97.98%).

By depending on both individual and group behavior, a novel approach to feature engineering has been proposed in [13]. The proposed method solves the temporal features problem in the frequency-based feature engineering method by depending on rule-based feature engineering.

A new approach of feature engineering benefited from sequential information for training non-sequential model (Random Forest classifier) used in [14] by using multiple perspectives Hidden Markov Model-based feature engineering method.

From previous studies we can conclude that selecting an effective method for feature engineering and solving data imbalance are very important factors to optimize the models' performance.

Our study focuses on the importance of choosing effective feature engineering techniques to process the raw dataset and how it is important in any machine learning model. Three ensemble classifiers are used to evaluate the dataset, LightGBM, XGboost, and Random forest which are approved for their effectiveness in classification and regression aspects.

III. PROPOSED MODEL

The proposed model is illustrated in

Fig. 1. proposed model . The raw dataset is analyzed by Exploratory Data Analyses (EDA), then feature engineering method is proposed to generate new feature variables from the raw dataset, features preprocessed and transformed to be recognizable by the machine learning model, standardization technique used for feature scaling and Principal Component Analysis (PCA) used for feature reduction. SMOTE algorithm is used for solving dataset imbalance[7]. After that, the dataset was divided into 75% training data and 25% testing data. The training data are used to train the three ensemble classifier LightGBM, XGboost, and Random forest (each one explained in detail below), testing data then are used to evaluate the models. Finally, by using some evaluation metrics, the results of classification will be compared.

Received 24/4/2021; Accepted 18l7l2021

 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.21.3.4

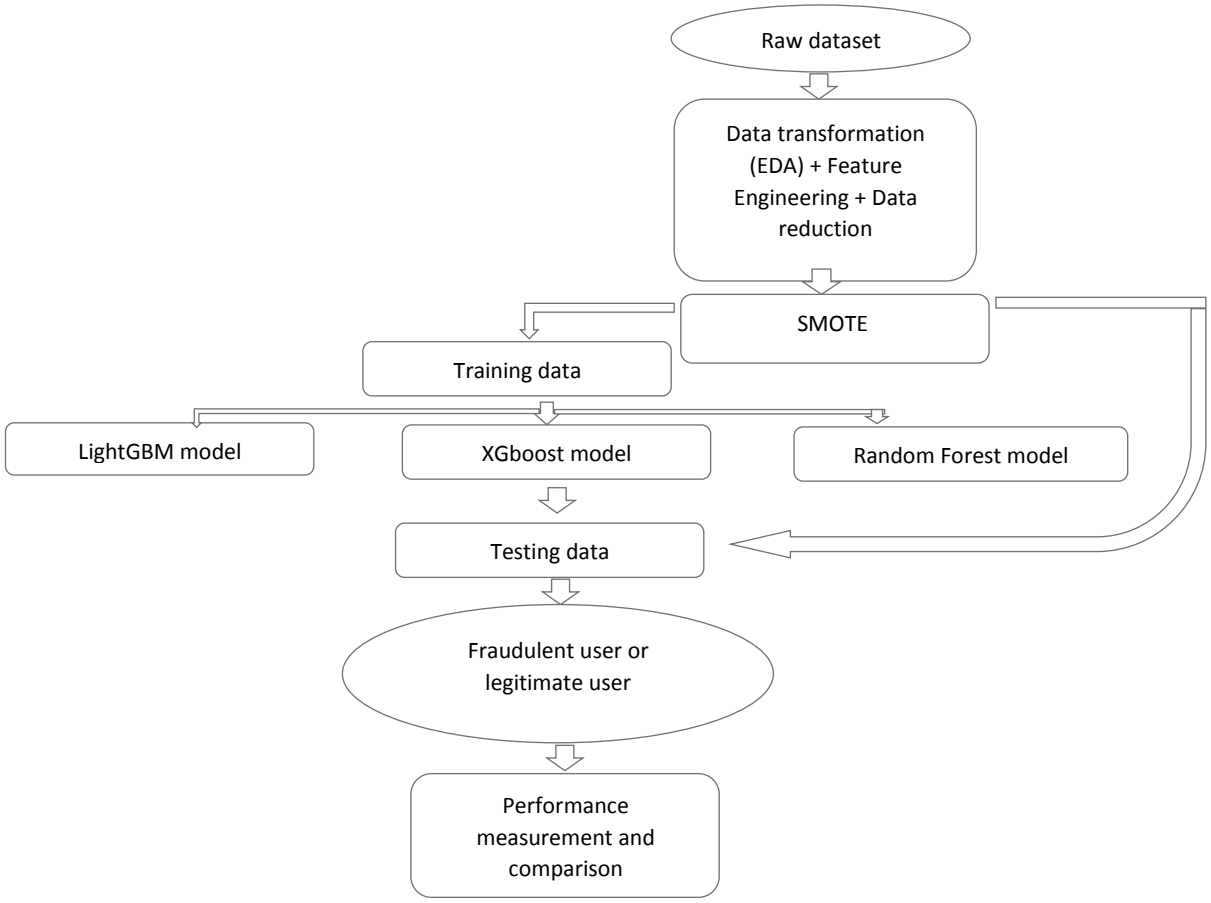

FIG. 1. PROPOSED MODEL

LightGBM is one of the Gradient Boosting Decision Tree (GBDT), which depends on series of weak learners (Decision trees). LightGBM provides many Improvements on GBDT concept, such as Gradient-based One-Side Sampling (GOSS) (ALGORITHM 1) and Exclusive Feature Bundling (EFB) ((ALGORITHM 1)[15]. These improvements solve the weakness found in traditionally GBDT which is the time-consuming problem in large datasets results from the scanning process on all data instances for each feature to compute information gain at each split point. The GOSS algorithm firstly selects samples with a large gradient for training the model, and only chooses small parts of low gradients samples of data. This will speed up the training and obtain higher performance. When the data have many tags in the training process that may be lead to have more empty values. So, some special columns can be combined with keeping the information from loss. Therefore, the dimensions can be reduced by EFB that will accelerate the training process. Subsequently, LightGBM could outperform many other machine learning algorithms in accuracy and speed[15].

For LightGBM algorithm parameters like learning rate, colsample-bytree, and num-leaves, etc. was tuned (as shown in Table 1). The other parameters (LightGBM has more than 100 parameters) stay default.

Received 24/4/2021; Accepted 18l7l2021

DOI: https://doi.org/10.33103/uot.ijccce.21.3.4

XGboost is one of the ensemble tree algorithms (as shown in *Fig.* 2). It is developed based on the gradient boosting decision tree (GBDT). The XGboost generates boosted trees and operates them in parallel. It is efficient in dealing with both classification and regression problems. The optimal parameters can be founded by using several optimization methods[16].

The Random forest is also one of the ensemble tree methods (as shown in *Fig.* 2). The best performance in Ensembles methods achieved when the trees are different, and this dissimilarity among each one of members could be done by applying some randomness: firstly, a separate bootstrapped samples from the training data are used to build each tree; secondly, at each node in implementing the individual trees only a subset of data attributes are considered randomly[17].

TABLE 1. PARAMETERS OF LIGHTGBM

parameters	value
learning_rate	0.07
is_umbalance	true
num_leaves	27
boosting type	gbdt
objective	binary
min_split_gain	0.5
min_child_weight	1
colsample_bytree	0.65
subsample	0.7
reg_lambda	1.2

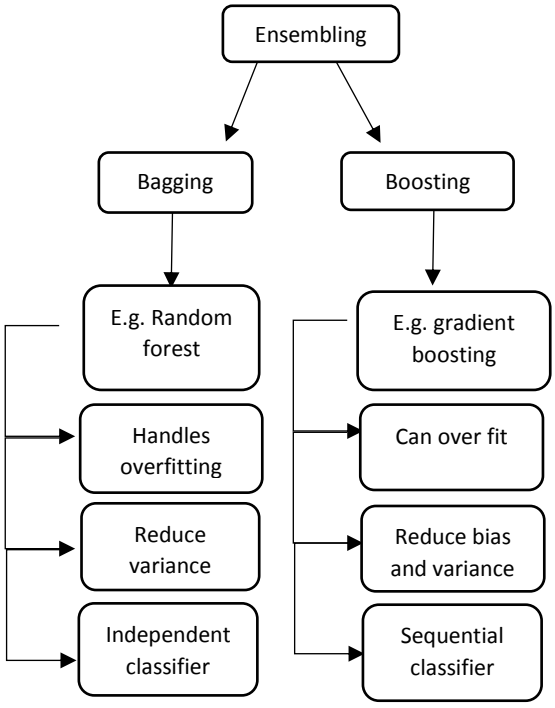

FIG. 2. ENSEMBLE METHODS

Received 24/4/2021; Accepted 18l7l2021

DOI: https://doi.org/10.33103/uot.ijccce.21.3.4


```
Algorithm 2.1: Greedy Bundling [16]
Input: F : features, K : max conflict count Construct graph G
searchOrder ← G.sortByDegree()
bundles ← {}, bundlesConflict ← {}
for i in searchOrder do
    needNew ← True
    for j = 1 to len(bundles) do
        cnt ← ConflictCnt(bundles[j],F [i])
        if cnt + bundlesConflict[i] ≤ K then
            bundles[j].add(F [i]), needNew ← False
            break
   if needNew then
       Add F [i] as a new bundle to bundles Output: bundles
Output: bundles
Algorithm 2.2: Merge Exclusive Features
Input: numData: number of data
Input: F : One bundle of exclusive features
binRanges ← {0}, totalBin ← 0
for f in F do
    totalBin += f.numBin
    binRanges.append(totalBin)
newBin ← new Bin(numData)
for i = 1 to numData do
    newBin[i] ← 0
    for j = 1 to len(F ) do
        if F [j].bin[i] ≠ 0 then
            newBin[i] ← F [j].bin[i] + binRanges[j]
Output: newBin, binRanges
```

ALGORITHM 2. EFB ALGORITHM

```
Algorithm 1: Gradient-based One-Side Sampling(GOSS)[16]
Input: I: training data, d: iterations
Input: a: sampling ratio of large gradient data
Input: b: sampling ratio of small gradient data
Input: loss: loss function, L: weak learner
models ← { }, fact ←(1−a)/ b
topN ← a × len(I ) , randN ← b × len(I )
for i = 1 to d do
    preds ← models. Predict(I )
    g ← loss(I , preds), w ← {1,1,...}
    sorted ← GetSortedIndices(abs(g))
    topSet ← sorted[1:topN]
    randSet ← RandomPick(sorted[topN:len(I)], randN)
    usedSet ← topSet + randSet
    w[randSet] × = fact ► Assign weight fact to the small
    gradient data.
    newModel ← L(I [usedSet], − g[usedSet], w[usedSet])
    models.append(newModel)
```

ALGORITHM 1. GOSS ALGORITHM

Received 24/4/2021; Accepted 18l7l2021

DOI: https://doi.org/10.33103/uot.ijccce.21.3.4

IV. I. THE DATASET

A dataset contains real transactions occurring in a website selling clothes [18]used to train and test the proposed models. It contains 151,113 transactions with 12 features. The raw dataset contains information of new customers (device-id, IP address, source, browser, age, country, and sex) and their activities (signup-time, purchase-time, and purchase-value). Exploratory data analysis (EDA) shows a strong indication of fraud in flash transactions noticed by calculating the time difference between purchase-time and signup-time. The dataset contains two files type of comma-separated values (CSV), which store numbers and text as tabular data in plain text, so all data lines contain an equal number of fields [19]. The first file(Fraud-_data.csv) contains features as illustrated in(Table 3), the second file is (IpAddress_to_Country.csv) contains 3 features as illustrated in (Table 2). By using left join to combine the two files, then feature extraction and preprocessing are done as explained in the next section.

TABLE 3. FRAUD_DATA TABLE

variables	Type
user_id	Number
signup_time	Date Time
purchase_time	
purchase_value	Number
device_id	Categorical
source	
browser	
sex	
age	Number
ip_address	Categorical
class	Number

TABLE 2. IPADDRESS_TO_COUNTRY TABLE

variables	Type
lower_bound_ip_address	Number
upper_bound_ip_address	
country	Categorical

IV. II. FEATURE ENGINEERING

The raw data consists of the columns as mentioned in Table 2. Features including (source, user id, sex, browser, country, and age) can be considered as identification information, feature transformation could be done to be recognizable by the machine learning model directly. But, features like (device-id, purchase time, signup time, and IP address) are activity-based that cannot be directly transformed for machine learning. Instead of that much worthy information can be extracted for the machine learning model.

The Extracted characteristics can be explained as follows:

Received 24/4/2021; Accepted 18l7l2021

 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.21.3.4

1) The time difference between sign-up time and purchase-time is a strong indicator of fraudulent activity likes flash transaction or time difference when become periodic or constant.

2) The IP address is always unique .So, many users are using the same IP address and this is a good indicator of fraudulent activity. Therefore, adding a feature like "IP users" to determine the maximum number of users who can use the identical IP address will detect such activities.

3) In a similar fashion, when the same device is used by many users may be alarm of Fraudulent activity. So, determine the number of users who can use the same device-id for transactions may be a good detector.

4) Again purchase time and sign-up time can be benefitted from daily and weekly transaction features which are more frequently occurred. If there were discovered patterns of transactions occurring, it may indicate fraudulent activities. Therefore, features like "week-of-the-day" and "week-of-the-year" are needed.

5) Features like "total-purchase" and/or "average-purchase" will be useful since many users can share the same device-id and may suggest that the activity is Fraudulent or not.

6) Fraudulent activities in some regions may occur more frequently than in other regions (country here). So, ('country count') feature for counting the number of users from the same region (country).

7) Many users can share the same 'purchase time' because of which could be transaction traffic or automatic transaction. Therefore, a feature like 'purchase times' is useful to gather the same purchase time.

Then numeric features skewness reduced by using standardization as a feature scaling technique, the large values of numeric features scaled-down, categorical features encoded, redundant and correlated features were dropped by using (PCA) as a feature selection technique (feature selection is an important step to choose a subset of data to obtain the perfect solution) [20] Finally, splitting the data into 75% for training and 25% for testing.

V. I. EVALUATION METRICS

There is a huge number of metrics for model evaluation to choose from, here several measures have been used depending on the confusion matrix, like accuracy, Recall, Precision, and F1-score (see Table 4). Accuracy is an evaluation metric for classification models. It is defined as the number of correct predictions divided by the total number of predictions as shown in equation (1)

$$\textbf{Accuracy} = \frac{\textbf{TP+TN}}{\textbf{TP+TN+FP+FN}} \quad \textbf{(1)}$$

By using the (confusion matrix) the accuracy is usually calculated. The number of false-positive errors **TPTP+FP** ()), whereas the number of false-negative errors will minimize by maximizing the recall (as shown in equation (3)).

$$\textbf{Precision} = \frac{\textbf{TP}}{\textbf{TP+FP}} \quad (2)$$

Received 24/4/2021; Accepted 18l7l2021

DOI: https://doi.org/10.33103/uot.ijccce.21.3.4

$$\textbf{Recall} = \frac{\textbf{TP}}{\textbf{TP+FN}} \tag{3}$$

F-measure is a single measure that combines precision and recall and captures both properties (as shown in equation $\textbf{F1-score} = \frac{2\times(\textbf{Recall}\times\textbf{Precision})}{(\textbf{Recall}+\textbf{Precision})}$ ()).

$$\textbf{F1} - \textbf{score} = \frac{2\times(\textbf{Recall}\times\textbf{Precision})}{(\textbf{Recall}+\textbf{Precision})} \tag{4}$$

In addition ROC curve (Receiver Operating Characteristic curve) used as a plot shows the binary classification model performance on the positive class. The True Positive Rate (TPR)(as shown in equation $\textbf{TPR}=\frac{\textbf{TP}}{\textbf{TP+FN}}$ (5)), and the False Positive Rate (FPR) (as shown in equation 6) are represented by the x-axis and y-axis respectively[21].

$$\textbf{TPR} = \frac{\textbf{TP}}{\textbf{TP+FN}} \tag{5}$$

$$\textbf{FPR} = \frac{\textbf{FP}}{\textbf{FP+TN}} \tag{6}$$

V. II. EXPERIMENT RESULTS

This section shows the process of training and validation of our model on the same dataset before and after the further preprocessing. (As shown in the Table) the performance of the three models increased when the data scaled and using (PCA) for feature selection. For each classifier, five metrics (Precision, Recall, F1_score, AUC and accuracy) are calculated for the class 0 (fraudulent transaction) and class 1(legitimate transaction). LightGBM and XGboost classifiers achieved the best accuracy with 100%, Random forest in the two situations achieved less accuracy than LightGBM and XGboost classifiers but, a good accuracy with 99%. Because the results of classifiers are very convergent, the LightGBM classifier will be relied upon to illustrate the confusion and feature importance metrics. Area Under Curve (AUC) is calculated depending on two areas in the confusion matrix (as shown in *Fig.* 3 and *Fig.* 4).

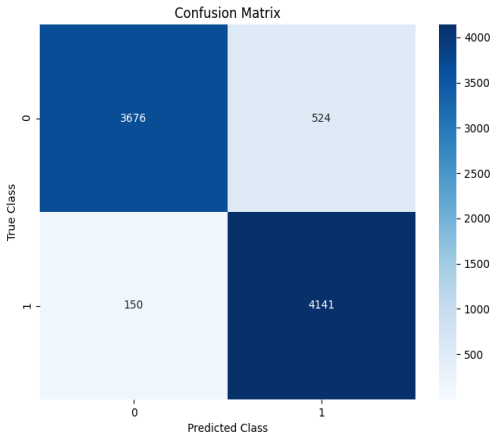

FIG. 4. CONFUSION MATRIX OF LIGHTGBM CLASSIFIER BEFORE MODEL OPTIMIZATION

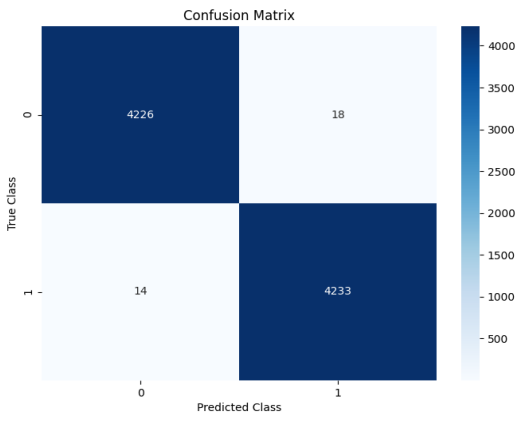

FIG. 3. CONFUSION MATRIX OF LIGHTGBM CLASSIFIER AFTER MODEL OPTIMIZATION

Received 24/4/2021; Accepted 18l7l2021

DOI: https://doi.org/10.33103/uot.ijccce.21.3.4

The plotting processes the True Positive Rate (TPR) and False Positive Rate points (FPR) in (a two-dimension plane) created the Receiver Operating Characteristic curve (ROC). [As shown in *Fig.* 5] the area under the ROC curve is defined as the AUC score.

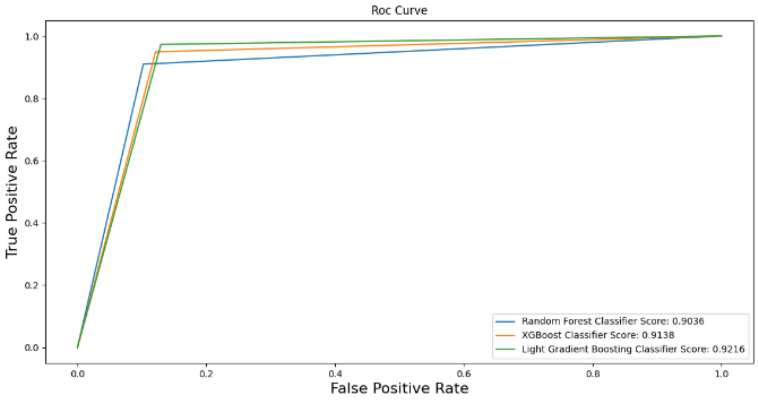

FIG. 5. ROC CURVE

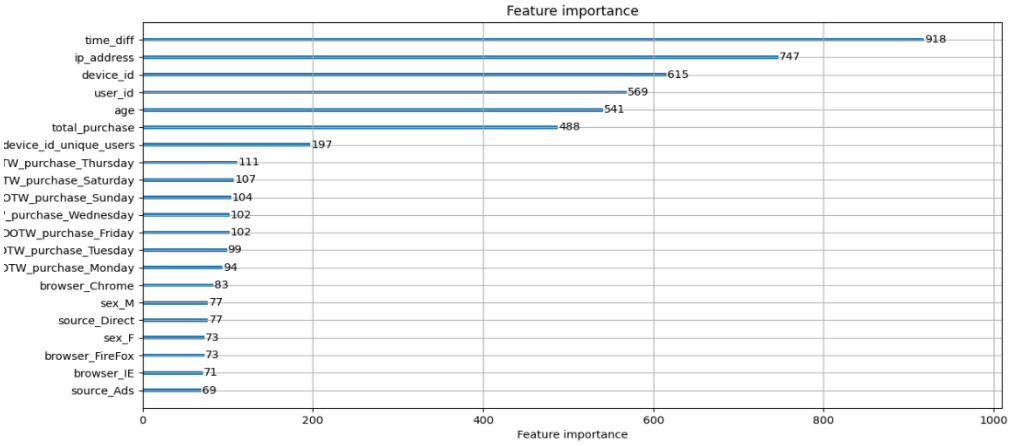

FIG. 6. FEATURE IMPORTANCE BY LIGHTGBM BEFORE THE OPTIMIZATION

Algorithms like LightGBM are easy to represent the importance of each feature. The importance of features are evaluated by measuring the value of error in model prediction after permuting the feature, so the feature considered as important if any mixing up in its values contributes to maximizing the model errors. In (*Fig.* 6), the importance of features by LightGBM organized in lessening order before further preprocessing applied. So, this graph produced an imagination about the features which are so important to keep or features with fewer scores that are not important and could be deleted to optimize the performance. The accuracy of the proposed models is compared with other studies' accuracy (as shown in Table). In study [8] the highest accuracy of their models achieved 95% by ANN and NB. Whereas in study [11], the four models give the same accuracy with 96%, while in study [12] achieved higher accuracy than the previous experiments with 97.98% for Recurrent Neural Network (RNN) Finally, the table showing that LightGBM and XGboost achieved the best accuracy among other classifiers with our models.

TABLE 4. MODEL METRICS RESULTS

Received 24/4/2021; Accepted 18l7l2021

DOI: https://doi.org/10.33103/uot.ijccce.21.3.4

paper number	DT	NB	RF	ANN	BBN	SVM	LR	KNN	NSA	BPNN	DBN	CNN	RNN	XGb-oost	Light-GBM
	Models			class	Precision				Recall		F1_score		AUC score	Accuracy	
[8]	91	95	95	85	-	-	-	-	-	-	-	-	-	-	-
[11]	96	93	-	0	0.90				0.89		0.90		0.90	0.906	
	Random Forest			98	94	96	96	95	92	-		-	-		-
[12]	-	-	97.65	1	-	0.89	-	-	-	97.59	0.90	97.94	97.98	-	-
						97.46					98.11				
proposed model	-	-	0.99	-	-	-	-	-	-	-	-	-	-	1.00	1.00
	XGboost			0		0.95			0.88		0.91		0.91	0.917	
				1		0.88			0.95		0.92				
	LightGBM			0		0.98			0.87		0.92		0.92	0.924	
				1		0.88			0.98		0.93				
	Random Forest+ standardization +PCA			0		0.99			0.98		0.99		0.986	0.99	
				1		0.98			0.99		0.99				
	XGboost+ standardization +PCA			0		1.00			1.00		1.00		0.996	1.00	
				1		1.00			1.00		1.00				
	LightGBM+ standardization +PCA			0		1.00			0.99		1.00		0.995	1.00	
				1		0.99			1.00		1.00				

TABLE 5. COMPARISON RESULTS WITH OTHER MODELS

VI. CONCLUSIONS

Credit card Fraud keeps rising, which represents challenges, so developing different techniques for detecting the Fraud is necessary. In this paper, a feature engineering method was used by transforming the raw dataset using EDA, which shows that flash credit card transactions are an important indicator of Fraud. Then several preprocessing methods like standardization and dimensionality are reduced by using PCA, which approved its effectiveness in performance optimization. After that three classifiers are used for classification, LightGBM and XGboost achieved the best accuracy after preprocessing the dataset. Furthermore, LightGBM and XGboost algorithms outperformed others like Naïve Bayes, K-Nearest Neighbor, Random Forest, Artificial Neural Network, Support Vector Machine, and Logistic Regression. The future work will use a larger dataset with new feature engineering, and try to tune

Received 24/4/2021; Accepted 18l7l2021

 ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.21.3.4

LightGBM parameters using one of the optimization algorithms instead of a manually boring and consuming time process.

REFERENCES

[1] H. Harwani, J. Jain, C. Jadhav, and M. Hodavdekar, "Credit Card Fraud Detection Technique using Hybrid Approach: An Amalgamation of Self Organizing Maps and Neural Networks," *Int. Res. J. Eng. Technol.*, no. July, pp. 5071–5075, 2020, [Online]. Available: www.irjet.net.

[2] David Montague, *Essentials of Online Payment Security and Fraud Prevention*.

[3] R. A. Azeez, "Determination Efficient Classification Algorithm for Credit Card Owners : Comparative Study," vol. 39, no. 01, pp. 21–29, 2021.

[4] Q. Pan, W. Tang, and S. Yao, "The Application of LightGBM in Microsoft Malware Detection The Application of LightGBM in Microsoft Malware Detection," 2020, DOI: 10.1088/1742-6596/1684/1/012041.

[5] I. Benchaji, S. Douzi, and B. El Ouahidi *Using Genetic Algorithm to Improve Classification of Imbalanced Datasets for Credit Card Fraud Detection : Methods and Protocols Using Genetic Algorithm to Improve Classi fi cation of Imbalanced Datasets for Credit Card Fraud Detection*, no. April 2020. Springer International Publishing, 2019.

[6] K. Fu, D. Cheng, Y. Tu, and L. Z. B, "Neural Information Processing - 23rd International Conference, ICONIP 2016, Kyoto, Japan, October 16-21, 2016, Proceedings, Part III," pp. 483–490, 2016, DOI: 10.1007/978-3-319-46675-0.

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "snopes.com: Two-Striped Telamonia Spider," *J. Artif. Intell. Res.*, vol. 16, no. Sept. 28, pp. 321–357, 2002, [Online]. Available: https://arxiv.org/pdf/1106.1813.pdf%0Ahttp://www.snopes.com/horrors/insects/telamonia.asp.

[8] A. Saputra and Suharjito, "Fraud detection using machine learning in e-commerce," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 9, pp. 332–339, 2019, DOI: 10.14569/ijacsa.2019.0100943.

[9] C. Meng, L. Zhou, and B. Liu, "A case study in credit fraud detection with SMOTE and XGboost," *J. Phys. Conf. Ser.*, vol. 1601, no. 5, 2020, DOI: 10.1088/1742-6596/1601/5/052016.

[10] R. Shakya, "Application of Machine Learning Techniques in Credit Card Fraud Detection," no. December, pp. 1–67, 2018.

[11] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M. S. Hacid, and H. Zeineddine, "An Experimental Study With Imbalanced Classification Approaches for Credit Card Fraud Detection," *IEEE Access*, vol. 7, pp. 93010–93022, 2019, DOI: 10.1109/ACCESS.2019.2927266.

[12] X. Zhang, Y. Han, W. Xu, and Q. Wang, "HOBA: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture," *Inf. Sci. (NY).*, no. May, 2019, DOI: 10.1016/j.ins.2019.05.023.

[13] Y. Xie, G. Liu, R. Cao, Z. Li, C. Yan, and C. Jiang, "A Feature Extraction Method for Credit Card Fraud Detection," *Proc. - 2019 2nd Int. Conf. Intell. Auton. Syst. ICI.....v.vv.v.vv.v.v..AS 2019*, no. February, pp. 70–75, 2019, DOI: 10.1109/ICoIAS.2019.00019.

[14] Y. Lucas *et al.*, "Towards automated feature engineering for credit card fraud detection using multi-perspective HMMs," *Futur. Gener. Comput. Syst.*, vol. 102, pp. 393–402, 2020, DOI: 10.1016/j.future.2019.08.029.

[15] G. Ke *et al.*, "LightGBM : A Highly Efficient Gradient Boosting Decision Tree," no. Nips, pp. 1–9, 2017.

[16] A. K. Agarwal, S. Wadhwa, and S. Chandra, "Diagnosis of tuberculosis--newer tests.," *J. Assoc. Physicians India*, vol. 42, no. 8, p. 665, 1994.

[17] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and C. Jiang, "Random forest for credit card fraud detection," *ICNSC 2018 - 15th IEEE Int. Conf. Networking, Sens. Control*, pp. 1–6, 2018, DOI: 10.1109/ICNSC.2018.8361343.

[18] https://github.com/rzhou1/FraudDetection.

[19] https://en.wikipedia.org/wiki/Comma-separated_values

[20] https://christophm.github.io/interpretable-ml-book/feature-importance.html

[21] N. Jameel and H. S. Abdullah, "Intelligent Feature Selection Methods : A Survey," vol. 39, no. 01, pp. 175–183, 2021.

Received 24/4/2021; Accepted 18l7l2021

DOI: https://doi.org/10.33103/uot.ijccce.21.3.4

[22] J. Brownlee, "Imbalanced Classification with Python," *Mach. Learn. Mastery*, p. 463, 2020.

Received 24/4/2021; Accepted 18l7l2021

 ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*