International Journal of Information Retrieval Research
Volume 12 • Issue 2

Credit Card Fraud Prediction Using XGBoost:

An Ensemble Learning Approach

Krishna Kumar Mohbey, Central University of Rajasthan, India*
https://orcid.org/0000-0002-7566-0703

Mohammad Zubair Khan, Taibah University, Saudi Arabia
https://orcid.org/0000-0002-2409-7172

Ajay Indian, Central University of Rajasthan, India

ABSTRACT

With the development of technology, the internet and eCommerce online payment has become an essential mode of payment. Nowadays, credit card payment is a convenient mode of payment online as well as offline transactions. As online credit card payment increases, fraud transactions are likewise increasing day by day. Increasing fraud transactions in the online payment system became a more significant challenge for banks, companies, and researchers. Therefore, it is essential to have an efficient methodology to detect fraud transactions while payment has completed via credit card. Although many traditional approaches are already available for fraud transaction prediction, existing methods lack accuracy, and it can be increased by ensemble techniques such as XGBoost. In this paper, the authors use an ensemble approach that is XGBoost (eXtreme gradient boosting) for credit card fraud prediction. The results are compared with existing machine learning approaches.

KEYWORDS

Credit Card Fraud Prediction, Decision Tree, Ensemble Learning, Machine Learning, Random Forest, SVM, XGBoost

1. INTRODUCTION

With the exponential growth of technology, the internet, and the electronic market, credit card prediction has become an exciting research topic. Online transactions also play a vital role in the present scenario. Electronic commerce also employs Big data and artificial intelligence techniques to gain profits in their business (Maes et al., 2002; Niu et al., 2019). While most transactions and companies are running online, payment transactions are also online and generate massive data. Online payments are made using credit cards, debit cards, net banking, and UPI options. The online credit card payment system needs a predictive model to distinguish whether a transaction is a fraud or not (Maes et al., 2002). Although numerous modern prediction models are proposed to detect credit card fraud transactions, accuracy is still challenging (Niu et al., 2019; Odegua, 2020). The existing model lacks inefficiency because of the hugeness of transaction data and data imbalance problems. Imbalanced data refers to whether one class has more instances than another class category. It leads

DOI: 10.4018/IJIRR.299940

*Corresponding Author

to the class imbalance problem (Divakar & Chitharanjan, 2019; Zhu et al., 2017). Another Challenge is a significant transaction because a large data set has increased heavy-tailed noise distribution and nonlinear patterns. Therefore, traditional approaches are not able to gain higher accuracy (Petropoulos et al., 2019). The preliminary study found that supervised and unsupervised learning techniques have been used to uncover credit card fraud and forecasting (Randhawa et al., 2018). Chan et al. recommended a cost model for fraud detection (Chan et al., 1999). They have combined multiple fraud detectors in the cost model and demonstrated that the loss had been reduced due to distributed data mining of fraud models. A fraud detection model is suggested by Bolton et al. using user behavior for a credit card transaction (Bolton & Hand, 2001). Zojaji et al. review various techniques, data set, and evaluation criteria for credit card fraud transactions (Zojaji et al., 2016).

Figure 1. Process of credit card fraud discovery

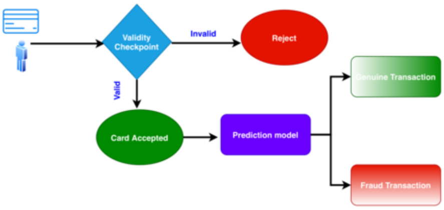


Figure 1 shows the process of credit card fraud discovery. This process described those various essential conditions that are checked before completing the transaction. It includes enough balance, card number, expiry valid, PIN details, etc. According to the provided details prediction model, it classifies transactions as fraud or genuine. This paper investigated the analysis of credit card fraud prediction using Gradient Boosting algorithms XGBoost (Chen & Guestrin, 2016). In addition, we perform an extensive comparison of the prediction accuracy of the XGBoost approach with other machine learning approaches. We also presented evaluation matrices, such as accuracy, precision, recall, and F1 score.

Further, the paper is organized into four sections: Section 2 provides a relevant overview of credit card fraud discovery. Section 3 explains the selected models. In section 4, we have discussed our outcomes and discoveries, and section 5 concludes.

2. RELATED WORKS

Credit card fraud prediction is a huge problem for financial markets and banking systems. Various accurate and robust methods have been developed in recent years to identify credit card fraud transactions. Existing models use different statistical techniques, data mining algorithms, machine learning approaches, and deep learning to detect credit card fraud transactions (Randhawa et al., 2018). The linear regression model (Avery et al., 2004) and the probit model (Mizen & Tsoukas, 2012) were earlier used for credit card fraud detection. However, these models are not able to handle the no-linear dynamics of data. Another statistical model is known as the hazard rate model used for credit card rating. These models use the probability concept to detect defaulter customers from loan or portfolio data (Chava & Jarrow, 2004). Huang (2011) proposed a Gaussian process with a support vector machine (SVM) Vapnik, (1999) to detect credit card fraud transactions. This process requires high computation and has high complexity.

Shrivastava et al. (2008) used a Hidden Markov Model to detect credit card fraud transactions from the dataset. SVM and random forest were applied along with logistic regression to detect credit card fraud from international credit card transaction data Bhattacharyya et al., (2011). Decision tree-based methods are also employed by Sachin et al. (2013) to detect credit card fraud from real-life datasets. The problems of Class imbalance, concept drift, and verification latency were considered by Pozzalo et al. for credit card fraud detection Dal Pozzolo et al., (2017).

Bigdata technologies, Casandra, Spark, and Kafka, were used by Carcillo et al. (2018) in scalable real-time fraud finder (SCARFF) to detect fraud transactions. This model works in a real-time scenario and can handle a considerable volume of data. Different machine learning and deep learning methods were used by Addo et al. (2018) for credit risk scoring. They build a binary classification model to predict defaulters from massive data sets.

Six primary commercial bank data from January 2009 to December 2013 are used by Butaru et al. (2016) to detect credit risk. They have used consumer tradeline, credit bureau, and macroeconomic parameters to predict crime using classifiers such as decision tree, logistic regression, and random forest. This result suggested that a more customized approach is needed for financial institutions that include more features for prediction.

In recent years, deep learning approaches have encouraged outcomes in many fields, such as image processing, classification, and predictions Wang et al., (2015). These approaches are also appropriate to predict credit card fraud. Jurgovsky et al. (2018) formulated the credit card fraud prediction for a sequence classification problem. It is based on a supervised learning concept. They have used the LSTM (Long Short-Term Memory) approach to detect credit card fraud transactions. Roy et al. (2018) proposed a framework of deep learning with various tuning parameters to detect credit card fraud.

Seera et al. suggested using publicly accessible and real transaction records and employed 13 statistical and machine learning approaches for payment card fraud detection. The findings of the original and aggregated features are compared and examined Seera et al., (2021). The majority of data scientists regard credit risk assessment as a classification challenge, and they believe it helps uncover hidden patterns in credit scoring data analysis. It also aids specialists in improving their understanding of credit risk assessment. A wide range of machine learning algorithms has been used to demonstrate risk assessment systems in this field. Classification, ensemble classifiers, and hybrid classifiers are the three types of classification strategies Tripathi et al., (2021).

Although, there are multiple studies carried out for credit card fraud detection, like cardholder's behavior prediction, enhancing the system's processing time, etc. We are motivated by all the above works. We have used the XGBoost ensemble learning technique to improve the accuracy of the credit card fraud prediction system. It can create threads for parallel processing, have a cross-validation method, and handle missing values Mishra & Pandey, (2021). It reduces computation time and allocates memory resources efficiently. Because it was built and developed solely for the goal of model performance and computational speed, XGBoost has proved to push the limits of processing power for boosted trees algorithms. It was designed specifically for tree boosting techniques to use all available memory and hardware resources fully. Its implementation includes several additional features for model tuning, computing environments, and algorithm improvement. Both XGBoost and gradient boosting machines are ensemble tree approaches that employ the gradient descent architecture to boost weak learners. XGBoost, on the other hand, improves the base gradient boosting framework through system optimization and algorithm development Chen & Guestrin, (2016). We explore the XGBoost approach to gain the highest efficiency in large data sizes.

3. SELECTED MODELS

This section discusses various models for credit card fraud predictions. Our forecast and classification problem has considered Naïve Bayes, Support vector machine, Logistic regression, Random forest, Decision tree, and XGBoost Pandey et al., (2021).

3.1 Naïve Bayes Classifier

It is a famous machine learning model based on the Bayes theorem of conditional probability Rane & Kumar, (2018). It passes each transaction and calculates the posterior probability for available classes. According to the highest probability, the transaction is allocated to a class. Naïve Bayes classification is described using Equation (1):

$$\mathrm{P}\left(C / x\right) = \left(\mathrm{P}\left(x / C\right).P\left(C\right)\right) / \left(P\left(x\right)\right) \qquad \left(1\right)$$

Where,

C: specified class
x: transaction used for classification
$P\left(C\right)$ and $P\left(x\right)$: prior probabilities
$\mathrm{P}\left(C / x\right)$: posterior probability

Given the Naïve assumption, which says that a data point X={x_1, x_2, x_3, ….., x_i}, the possibility of each of its features taking place in a given class, the Equation can be rewritten as:

$$\mathrm{P}\left(C / x\right) = P\left(C\right).P\left(x_i / C\right) \qquad \left(2\right)$$

In this experiment, the following parameters (Table 1) are tuned for the Naïve Bayes classifier to get the results.

Table 1. Naïve Bayes classifier parameters

Parameter	Value
priors	None
var_smoothing	1e-09

3.2 SVM Classifier

To solve regression and classification problems, Vapnik first introduced a support vector machine Cortes & Vapnik, (1995). This classifier is popular because of its high reliability, varied applications, and less vulnerability for the overfitted model Vapnik, (2006). It derives the finest hyperplane that maximizes the margin between two classes. It can be used for linear and nonlinear datasets. It can discover a nonlinear decision boundary by projecting the data with a nonlinear function to space with a higher dimension.

We traverse linearly separable classes using two-class problems. For a given Dataset S as (P1, Q1), (P2, Q2),……….(P|S|, Q|S|), where Qj is the class label whose value is from +1 to -1 (Qj Î (-1, +1)). Qj is associated with the Pj set of training tuples.

A hyperplane can be described as a P set of points satisfying

$$W.P - B = 0 \qquad (3)$$

Where W is a normal vector to the hyperplane and $\frac{B}{\|W\|}$ is the offset of the plane from the origin and average vector W. The best line, which has the least classification error, in general. Best hyperplane by the maximum distance of the hyperplane to the closest of the negative instance and positive instance.

Figure 2. Support vector machine Classifier

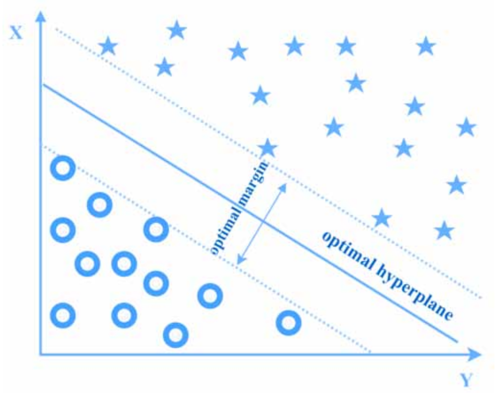


Figure 2 displays SVM optimal hyperplane in training with sample transactions to classify credit card fraud transactions, and not fraud transactions are represented as disk-shaped Kumar et al., (2019). In this experiment, the following parameters (Table 2) are tuned for the SVM Classifier to get the results.

Table 2. SVM classifier parameters

Parameter	Value
alpha	0.0001
average	False
class_weight	None,
early_stopping	False
epsilon	0.1
eta0	0.0
fit_intercept	True,
l1_ratio	0.15
learning_rate	‘optimal’
loss	‘hinge’
max_iter	1000
n_iter_no_change	5
n_jobs	None

Table 2 continued on next page

Table 2 continued

Parameter	Value
penalty	‘l2’
power_t	0.5
random_state	None
shuffle	True
tol	0.001
validation_fraction	0.1
verbose	0
warm_start	False

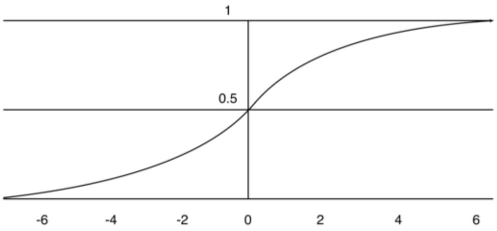
3.3 Logistic Regression

Logistic regression is the most popular classification model in machine learning. In this model probability of each class is calculated for prediction. This model predicts output by combining the input variable with weight. Suppose *x* is an independent variable and *y* is a dependent variable, then linear regression can be represented as

$$y = a_0 + a_1 * x \qquad (4)$$

Where a_0 is a bias term and a_1 is the weight for input variable *x*. If a logistic function is used for logistic regression, it predicts each class’s probability and will use a sigmoid or logistic function Shakya, (2018). Figure 3 displays an example of the sigmoid function.

Figure 3. Sigmoid function



This function predicts values between 0 and 1, and it can be expressed as

$$sigm\left(z\right) = \frac{1}{1 + e^{-z}} \qquad \left(5\right)$$

If we have an independent variable (*x*) and a dependent variable (*y*), then logistic regression can be stated as

$$P\left(y = 1\right) = sigm(a_0 + a_1 * x) \qquad \left(6\right)$$

Where a_0 and a_1 are the parameters of logistic regression model and learned while the training process. The predicted value can be expressed as (if threshold =0.5)

y=1 if P(y=1)>=0.5
y=0 if P(y=1) <0.5

In this experiment, the following parameters (Table 3) are tuned for the Logistic regression approach to get the results.

Table 3. Parameters used in Logistic regression classifier

Parameter Name	Value
C	1.0
class_weight	None
dual	False
fit_intercept	True
intercept_scaling	1
l1_ratio	None
max_iter	100
multi_class	'auto'
n_jobs	None
penalty	'l2'
random_state	None
solver	'lbfgs'
tol	0.0001
verbose	0
warm_start	False

3.4 Random Forest Classifier

This classifier is mainly used for classification and regression problems. It uses ensemble learning methods for prediction by averaging the prediction of several independent base models. The average of forecasts combined with several random decision trees Breiman, (2001). It shows excellent performance while more features are selected. However, it can be used for significant scale problems also. While working on the classification, random forest classifier *m* can be gained via the majority of votes among *K* classification trees with input *x* as

$$m\left(x : C_1, C_2, \ldots..C_K\right) = \left\{ \begin{matrix} 1 \ if \dfrac{1}{K}\sum_{i=1}^{K} m\left(x; C_i\right) > \dfrac{1}{2} \\ 0 \qquad\qquad otherwise \end{matrix} \right\} \qquad (7)$$

In this experiment, C is a parameter set value. The following parameters (Table 4) are tuned for the Random Forest classifier approach to get the results in this experiment.

Table 4. Parameters used in Random Forest classifier

Parameter Name	Value
bootstrap	True
ccp_alpha	0.0
class_weight	None
criterion	‘gini’
max_depth	None
max_features	‘auto’
max_leaf_nodes	None
max_samples	None
min_impurity_decrease	0.0
min_impurity_split	None
min_samples_leaf	1
min_samples_split	2
min_weight_fraction_leaf	0.0
n_estimators	100
n_jobs	None
oob_score	False
random_state	None
verbose	0
warm_start	False

3.5 Decision Tree Classifier

It is a top-down method in which the root node produces a binary split up to a specific criterion. It is also used to solve classification problems in various domains. In the decision tree, internal nodes denote the test conditions, whereas leaf nodes denote class categories. Each produced terminal node indicates a predicted class. A decision tree creates a sequence of rules to identify a class of test transactions or instances according to classes. In this model, overfitting is handled by a post pruning strategy Nithyassik & Nandhini, (2010). The following parameters (Table 5) are tuned for the Decision Tree classifier approach to get the results in this experiment.

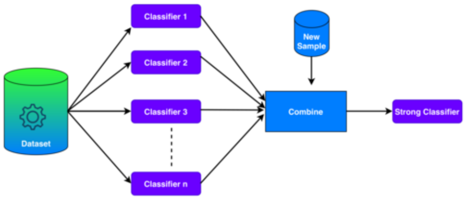
Table 5. Parameters used in Decision tree classifier

Parameter	Value
ccp_alpha	0.0
class_weight	None
criterion	'gini'
max_depth	None
max_features	None
max_leaf_nodes	None
min_impurity_decrease	0.0
min_impurity_split	None
min_samples_leaf	1
min_samples_split	2
min_weight_fraction_leaf	0.0
presort	'deprecated'
random_state	None
splitter	'best'

3.6 XGBoost Approach

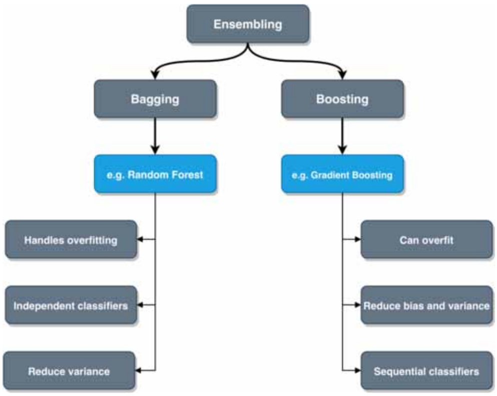
XGBoost stands for eXtreme Gradient Boosting used for classification, regression, and problems related to ranking. This approach is an advanced version of the Gradient boosting approach Chen & Guestrin, (2016), which improves the results compared to the Gradient boosting approach. It uses ensemble techniques to improve results. Ensemble techniques are used to modify existing classification models to handle imbalanced class distributions. In ensemble learning, classification problems are solved by training multiple learners. Its central concept is to combine numerous weak learners into keen learners to boost the classifier's performance. Figure 4 shows the idea of ensemble learning techniques.

Figure 4. Concept of ensemble learning technique



Ensemble techniques can be categorized as bagging and boosting. It is shown in figure 5.

Figure 5. Ensemble techniques



Bagging (Bootstrap Aggregation) is an ensemble technique. In this technique, various training samples are generated from original training sets known as a bootstrap sample, and this process is called bootstrapping. For prediction, each bootstrap sample is trained for an individual model. Final prediction calculates as aggregated by averaging in regression or voting in the classification of all bootstrap models.

Boosting is also an ensemble technique. It combines various weak learners to build a keen learner that provides improved results compared to individual learners. In each boosting step, weak learners are sequentially trained and correct their predecessor by adding weights to previously misclassified samples. In boosting, the technique bootstrapping is used to avoid variance and overfitting. XGBoost model use boosting techniques for classification and predictions.

XGBoost adopts a more generalized method to control overfitting and contributes to improving the results. XGBoost works on parallel computing; hence it is fast as compared to the standard Gradient boosting approach. It can handle missing data and includes cross-validation features used to determine to boost round in each run. XGBoost needs a few parameters to tune for getting better results. In this experiment, the following parameters (Table 6) are tuned for the XGBoost approach to get the results.

Table 6. Parameters used in XGBoost classifier

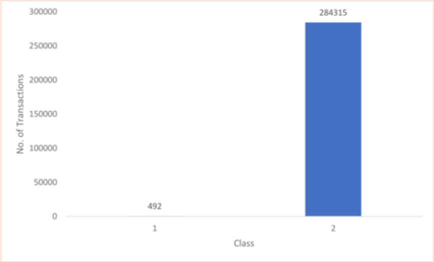
Parameter Name	Value
learning_rate	0.1
n_estimators	1000
max_depth	5
min_child_weight	1
gamma	0
subsample	0.8
colsample_bytree	0.8
objective	‘binary:logistic’
nthread	4
scale_pos_weight	1

4. THE EMPIRICAL ANALYSIS OF CREDIT CARD PREDICTION

4.1 Data Source and Description

This paper uses a public credit card transaction dataset of European cardholders Dal Pozzolo et al., (2015), and it was also released in the Kaggle[1] community. This dataset was made in September 2013. It comprises 284,807 transactions with only 492 transactions as a fraudulent class. It is a highly skewed dataset as the positive class contributes only 0.172% of the entire dataset. This unbalanced class distribution is presented in Figure 6.

Figure 6. Unbalance class distribution



This dataset has a total of 30 features. 28 features represented as V1 to V28 obtained after principal component analysis transformation and two features time and amount without PCA transformation. Table 7 shows the description of the dataset features.

Table 7. Dataset feature description

Features	Data types	Description
Time	Integer	Time difference between each transaction (second)
V1	Double	1 principle component
V2	Double	2 principle component
…	…	…
…	…	…
V28	Double	28 principle component
Amount	Double	Transaction amount
Class	Integer	1=fraud, 0=not fraud

4.2 Performance Assessment

Precision, recall, accuracy, AUC, and f-measure are utilized to evaluate the performance of the models. Table 8 shows the confusion matrix for assessment.

Table 8. Confusion matrix

	Actual Value		
Predicted Value		Positive	Negative
	Positive	**True Positive (TP)**	**False Positive (FP)**
	Negative	**False Negative (FN)**	**True Negative (TN)**

Table 9 presents the various performance measures with their definitions.

Table 9. Performance evaluation measures

Matric Name	Definition
Accuracy	$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$
Precision	$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$
Recall	$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$
F-Measure	$\text{F} - \text{measure} = \frac{2\text{*Precision*Recall}}{\text{Precision} + \text{Recall}}$
AUC	$AUC = \left(Recall - \frac{FP}{FP + TN} + 1\right) / 2$

4.3 Result Analysis and Discussion

This section describes the experimental results of the different models simulated on the credit card transaction dataset. After cleaning the transactional dataset, we have applied the normalization and data standardization process. The complete dataset is partitioned into a training dataset (70%) and testing datasets (30%) for the simulation study. The training dataset builds the models, and the testing dataset evaluates the performance of these models. In this paper, all experimentations are conducted using the Python programming language.

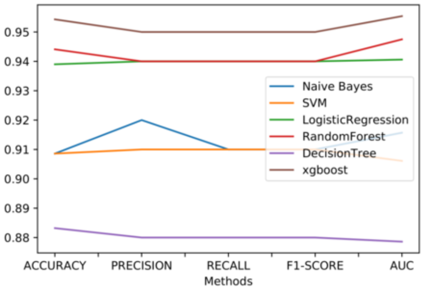
Firstly, the credit card transaction dataset is trained using different models such as Naïve Bayes, SVM, random forest, decision tree, logistic regression, and XGBoost; performances are evaluated. Table 10 shows the experimental results of the classification task.

Table 10. Performance comparison

	Accuracy	Precision	Recall	F1-Score
Naive Bayes	0.8934	0.90	0.90	0.89
SVM	0.9390	0.94	0.94	0.94
Logistic Regression	0.9443	0.94	0.95	0.94
RandomForest	0.9341	0.93	0.94	0.94
DecisionTree	0.9340	0.92	0.92	0.92
Xgboost	0.9644	0.96	0.97	0.96

Different models' performance is assessed using different measures such as accuracy, precision, recall, and F2-score. The performance of various machine learning methods is compared with the ensemble learning-based XGBoost model. From the analysis of table 4, the accuracy of the XGBoost model is higher than in other traditional models. It indicates that the ensemble learning-based approach, such as XGBoost, performs better prediction over other traditional approaches. Figure 7 represents the comparative graphical results of these models.

Figure 7. Comparative results graph



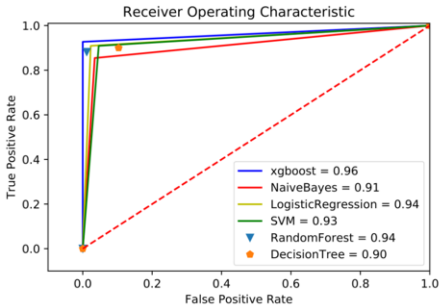
Another comparison made by the AUC (Area Under Curve) values, which are generated from precision and recall, is also higher in the XGBoost model. This comparison is depicted in table 11.

Table 11. AUC Results comparison

Model	AUC
Naive Bayes	0.9021
SVM	0.9382
Logistic Regression	0.9578
RandomForest	0.9430
DecisionTree	0.9349
Xgboost	0.9669

According to the true positive and false positives, ROC (Receiver operating characteristic) graph is also prepared for result evaluations. This ROC graph is depicted in figure 8.

Figure 8. ROC comparison



In conclusion, the ensemble techniques-based model, such as XGBoost used in this paper, has a better performance than state-of-the-art machine learning models. Therefore, the XGBoost model is a valuable model for credit card fraud prediction.

5. CONCLUSION

In this study, a comparative study is conducted among the ensemble technique-based XGBoost model and other traditional machine learning models to classify a credit card transaction as a fraud or not. Our study significantly reduces the risks of financial losses and the uncertainty that institutions encounter in their day-to-day operations. Our research includes a benchmark credit card transaction dataset to measure performances. Performance evaluation was conducted using machine learning and ensemble methodologies. For this, a publicly available credit cardholder's dataset is used. XGBoost produces better performance in the ensemble model's prediction accuracy, precision, recall, and AUC values. Considering the poor performance of traditional models in dealing with unbalanced data, the XGBoost model uses ensemble techniques to improve performance, gain prediction accuracy up to 96%, and outperform other traditional models. This research is significant in light of the tremendous growth of e-commerce operations, resulting in the exponential growth of online transactions in this digital era. Consumers are turning to internet purchasing as a result of the Covid-19 epidemic. As a result, a financial institution's ability to limit the risk of fraud transactions depends on the efficient system for the fraud detection system. Various models have dealt with credit card fraud predictions recently, including machine learning, AI, and deep learning approaches. However, there is still room to develop more accurate and efficient models. The current study can be improved from several perspectives. Hybrid models can be built by combining two or more models to improve the research. Furthermore, the detection algorithms' online implementation will be examined.

FUNDING AGENCY

Publisher has waived the Open Access publishing fee.

REFERENCES

Addo, P. M., Guegan, D., & Hassani, B. (2018). Credit risk analysis using machine and deep learning models. *Risks*, *6*(2), 38. doi:10.3390/risks6020038

Avery, R. B., Calem, P. S., & Canner, G. B. (2004). Consumer credit scoring: Do situational circumstances matter? *Journal of Banking & Finance*, *28*(4), 835–856. doi:10.1016/S0378-4266(03)00202-4

Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems*, *50*(3), 602–613. doi:10.1016/j.dss.2010.08.008

Bolton, R. J., & Hand, D. J. (2001). Unsupervised profiling methods for fraud detection. *Credit Scoring and Credit Control, 7*, 235-255.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. doi:10.1023/A:1010933404324

Butaru, F., Chen, Q., Clark, B., Das, S., Lo, A. W., & Siddique, A. (2016). Risk and risk management in the credit card industry. *Journal of Banking & Finance*, *72*, 218–239. doi:10.1016/j.jbankfin.2016.07.015

Carcillo, F., Dal Pozzolo, A., Le Borgne, Y. A., Caelen, O., Mazzer, Y., & Bontempi, G. (2018). Scarff: A scalable framework for streaming credit card fraud detection with spark. *Information Fusion*, *41*, 182–194. doi:10.1016/j.inffus.2017.09.005

Chan, P. K., Fan, W., Prodromidis, A. L., & Stolfo, S. J. (1999). Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems & their Applications*, *14*(6), 67–74. doi:10.1109/5254.809570

Chava, S., & Jarrow, R. A. (2004). Bankruptcy prediction with industry effects. *Review of Finance*, *8*(4), 537–569. doi:10.1093/rof/8.4.537

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794). doi:10.1145/2939672.2939785

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. doi:10.1007/BF00994018

Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2017). Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(8), 3784–3797. PMID:28920909

Dal Pozzolo, A., Caelen, O., Johnson, R. A., & Bontempi, G. (2015, December). Calibrating probability with undersampling for unbalanced classification. In *2015 IEEE Symposium Series on Computational Intelligence* (pp. 159-166). IEEE. doi:10.1109/SSCI.2015.33

Divakar, K., & Chitharanjan, K. (2019). Performance evaluation of credit card fraud transactions using boosting algorithms. *Int. J. Electron. Commun. Comput. Eng. IJECCE*, *10*(6), 262–270.

Huang, S. C. (2011). Using Gaussian process based kernel classifiers for credit rating forecasting. *Expert Systems with Applications*, *38*(7), 8607–8611. doi:10.1016/j.eswa.2011.01.064

Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P. E., He-Guelton, L., & Caelen, O. (2018). Sequence classification for credit-card fraud detection. *Expert Systems with Applications*, *100*, 234–245. doi:10.1016/j.eswa.2018.01.037

Kumar, S., Koolwal, V., & Mohbey, K. K. (2019). Sentiment analysis of electronic product tweets using big data framework. *Jordanian Journal of Computers and Information Technology*, *5*(1), 43–59. doi:10.5455/jjcit.71-1546924503

Maes, S., Tuyls, K., Vanschoenwinkel, B., & Manderick, B. (2002, January). Credit card fraud detection using Bayesian and neural networks. In *Proceedings of the 1st international naiso congress on neuro fuzzy technologies* (pp. 261-270). Academic Press.

Mishra, K. N., & Pandey, S. C. (2021). Fraud Prediction in Smart Societies Using Logistic Regression and k-fold Machine Learning Techniques. *Wireless Personal Communications*, *119*(2), 1–27. doi:10.1007/s11277-021-08283-9

Mizen, P., & Tsoukas, S. (2012). Forecasting US bond default ratings allowing for previous and initial state dependence in an ordered probit model. *International Journal of Forecasting*, *28*(1), 273–287. doi:10.1016/j.ijforecast.2011.07.005

Nithyassik, B., & Nandhini, D. E. C. (2010). Classification techniques in education domain. *International Journal on Computer Science and Engineering*, *2*(5), 1647–1684.

Niu, X., Wang, L., & Yang, X. (2019). A comparison study of credit card fraud detection: Supervised versus unsupervised. arXiv preprint arXiv:1904.10604.

Odegua, R. (2020). Predicting Bank Loan Default with Extreme Gradient Boosting. arXiv preprint arXiv:2002.02011.

Pandey, A., Shukla, S., & Mohbey, K. K. (2021). Comparative Analysis of a Deep Learning Approach with Various Classification Techniques for Credit Score Computation. *Recent Advances in Computer Science and Communications*, *14*(9), 2785–2799. Advance online publication. doi:10.2174/2666255813999200721004720

Petropoulos, A., Siakoulis, V., Stavroulakis, E., & Klamargias, A. (2019). A robust machine learning approach for credit risk analysis of large loan level datasets using deep learning and extreme Gradient boosting. *IFC Bulletins Chapters, 49*.

Randhawa, K., Loo, C. K., Seera, M., Lim, C. P., & Nandi, A. K. (2018). Credit card fraud detection using AdaBoost and majority voting. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 14277–14284. doi:10.1109/ACCESS.2018.2806420

Rane, A., & Kumar, A. (2018, July). Sentiment classification system of twitter data for US airline service analysis. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)* (Vol. 1, pp. 769-773). IEEE. doi:10.1109/COMPSAC.2018.00114

Roy, A., Sun, J., Mahoney, R., Alonzi, L., Adams, S., & Beling, P. (2018, April). Deep learning detecting fraud in credit card transactions. In *2018 Systems and Information Engineering Design Symposium (SIEDS)* (pp. 129-134). IEEE. doi:10.1109/SIEDS.2018.8374722

Sahin, Y., Bulkan, S., & Duman, E. (2013). A cost-sensitive decision tree approach for fraud detection. *Expert Systems with Applications*, *40*(15), 5916–5923. doi:10.1016/j.eswa.2013.05.021

Seera, M., Lim, C. P., Kumar, A., Dhamotharan, L., & Tan, K. H. (2021). An intelligent payment card fraud detection system. *Annals of Operations Research*, 1–23. doi:10.1007/s10479-021-04149-2 PMID:34121790

Shakya, R. (2018). *Application of Machine Learning Techniques in Credit Card Fraud Detection* (Doctoral dissertation). University of Nevada, Las Vegas, NV.

Srivastava, A., Kundu, A., Sural, S., & Majumdar, A. (2008). Credit card fraud detection using hidden Markov model. *IEEE Transactions on Dependable and Secure Computing*, *5*(1), 37–48. doi:10.1109/TDSC.2007.70228

Tripathi, D., Edla, D. R., Bablani, A., Shukla, A. K., & Reddy, B. R. (2021). Experimental analysis of machine learning methods for credit score classification. *Progress in Artificial Intelligence*, 1-27.

Vapnik, V. (2006). *Estimation of dependences based on empirical data*. Springer Science & Business Media. doi:10.1007/0-387-34239-7

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, *10*(5), 988–999. doi:10.1109/72.788640 PMID:18252602

Wang, L., Liu, T., Wang, G., Chan, K. L., & Yang, Q. (2015). Video tracking using learned hierarchical features. *IEEE Transactions on Image Processing*, *24*(4), 1424–1435. doi:10.1109/TIP.2015.2403231 PMID:25700445

Zhu, B., Baesens, B., & vanden Broucke, S. K. L. M. (2017). An empirical comparison of techniques for the class imbalance problem in churn prediction. *Information Sciences*, *408*, 84–99. doi:10.1016/j.ins.2017.04.015

Zojaji, Z., Atani, R. E., & Monadjemi, A. H. (2016). *A survey of credit card fraud detection techniques: data and technique oriented perspective.* arXiv preprint arXiv:1611.06439.

ENDNOTE

[1] https://www.kaggle.com/mlg-ulb/creditcardfraud/data

Krishna Kumar Mohbey is an Assistant Professor of Computer Science at Central University of Rajasthan, India. He received his Bachelor's degree in Computer Application from MCRPV Bhopal (2006), Master's in Computer Application from Rajiv Gandhi Technological University Bhopal (2009) and PhD from Department of Mathematics and Computer Applications from National Institute of Technology Bhopal, India (2015). His areas of interest are data mining, mobile web services, big data analysis and user behavior analysis.

Ajay Indian received the M.C.A. degree in 2001 from Dr. B.R. Ambedkar University, University Campus, Agra, India, M.Tech. (Computer Science & Engineering) in 2010, Gautam Buddha Technical University, Lucknow, India, received a Ph.D. (Computer Science) in 2019 from the Department of Computer Science, Faculty of Technology, Gurukula Kangri Vishwavidyalaya Haridwar, India. He has also qualified GATE (CSIT) and UGC-NET (C.S.) in 2010 and 2013, respectively, and has more than 19 years of teaching and research experience. Currently, he works as an Assistant Professor in the Department of Computer Science, Central University of Rajasthan, Ajmer, India. Earlier, he worked as an Associate Professor in Computer Science, Invertis University, Bareilly, India. He has also worked as an Assistant Professor in the Department of Computer Science & Information System, University of Jazan (U.S. pattern-based University), Jazan States, K.S.A. from 2008 to 2009. His main research work focuses on Image Processing, Artificial Neural Networks, and Data Mining. He has published around 20 research papers in International Journals of repute and National/ International conference proceedings. Seven M.Tech. Degrees have been awarded under his supervision.

M. Zubair Khan got the Ph.D. degree in computer science and information technology from Faculty of Engineering, M.J.P. Rohilkhand University, Bareilly India, and the master of technology in computer science and engineering from U.P. Technical University, Lucknow, India. He is currently working as senior faculty member in the Department of Computer Science, Deanship of Academic Services, Taibah University. Past he has worked as head and associate professor, in the Department of Computer Science and Engineering, Invertis University, Bareilly India. He has published more than 36 journals and conference papers. Dr. M. Zubair Khan is a member of Computer Society of India since 2004. His current research interests are data mining, big data, parallel and distributed computing, theory of computations, and computer networks. He has more than 13 years teaching and research experience.