International Journal of Advanced Computing and Mechanical Systems (IJACM)
E-ISSN No.: xxxx-xxxx www.ijacm.com
Volume Number: 2 Issue Number: 3 pp: 11 - 19 Publication Month: March-2026

Predicting Credit Card Fraud Detection Using Machine Learning

S Shilpa[1], Asgar Ali[2], D Niranjan[2], Saurabh Suman[2], Wagh Apeksha[2], A Yakshitha[2]
[1]Assistant Professor, Department of CSE, Siddartha Institute of Science and Technology, Puttur, Andhra Pradesh, India
[2]UG Students, Department of CSE, Siddartha Institute of Science and Technology, Puttur, Andhra Pradesh, India

Autor1 E-Mail: shilpamani0614@gmail.com
Autor2 E-Mail: asgaralisirsa@gmail.com
Autor3 E-Mail: devanapalliniranjan@gmail.com
Autor4 E-Mail: saurabhsumanmganj@gmail.com
Autor5 E-Mail: apekshawagh127@gmail.com
Autor6 E-Mail: agraharamyakshitha@gmail.com

ABSTRACT

In the face of a financial security point of view fraud detection systems emerge as key element in as far as the threats of losing money as a cost are concerned. The act of resistance against impersonation in the financial transactions arena has become very important in terms of preserving privacy. This study involves discussion of application of Machine Learning algorithms which includes Decision Tree, Random Forest and XGBoost for the process of detection of fraudulent credit card transactions. These models were selected because they can handle large and complicated data sets and derive hidden patterns that represent fraudulent activity. By making use of such features as transaction history and user behavior, the proposed system was able to show off its exceptional accuracy in differentiating normal or fraudulent transactions. The result for Random Forest model is 99.96% accuracy, the result for Decision tree model is 99.92% and XG boost model result accuracy is 99.95%. These results validate machine learning to continuously detect fraud in financial services. Moreover, continuous retraining of models is important to adapt to changing fraud tactics to make the system effective and scalable in the long term.

Keywords: Financial transactions, Decision Tree, Random Forest, XGBoost, Machine Learning, Fraud detection.

I. INTRODUCTION

With fast digitalization, transaction Credit card fraud has emerged as one of the major challenges in recent years. As online shopping and digital payments increase, criminals are finding more space in the vulnerabilities in payment systems and these risks in the financial sector increase. Credit card fraud leads not only to huge financial loss to the individuals and organizations it is involved in but also leads to distrust of digital means of payment. ML algorithms, such as decision tree, random forest and xgboost have been used in this project in prediction of fraud in credit card industries.

These algorithms are chosen due to their capability to process huge amounts of data and adapt to new patterns of fraud. By making use of data of the transaction and the behaviors of the user, the model learns to identify suspicious examples of fraud. The aim is to create a system capable of detecting fraudulent transactions with high accuracy and reducing the financial loss which it causes, which in turn can help to create a more secure financial system.

doi: https://doi.org/10.65883/ijacm.2026v2i3.02

1.1 Objective of The Study

The aim of the present study is to solve the problem of developing a system for effective detection of fraud on the credit card using machine learning based method. The system will make use of decision tree, random forest, and XG Boost algorithm with the ability to correctly denote if transactions are fraudulent or not. By testing these models, with a large and varied dataset, the paper aims, it is possible to compare the performance of these models with respect to accuracy, speed and scalability, and we may focus on the treatment of high-volume data. The goal would be to have a robust system to detect fraud which can have a direct integration with the current financial infrastructure that would have the least level of disruption to the legitimate pattern of transactions with a minimum occurrence of any fraudulent pattern of transactions. Ultimately, the aim of this research is to bring more security and trust to the field of digital payment since it is an effective solution to financial institutions that can provide security to protect from fraud.

1.2 Problem Statement

Credit card fraud has been skyrocketing due to the increasing numbers of digital payments and is a threat in the consumer as well as financial industry. Detecting Fraud in Financial Transactions is Expensive and Often Result in Customer Trust Loss Machine learning (ML) is a good solution, but fraud detection models will have to deal with big datasets with imbalanced and subtle fraudulent behaviors. This study aims to formulate an efficient ML-based fraud detection scheme that can be capable of identifying the transactions as fraudulent or genuine in lesser amount of time and with improved accuracy while reducing the false positive and increase the security as a way of bringing back the optimisms of the consumers towards the usage of digital payments.

II. LITERATURE SURVEY

Previous research in Credit Card fraud detection has looked at various methodologies, where the model of Transformer has been used for Natural Language Processing, but in this study, Transformer has been used to detect fraud. These models use self-attention mechanism and sequential transaction data to make super detection accuracy and less false positives compared to the traditional machine learning algorithm based on which it can be used for the complex fraud detection tasks [1]. Another innovative method is Asexual Reproduction Optimization (ARO) for Features selection and optimization in combination with Random Forest and XGBoost that brings significant improvement in accuracy and efficiency with less false positive [2]. In the scope of IoT-based fraud detection, a hybrid approach of machine learning and deep learning techniques, which favors the analysis of factors such as the transaction time, location, type of device provided a robust and efficient approach in terms of noisy and incomplete data [3]. Studies are also made on the different architectures of deep learning such as CNN, LSTMs and autoencoders which came up as an option but still have issues for interpretability and need large volume of labelled data [4].

A semi-supervised graph-based methodology based on graph convolutional networks (GCNs) handling this issue of imbalanced data sets has been presented with focus preset on the problem of temporal relations and anomalies in transaction data [5]. A critical look on methodologies that exist to solve the problem of fraud detection brings out problems like overfitting, imbalanced data sets and need for better treatment of temporal dependencies in transactions [6, 7]. Further, application of hybrid machine learning models like combining SVM, Decision Tree, Random Forest, XGBoost and MLP to detect frauds had been also tested where random forest and XGboost predict optimal [8]. Further, the combination method of Iterative hard thresholding logistic regression (IHT-LR), grid search for

doi: https://doi.org/10.65883/ijacm.2026v2i3.02

optimization methods have been proved to help in improving the accuracy as well as computing power of fraud detection [9].

Finally, a distributed deep neural network (DNN) model with federation learning is proposed which enables the sharing of transaction data from other financial institutions for the accurate fraud detection without violating privacy [10]. This paper aims at addressing the gap in the existing literature by proposing ensemble model XGBoost and Decision Tree, Random Forest and that builds models with better detection accuracy and scalability, providing a more efficient way of solving a problem such as fraud detection in financial transactions.

III. PROPOSED SYSTEM WORKFLOW

The idea behind the proposed system is to create efficient and reliable model fraud detection algorithms for credit card transactions like Decision Tree algorithm, Random Forest algorithm and XGBoost algorithm. The conventional fraud detection systems may encounter some problems, such as slow data processing and can only rely on centralized management, which is much delayed. This system deals with these problems by providing an efficient mechanism to find out the accurate fraud cases, which bring in more security for both the constituents, that is, the customer and the bank. The implementation has been done using python packages like Scikit learn, XG boost etc.

The system begins with the pre-processing of the data of the transactions processed by the service providers (banks, payment gateways etc.). Standardization of the numerical features, encoding of the categorical features, dealing with the missing values, etc. Decision Trees are used for the classification problem of fraudulent or the legitimate transactions along with the features such as Merchant type, Transaction amount, User behavior. Random Forest helps to optimize the accuracy of the developed models by overfitting but XGBoost helps to increase performance by using gradient boosting techniques (Figure 1).

IV. METHODOLOGY

4.1 Data Acquisition and Preprocessing

The data set that is being used for fraud detection represents an anonymous financial attribute of fake credit card transactions. It consists of features like transaction time, amount, 28 unidentifiable features (V1 to V28) represent some pattern of types of transaction and customer behavior. These features are unidentifiable for the purpose of privacy of the customers, but these features are quite impressive when it comes to identifying fraud. Target variable: It is the indicator of whether the transaction is fraudulent transaction [1] or not [0]. Fraudulent transaction is a small and unbalanced piece in the data which require special techniques for handling the classes and class imbalance. The data is of such a high scale of volume and complication not to mention the monetary value of the transactions and the behavior patterns of the users. Preprocessing steps like data normalization and feature engineering is important to prepare the data for machine learning models like Decision Tree, Random Forest and XGboost models to accurately detect fraud.

The preprocessing starts by importing the dataset to create Data Frame by using panda's library and some basic exploratory data analysis (EDA), To get an idea about the structure and summary statistics of the data and existence of missing values. The class variable, which identifies fraudulent (1) and legitimate transactions (0) is identifiable. The data set is then split into training and test data in the

doi: https://doi.org/10.65883/ijacm.2026v2i3.02

ratio 70-30 which ensures the stratification of the target data for maintaining the balance of the fraud and non-fraud cases. train_test_split is implemented in which the data will be split and then the machine learning models are trained with them. Hyperparameter tuning is done by using RandomizedSearchCV in the order to choose the best combinations of hyperparameters for the models like Random Forest, Decision Tree and XGBoost. The model evaluation will be done based on accuracy Score which is a measure of the ability of the model to accurately identify the fraudulent transactions. Data cleaning, normalizing and proper splitting is done in such a manner that the fraud and non-fraud predicaments and both are equally represented in the training testing data.

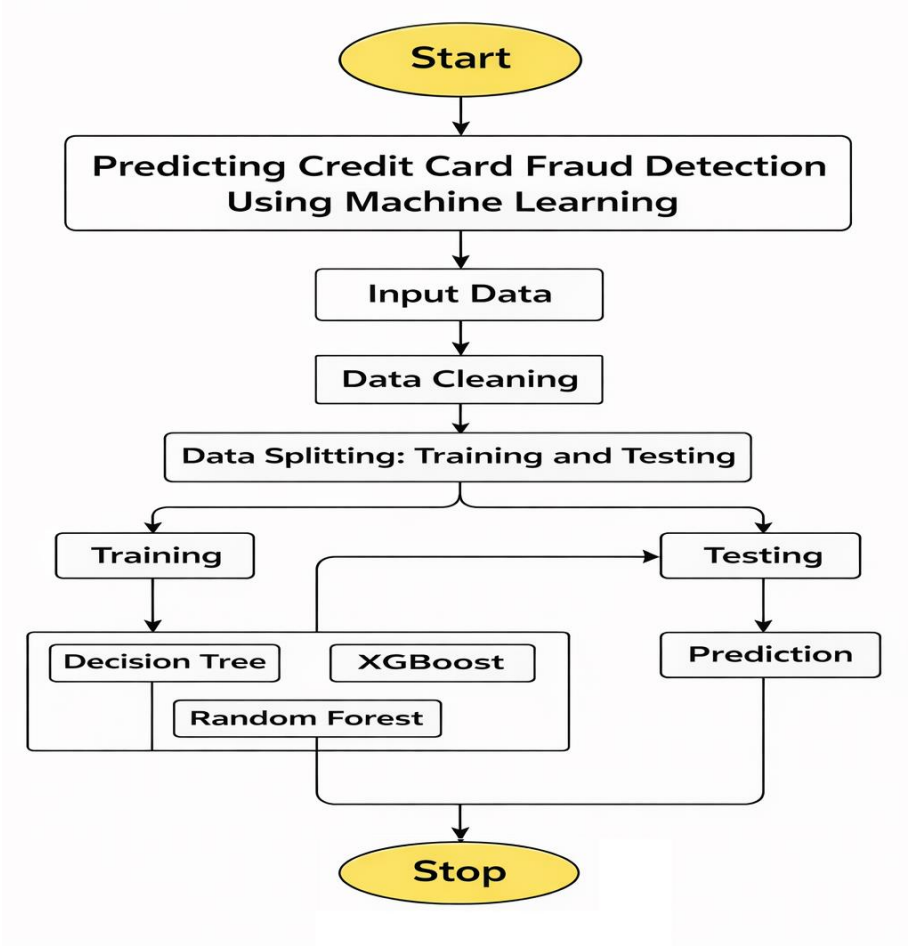

Figure 1: Project Flow

4.2 Model Training

Decision Tree

Since Decision Trees are Supervised learning algorithms, they basically learn through recursively splitting the given Dataset into subsets (based on classifying/regressing feature values) to form a decision tree. Every node represents a decision, which tests some feature. Leaf nodes are training results which are the class labels of classification, and or the regression prediction value. The second useful use of decision trees arises from their simplicity; and they are also very intuitive. To form a decision tree the algorithm has the feature of splitting the data in each specific node. Generally, the criteria used for splitting can be Gini Index, Information Gain, Chi-square. This is where the Gini Index is used so much in the classification problem domain.

doi: https://doi.org/10.65883/ijacm.2026v2i3.02

The formula for calculating the Gini Index that applies to a node is

$$Gini(D) = 1 - \sum_{i=1}^{k} p_i^2 \qquad (1)$$

Where: D is the dataset at that node, P i: A declaration that there is a probability that the data element observed at that node will have a class I, k is the number of classes.

We determine the Gini Index (for all possible split) at a given internal node of tree. The courts, fared out the minimal value of Gini Index for splitting This then continues recursively till a certain depth in the tree This, are reached or the splitting of nodes can be continued no further. The rides you get per transaction on basis of certain test on some of the transaction characteristics such as amount, time, location and user behaviour. Each ER node of the decision tree represents some test on one of the input features; the leaf nodes show the resultant output class: fraud or no fraud. Decision trees are easily interpretable and visualizable, hence, it's useful in determining the key features where into determining fraud. However, decision trees tend to be overfitting; the more depth the tree has the more overfitting there will be. To push back this Giantism, then the techniques of reduction along with group procedures like Random Forest are used.

Random Forest

Random Forest (with its own type of ensemble learning) creates a series of decision trees which can't have any adverse effect on accuracy (through averaging), while avoiding overfitting. Every tree is constructed using a random sub-sample of the data and a random group of features are used to find the best split for each node. Finally, the output of all the trees is combined to form the prediction (majority votes in the case of classification). Random Forests is an ensemble algorithm consisting of bagging: Many models (decision trees in this case) are defined on various random parts of data. The most popular voting scheme for classification is:

$$\hat{y} = \text{majority vote of individual decision trees} \qquad (2)$$

Where: The class label ($\hat{y}$ is the actual (predicted) class label), The prediction of each decision tree will be input to a vote.

Each tree is built on a sample of random bootstrap (i.e. sample of the training set (with replacement)). On the other hand, a potential tree randomness force may be due to the random selection of subset features, for splitting in the node splitting process, so that it also has a power of suppressing the overfitting. Detecting parties for fraud: To detect which are fraud and which is unfraud. Random Forest which are used, it is a merging the decision in a few trees to label the transaction. The Random Forest algorithm deliver the solution to the overfitting problem of a problem which quite popular for single decision tree for the complex looking data. Thus, though the combined decision trees would be less likely to overfit than single decision trees, Random Forests would be of good application for fraud detection as the patterns are highly complex and can vary considerably. It's one of the relevant techniques in the case of getting an imbalanced data set. For example, a very small number of cases is fake transactions relative to the sum of actual transactions, and so fraud detection is the common type of imbalanced classification system. Another interesting feature of the algorithm is that it can cope with missing data, and it does not require intensive pre-processing of the data. It is a computationally efficient model, especially if trained in parallel fashion.

doi: https://doi.org/10.65883/ijacm.2026v2i3.02

Extreme Gradient Boosting or XG Boost

XGBoost is collectively considered as an evolved version of the Techniques of gradient boosting taken from Hence, XGBoost succeeded to become one of the most popular machine learning algorithms owing to its speed and competitive nature. The decision trees are one by one added in the XGBoost. The method item is selected so that the new trees will be reversion in some way towards correction of mistakes from new trees. Each new tree can be compared with the residual errors of the entire assembling process and indeed, whichever is the process through which the prediction is being made, it is the result of the sum of all the decision trees.

In principle, gradient boosting is a process, which is done to minimize a given function (loss function) based on gradient descent. For each iteration the algorithm attempts to rectify the residualities left behind from the previous model by fitting on another new model. In fact, the generic rule for updating to boost is:

$$\hat{y}_m = \hat{y}_{m-1} + \eta \cdot f_m(x) \qquad (3)$$

Where: $\hat{y}_m$ would be the prediction if we added the model, $\hat{y}_{m-1}$ was the prediction in one iteration earlier, $f_m(x)$- the new model, typically some decision tree, eta is a rate in learning which is used to determine how much each tree is used. Because of the modelling of complex, non-linear relationships in data, XGBoost is ideal for fraudulent transaction detection. To put it simply, fraud detection is about very large datasets on the one side with an imbalanced set to a small number of truly fraudulent records on the other.

V. EXPERIMENTAL RESULTS AND ANALYSIS

5.1 Overall Performance Analysis

Random Forest Model

The Random Forest was the best performing algorithm obtained in terms of the measure of the accuracy metric showing a 99.96% accuracy rate of fraudulent transactions. It was able to identify 85,286 real transactions and 124 fraudulent transactions with a minimum of nine false positives (i.e., missed detection and deserted transactions), and with a minimum of 24 false negatives (i.e., being unable to identify some fraudulent transactions). When used on legit transactions, the precision is 1.00 so our model identifies legit transactions with a virtual lack of false positives. It seemed that the accuracy of fraudulent transactions was slightly lower, equal to 0.93, because there were few false positives, and the F-score for fraudulent transaction detection was 0.84 and the recall was 0.51. This means that many fake commands have been identified but some that were not caught by the model. The fraud detection F1 score is 0.88, which is a nice balance between a precision/recall of the results (Figure 2).

With Random Forest, 8526 legitimate transactions entered Ruby to fall under the legitimate class and 124 fraudulent transactions to the fraudulent class; there were 9 false positive and 24 false negative. In fact, this is a very high performing model. Macro-averaged scaling shows ((including Patients, Hits and H1) that the model is good at planning objective while the weighted average which negates bias which is caused by class imbalance clearly indicates that the model is great at classifying legitimate transactions. To summarize: the RF is very accurate but suffers a blow - the blow (which is a small blow in the case of valid transactions) to the accuracy is for the fraud detections.

doi: https://doi.org/10.65883/ijacm.2026v2i3.02

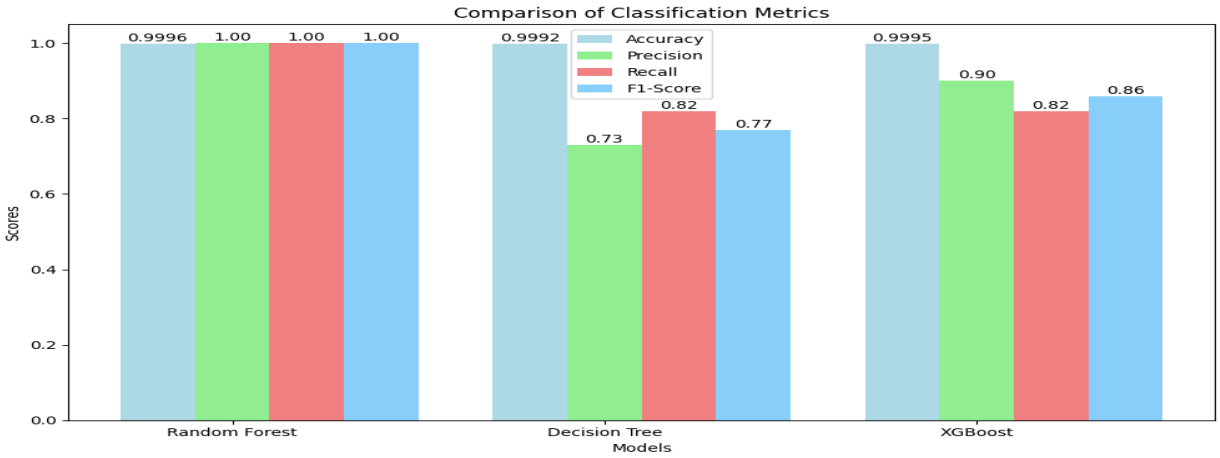

Figure 1: Comparison of Classification Metrics

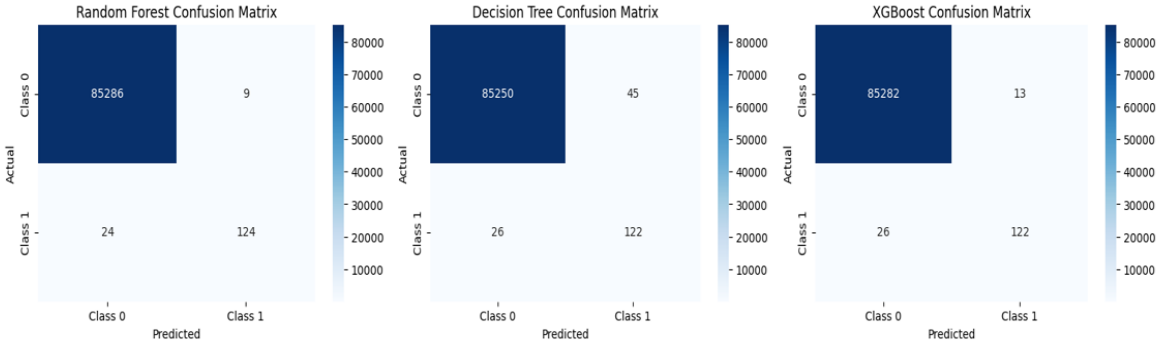

Figure 3: Comparison of Confusion Matrix

Decision Tree Model

It returned accuracy of 99.92% which is a little less than Random Forest but still very high accuracy. Number of matched transaction types found is used to phenomenologically accurately infer actual transactions with precision and recall of 1.00, which essentially means that our models didn't over-fit too much for transactions of this type. From the same set of transactions, we achieved a high recall on 0.82 and an average precision of 0.73. With respect to the fraud prediction, these are comparatively low scores as compared to Random Forest and XGBoost. The F1 score was then set to 0.77 for fraud detection - this score signals mid-range performance with regards to the number of false positives vs. false negatives in which the algorithm was developed for.

The Decision Tree confusion matrix shows it has 45 false positive predictions and has 26 false negative predictions, indicating how much of a challenge the model had identifying fraud transactions. The weighted averages are so high in this model that it is likewise effective at finding proper transactions. Despite its considerable general elegance and interpretability, Decision Tree performance on the Fraud detection task, however, is less than satisfactory; same score level performance achieved by significantly more sophisticated techniques such as Random Forest, or even gradient boosted techniques such as XGBoost.

XGBoost Model

This intimidated me and my point isn't so much that XGBoost is massively better (they're only 1/10 of a percent better than RF), but rather that even if you've accounted for the true standard deviation (dubbed

doi: https://doi.org/10.65883/ijacm.2026v2i3.02

"precision/recall bias" here), you find that RF still had a very large lead. Precision for Legit was 1.00, meaning that we were generally pretty accurate about calling a legit transaction legit, or calling a fraudulent transaction legit. The combination was used to detect fraud where it reached a Precision of 0.90, Recall of 0.82 and F-show case of 0.86 (different weights are used by this decision criteria). In any case, the results show the model slightly better at finding fraudulent transactions than at the Decision tree, though still can decrease false positives (labeling as fraudulent valid transactions) and false negatives (found tested fraudulent transactions as non-fraudulent).

As far as the confusion matrix is concerned, it's obvious that the XGBoost leads to 13 FPs and 26 FNs, which is very close to Random Forest and slightly better than Decision Tree. By macro-average and weighted mean scores, however, the XGBoost does a better job of balancing precision and recall than the Decision Tree and is far and away the strongest baseline in terms of identifying non-fraudulent transactions. Moreover, XGBoost does a perfect job of fraud detection (results closer to pure precision-recall than Random Forest). But without any doubts, XGBoost comes with a decent punch in fraud detection, since it can enjoy massive imbalance and better data arcs.

5.2 Comparative Model Analysis:

TABLE1: MODEL RESULTS

Metric	Random Forest	Decision Tree	XGBoost
Precision (Class 0)	1.0000	1.0000	1.0000
Recall (Class 0)	1.0000	1.0000	1.0000
F1-Score (Class 0)	1.0000	1.0000	1.0000
Precision (Class 1)	0.9300	0.7300	0.9000
Recall (Class 1)	0.8400	0.8200	0.8200
F1-Score (Class 1)	0.8800	0.7700	0.8600
Accuracy	0.9996	0.9992	0.9995

Among the methods, Random Forest shows the best performance of all other methods: accuracy: 99.96 percent, which is almost perfect as far as perfect precision for the true transaction is concerned but little lower because the false positive detection of fraudulent transactions (precision: 0.93). The recall of this is good (0.84) to predict the transaction as being fraudulent (F1 score is also good (0.88)), so this seems like a good machine learning technique to use for predicting fraudulent transactions in real time. Heree is showcased the first XGBoost model which generated an accuracy of 99.95% resulting in a relatively balanced result for fraud with a precision of 0.90, a recall of 0.82, and an F1 score of 0.86. Decision tree - Although it achieved accuracy of 99.92, its fraud detection performance fell short - precision was 0.73, recall was 0.82 and F1 was 0.77. Therefore, the suitable algorithm for fraud detection is Random Forest, because the quality of the Random Forest result is contained in the foot of normal and high efficiency, so we can select the first, then the algorithm XGBoost. Decision tree is not anyone's pet but can perform some work when it comes to detecting fraudulent transactions.

VI. CONCLUSION

AI-based credit card fraud detection and prevention are becoming an important chunk in the financial sector transactions protection. This system is successful in integrating the best machine learning algorithms to detect and prevent fraudulent activities to ensure that the potential cybercriminals cannot make authorization or un-authorization transactions. As such it brings security of system, build customer trust, encourages the financial stability of system. However, this to advance in the future may have the purpose of further honing the system in its ability to work with even more complex fraud situations and help in detecting these more accurately.

doi: https://doi.org/10.65883/ijacm.2026v2i3.02

Despite the success of nowadays algorithms still though are the limitations, especially in the capture of the modification of fraud Mold and of an awestable number of fraudulent/cut straight transactions. Future research could be to consider deep learning methods to better capture the temporal dependencies in transactional data as there may be long range dependencies. Furthermore, a combination of blockchain to verify fraudulent transactions and explainable AI methods might potentially improve transparency and accountability of fraud detection models which answer some of the regulations. The possibilities of such federated learning to allow for hidden inter-organizational model training is promising in further improving the accuracy and strength of the system, ensuring that the system is flexible to the tactics that the fraudsters are using, and is also scalable across financial institutions.

REFERENCES

[1] C. Yu, Y. Xu, J. Cao, Y. Zhang, Y. Jin, and M. Zhu, "Credit Card Fraud Detection Using Advanced Transformer Model," Proceedings - 2024 IEEE International Conference on Metaverse Computing, Networking, and Applications, MetaCom 2024, pp. 343–350, Jun. 2024, doi: 10.1109/MetaCom62920.2024.00064.

[2] A. F. Ghahfarokhi, T. Mansouri, M. R. S. Moghadam, N. Bahrambeik, R. Yavari, and M. F. Sani, "Credit Card Fraud Detection Using Asexual Reproduction Optimization," Kybernetes, vol. 51, no. 9, pp. 2852–2876, May 2023, doi: 10.1108/K-04-2021-0324.

[3] M. N. Alatawi, "Detection of fraud in IoT based credit card collected dataset using machine learning," Machine Learning with Applications, vol. 19, p. 100603, Mar. 2025, doi: 10.1016/J.MLWA.2024.100603.

[4] Y. Chen, C. Zhao, Y. Xu, C. Nie, and Y. Zhang, "Deep Learning in Financial Fraud Detection: Innovations, Challenges, and Applications," Data Science and Management, Aug. 2025, doi: 10.1016/J.DSM.2025.08.002.

[5] S. Xiang et al., "Semi-supervised Credit Card Fraud Detection via Attribute-Driven Graph Representation," Proceedings of the 37th AAAI Conference on Artificial Intelligence, AAAI 2023, vol. 37, pp. 14557–14565, Dec. 2024, doi: 10.1609/aaai.v37i12.26702.

[6] K. Hayat and B. Magnier, "Data Leakage and Deceptive Performance: A Critical Examination of Credit Card Fraud Detection Methodologies," Jun. 2025, doi: 10.3390/math13162563.

[7] Y. Chen, C. Zhao, Y. Xu, C. Nie, and Y. Zhang, "Year-over-Year Developments in Financial Fraud Detection via Deep Learning: A Systematic Literature Review," Jul. 2025, Accessed: Sep. 17, 2025. [Online]. Available: https://arxiv.org/pdf/2502.00201v1

[8] M. A. Alrasheedi, "Enhancing Fraud Detection in Credit Card Transactions: A Comparative Study of Machine Learning Models," Comput Econ, pp. 1–27, Aug. 2025, doi: 10.1007/S10614-025-11071-3/TABLES/12.

[9] M. A. Talukder, R. Hossen, M. A. Uddin, M. N. Uddin, and U. K. Acharjee, "Securing Transactions: A Hybrid Dependable Ensemble Machine Learning Model using IHT-LR and Grid Search," Cybersecurity, vol. 7, no. 1, Feb. 2024, doi: 10.1186/s42400-024-00221-z.

[10] Y. T. Lei, C. Q. Ma, Y. S. Ren, X. Q. Chen, S. Narayan, and A. N. Q. Huynh, "A distributed deep neural network model for credit card fraud detection," Financ Res Lett, vol. 58, p. 104547, Dec. 2023, doi: 10.1016/J.FRL.2023.104547.