International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Ensemble Learning for Credit Card Fraud Detection Using Machine Learning

Dr. S Gnanapriya
Associate Professor, Department of Computer Applications, Nehru College of Management, Coimbatore, Tamil Nadu, India
Email:ncmdrsgnanapriya@nehrucolleges.com

Kavikannan N
Student, II MCA, Nehru College of Management, Coimbatore, Tamil Nadu, India
Email:kavikannan1908@gmail.com

Abstract

Credit card fraud causes substantial financial loss to cardholders, banks and merchants, and its detection is difficult because fraudulent transactions form only a tiny fraction of all transactions. A single classifier trained on such highly imbalanced data tends to favour the majority (legitimate) class and may miss fraud cases. This research presents an ensemble learning approach for credit card fraud detection and compares six models: Logistic Regression and Decision Tree as individual classifiers, Random Forest and XGBoost as tree-based ensembles, and a Voting ensemble and a Stacking ensemble as hybrid ensembles. Preprocessing and class-imbalance handling are applied to the training data only, so that no information from the test set leaks into model building. Because overall accuracy is misleading on imbalanced data, the models are compared using precision, recall, F1-score and ROC-AUC. In the experiments, the Stacking ensemble achieved the highest ROC-AUC (0.8577), while the Voting ensemble achieved the highest F1-score (0.3600) with a precision of 0.3913 and a recall of 0.3333. Logistic Regression detected the largest share of fraud (recall 0.4815) but produced very low precision (0.0084), and Random Forest and Stacking reached a precision of 0.5000 with a recall of only 0.0370. The results show that ensemble methods improve the ranking and balance of fraud detection, but that low precision and recall on the minority class remain a challenge, and that decision-threshold tuning and larger fraud samples are needed.

Keywords: Credit Card Fraud Detection, Ensemble Learning, Machine Learning, Class Imbalance, Random Forest, XGBoost, Voting, Stacking.

1. Introduction

The growth of e-commerce, mobile payments and contactless cards has made credit cards one of the most widely used payment instruments. This growth has also increased opportunities for fraud, including stolen card details, card-not-present attacks and account takeover. Fraud losses reduce customer trust and increase operating costs for financial institutions, which must review suspicious transactions while keeping genuine payments uninterrupted.

Rule-based fraud systems are simple to explain but are difficult to maintain, because fraudsters adapt as soon as rules become known. Machine learning offers a data-driven alternative: classifiers learn patterns that separate fraudulent from legitimate behaviour using labelled historical transactions [12], [10]. However, two characteristics of the problem make it harder than ordinary classification. First, the data are extremely imbalanced, so a model that labels every transaction as legitimate can still obtain very high accuracy while detecting no fraud. Second, fraud patterns change over time, a phenomenon known as concept drift [17].

Ensemble learning addresses part of this difficulty by combining the predictions of several models so that their individual errors partly cancel out [14]. Bagging reduces variance [7], boosting reduces bias by focusing on difficult examples [5], [8], and stacking learns how to combine heterogeneous models [6]. Tree-based ensembles such as Random Forest [2] and XGBoost [3] are among the strongest methods for structured tabular data.

This research develops and evaluates an ensemble learning framework for credit card fraud detection. The contribution of this work is a systematic, leakage-free comparison of individual classifiers and ensemble strategies under a single evaluation protocol that uses metrics suited to imbalanced data.

2. Related Research Work

2.1 Machine Learning for Credit Card Fraud Detection

Alarfaj et al. [12] compared machine-learning and deep-learning algorithms for credit card fraud detection on the European cardholder benchmark dataset and also experimented with balancing the data to reduce the false-negative rate. Esenogho et al. [10] combined feature engineering with a neural-network ensemble and reported improved detection compared with single models. Shahapurkar and Patil [17] studied fraud detection in a streaming environment and addressed class imbalance together with concept drift, which shows that evaluation settings should reflect real deployments.

2.2 Class Imbalance Handling

SMOTE creates synthetic minority samples by interpolating between neighbouring minority-class examples rather than duplicating them. Azhar et al. [4] reviewed SMOTE-based methods and compared them experimentally on 24 imbalanced datasets, relating their behaviour to data-complexity measures. Hayat and Magnier [18] showed that applying resampling before the train-test split causes data leakage and inflated scores in credit card fraud studies, so resampling must be applied to training data only. Khalid et al. [1] combined undersampling and SMOTE with ensemble classifiers for credit card fraud detection and showed that balancing the data is important for the rare fraud class.

2.3 Ensemble Learning

Mienye and Sun [14] surveyed ensemble learning, covering bagging, boosting and stacking and algorithms such as Random Forest, AdaBoost, gradient boosting, XGBoost and LightGBM. Ganaie et al. [7] reviewed ensemble models and grouped them into bagging, boosting and stacking families. Ileberi et al. [5] combined SMOTE with AdaBoost for credit card fraud detection, and Abdulghani et al. [8] applied XGBoost to the same task. Bentéjac et al. [3] compared XGBoost, LightGBM, CatBoost and Random Forest, and Grinsztajn et al. [2] showed that tree-based models such as Random Forest and XGBoost remain state of the art on medium-sized tabular data. Almalki and Masud [15] combined XGBoost, LightGBM and CatBoost in a stacking ensemble, and Islam et al. [6] proposed a stacking ensemble, in which a meta-learner combines the predictions of base learners, for credit card anomaly detection with imbalanced classes.

2.4 Hybrid and Unsupervised-Supervised Approaches

Carcillo et al. [11] combined unsupervised outlier scores with supervised classifiers and reported that the added information can improve fraud detection. Such work shows that fraud detection benefits from combining complementary sources of evidence, which is the principle that ensemble learning applies at the model level.

2.5 Evaluation on Imbalanced Data

Richardson et al. [9] clarified how the ROC and precision-recall curves behave under class imbalance and showed that the precision-recall curve is highly sensitive to the class ratio, so it must be read together with that ratio. Chicco and Jurman [16] argued that the Matthews Correlation Coefficient gives a high score only when all four confusion-matrix rates are high, and that accuracy and F1-score can be over-optimistic on imbalanced data. For this reason precision, recall, F1-score and ROC-AUC are examined together in the present study.

2.6 Research Gap

Existing studies establish the value of resampling, ensemble classifiers and imbalance-aware metrics, but they often report these elements separately, apply resampling before the data split, or rely on accuracy and ROC-AUC alone. This research integrates preprocessing, leakage-free resampling, multiple ensemble families and imbalance-aware evaluation into a single reproducible workflow.

3. Problem Definition and Research Objectives

3.1 Problem Definition

Given a set of labelled card transactions described by numerical features, the task is to learn a function that assigns each new transaction to one of two classes: legitimate (0) or fraudulent (1). The task is difficult because (i) fraud is very rare, (ii) the cost of a missed fraud is usually higher than the cost of a false alarm, (iii) detection should be fast enough for near-real-time use and (iv) the model should remain reliable when behaviour changes.

3.2 Research Objectives

The main objectives of this research are:

- To prepare credit card transaction data through duplicate checking and feature scaling without data leakage.
- To handle class imbalance by applying resampling to the training partition only.
- To implement Logistic Regression and Decision Tree as individual baseline classifiers.
- To implement Random Forest, XGBoost, Voting and Stacking ensembles for fraud detection.
- To compare all models using precision, recall, F1-score and ROC-AUC rather than accuracy alone.
- To analyse the strengths and weaknesses of each model in detecting the rare fraud class.

4. Proposed Methodology

4.1 Overall Framework

The proposed framework consists of six stages: data acquisition, preprocessing, stratified splitting, imbalance handling, model training (individual and ensemble) and evaluation. The overall workflow is shown in Figure 1.

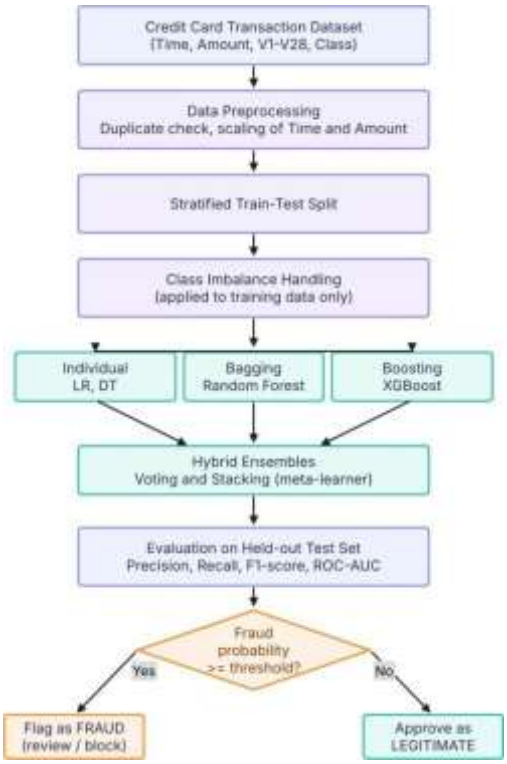

Figure 1. Overall Architecture of the Proposed Ensemble Fraud Detection Framework

4.2 Dataset Description

Property	Description
Dataset name	Credit Card Fraud Detection Dataset
Dataset source	Kaggle
Total transactions	284,807
Fraudulent transactions	492
Legitimate transactions	284,315
Fraud percentage	0.1727%
Legitimate percentage	99.8273%
Input features (30)	Time, V1–V28, Amount
Target column	Class (0 = legitimate, 1 = fraud)
Collection period	September 2013 (two days)
Missing values	To be confirmed by checking the CSV file
Duplicate records	To be confirmed by checking the CSV file

Table 1. Summary of the Credit Card Transaction Dataset

4.3 Data Preprocessing

The data are first checked for missing values and duplicate records. Numerical attributes with large ranges, such as the transaction amount, are standardised using a scaler that is **fitted on the training data only** and then applied to the test data, so that no test-set statistics influence the training process.

4.4 Stratified Splitting

The dataset contains 284,807 records. The reported held-out test set contains 8,000 records (27 fraudulent and 7,973 legitimate); therefore, the remaining 276,807 records are used for training. This corresponds to an approximate train-test split of 97.19:2.81. The test set is kept untouched until final evaluation. The reported test-set fraud rate (0.3375%) differs from the full-dataset rate (0.1727%); verify the stratified split using the original code.

4.5 Class Imbalance Handling

Fraud cases form a very small fraction of the dataset, so a classifier trained directly on it may favour the legitimate class. This study uses the Synthetic Minority Over-sampling Technique (SMOTE), which creates synthetic minority-class samples by interpolating between neighbouring fraud examples [4]. SMOTE is applied only to the training data. Applying it before splitting could cause information leakage and produce unrealistically high scores [18]. The test set therefore retains its original, imbalanced distribution.

4.6 Individual Classifiers

- **Logistic Regression (LR):** a linear probabilistic model used as an interpretable baseline.
- **Decision Tree (DT):** a rule-based non-linear model that is easy to interpret but prone to overfitting.

4.7 Ensemble Models

- **Random Forest (RF):** combines predictions from multiple decision trees trained on bootstrap samples and random feature subsets [2], [7].
- **XGBoost:** builds trees sequentially, with each new tree correcting errors from earlier trees; regularisation helps control overfitting [3], [8].
- **Voting Ensemble (Soft Voting):** combines predicted class probabilities from Logistic Regression (LR), Random Forest (RF) and XGBoost, then selects the class with the highest combined probability.
- **Stacking Ensemble:** uses Random Forest (RF), XGBoost and Decision Tree (DT) as base learners. Logistic Regression is the final meta-learner; 3-fold cross-validation (cv=3) is used to train the stacking classifier [6].

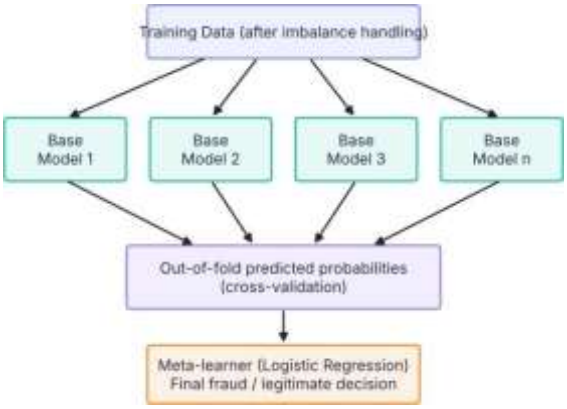

Figure 2. Stacking Ensemble Architecture

4.8 Evaluation Metrics

Accuracy alone is not used as a selection criterion, because on highly imbalanced data a classifier that predicts every transaction as legitimate would obtain very high accuracy while detecting no fraud. The following measures are used, with fraud as the positive class:

- **Precision** = TP / (TP + FP): the proportion of raised alerts that are real fraud.
- **Recall** = TP / (TP + FN): the proportion of real fraud that is detected.
- **F1-score** = 2 × Precision × Recall / (Precision + Recall).
- **ROC-AUC**: the probability that a random fraud transaction is ranked above a random legitimate one, independent of any decision threshold.

Here TP, FP, TN and FN denote true positives, false positives, true negatives and false negatives. Recall is examined separately because missed frauds are costly, and precision because false alarms burden investigators and inconvenience genuine customers.

5. System Design and Implementation

5.1 Technology Stack

- **Programming Language:** Python.
- **Data Handling:** Pandas and NumPy for loading, cleaning and transformation.
- **Machine Learning:** Scikit-learn [13] for classifiers, ensembles, pipelines and metrics; XGBoost [3] for gradient boosting.
- **Visualisation:** Matplotlib for charts and confusion matrices.
- **Development Environment:** Jupyter Notebook or Visual Studio Code with a Python virtual environment.

5.2 Implementation Modules

Module	Function
Data Loading	Reads the transaction file and verifies shape, data types, missing values and duplicates.
Preprocessing	Scales numerical attributes using statistics from the training data only.
Splitting	Creates the stratified train-test partition with a fixed random state.
Resampling	Balances the training data only.
Model Training	Trains Logistic Regression, Decision Tree, Random Forest, XGBoost, Voting and Stacking models.
Evaluation	Computes precision, recall, F1-score and ROC-AUC on the held-out test set.

Table 2. Implementation Modules and Their Functions

5.3 Leakage Prevention Measures

- Scaling statistics and resampling are fitted on training data only.
- The test set retains the original fraud ratio and is used only for the final evaluation.

6. Experimental Results and Analysis

6.1 Experimental Configuration

Setting	Value
Models evaluated	Logistic Regression, Decision Tree, Random Forest, XGBoost, Voting Ensemble, Stacking Ensemble
Train-test split	97.19:2.81 (276,807 training; 8,000 testing)
Imbalance handling	SMOTE, applied to training data only
Positive class	Fraud
Evaluation metrics	Precision, Recall, F1-score, ROC-AUC
Test set	8,000 records (27 fraud, 7,973 legitimate; fraud = 0.34%)
Training records	276,807 records

Table 3. Experimental Configuration

6.2 Comparative Performance of Models

Table 4 reports the performance of the six models on the held-out test set for the fraud class, and Figure 3 presents the same scores graphically.

Model	Precision	Recall	F1-score	ROC-AUC
Logistic Regression	0.0084	0.4815	0.0166	0.7988
Decision Tree	0.0103	0.3333	0.0200	0.5584
Random Forest	0.5000	0.0370	0.0690	0.8473
XGBoost	0.2759	0.2963	0.2857	0.8301
Voting Ensemble	0.3913	0.3333	**0.3600**	0.8380
Stacking Ensemble	0.5000	0.0370	0.0690	**0.8577**

Table 4. Comparative Results of Fraud Detection Models (Test Set, Fraud Class). Best F1-score and best ROC-AUC are in bold.

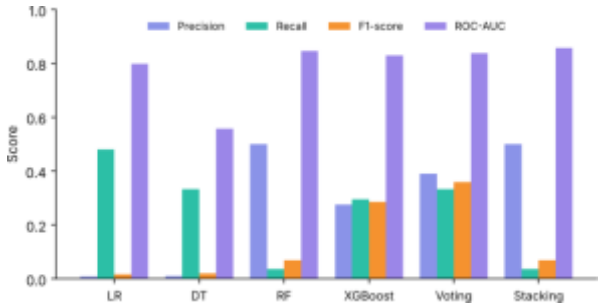

Figure 3. Comparison of Precision, Recall, F1-score and ROC-AUC Across Models

6.3 Confusion Matrix Analysis

The held-out test set contains 8,000 transactions, of which 27 are fraudulent and 7,973 are legitimate. Table 5 gives the confusion-matrix counts of each model, with fraud as the positive class, together with overall accuracy.

Model	TN	FP	FN	TP	Accuracy
Logistic Regression	6,444	1,529	14	13	80.71%
Decision Tree	7,111	862	18	9	89.00%
Random Forest	7,972	1	26	1	99.66%
XGBoost	7,952	21	19	8	99.50%
Voting Ensemble	7,959	14	18	9	99.60%
Stacking Ensemble	7,972	1	26	1	99.66%

Table 5. Confusion-Matrix Counts and Accuracy of the Models on the Test Set (TN: true negatives, FP: false positives, FN: false negatives, TP: true positives)

6.4 Discussion

Overall performance. Precision and F1-score are low for most models. This is expected on highly imbalanced data, where even a small false-positive rate produces many false alarms compared with the very few real frauds. ROC-AUC values are comparatively high (0.80–0.86 for five of the six models), but ROC-AUC can look favourable on imbalanced data, which is why precision, recall and F1-score are examined alongside it.

Individual classifiers. Logistic Regression detected the most fraud (recall 0.4815, 13 of 27), but its precision of 0.0084 means that only 13 of its 1,542 alerts (about one in 119) were real frauds, with 1,529 genuine transactions wrongly flagged, which would be impractical for investigators. The Decision Tree performed weakest overall, with an ROC-AUC of 0.5584, close to random ranking, which is consistent with the tendency of a single tree to overfit; it detected 9 frauds but raised 862 false alarms.

Tree-based ensembles. Random Forest ranked transactions well (ROC-AUC 0.8473) and reached a precision of 0.5000, but at the default decision threshold it raised only two alerts, one correct and one false, and detected only 1 of 27 frauds (recall 0.0370). XGBoost gave a more balanced result, with precision 0.2759, recall 0.2963 and F1-score 0.2857, detecting 8 frauds with 21 false alarms.

Hybrid ensembles. The Voting ensemble achieved the best F1-score (0.3600), improving on XGBoost by about 0.074, and detected 9 frauds with 14 false alarms, which gave the best balance between false alarms and missed frauds. The Stacking ensemble achieved the highest ROC-AUC (0.8577), which indicates the best ranking of fraud above legitimate transactions, but its precision and recall at the default threshold were the same as Random Forest (0.5000 and 0.0370, with an identical confusion matrix). This suggests that its predicted probabilities rank cases well but that the default threshold is too conservative for the rare fraud class, so threshold tuning is likely to improve recall.

Accuracy is misleading. Random Forest and Stacking obtained the highest accuracy (99.66%) although they detected only 1 of 27 frauds, while Logistic Regression obtained the lowest accuracy (80.71%) yet detected the most frauds. This confirms that accuracy cannot be used to select a fraud-detection model on imbalanced data.

Reliability of the comparison. Because the test set contains only 27 fraud cases, a change of a single detected fraud alters recall by about 0.037. Differences between models, particularly between XGBoost, Voting and Logistic Regression, should therefore be interpreted with caution and confirmed with cross-validation or a larger fraud sample.

7. Limitations and Future Enhancements

7.1 Limitations

- **Small number of fraud cases:** the test set contains only 27 fraudulent transactions, so the metrics have high variance and small differences between models may not be significant.
- **Low precision and recall:** most models achieved low precision, and the best recall was 0.4815, so a large share of frauds was still missed or many false alarms were raised.
- **Default decision threshold:** the models were evaluated without decision-threshold tuning, which limits recall for the Random Forest and Stacking models.
- **Metrics reported:** precision-recall AUC and Matthews Correlation Coefficient were not computed.
- **Offline evaluation:** the experiments use a static dataset and do not measure real-time latency, concept drift or alert-review cost.
- **Dataset dependence:** the results are specific to the selected dataset, split and configuration and should not be taken as a universal ranking of algorithms.

7.2 Future Enhancements

- Tune the decision threshold on validation data to balance recall and precision.

- Report PR-AUC [9] and Matthews Correlation Coefficient [16], and use stratified cross-validation for more stable estimates.
- Evaluate additional ensembles such as LightGBM [15] and cost-sensitive learning that uses transaction amount and review cost.
- Use a larger dataset with more fraud examples and compare several resampling strategies.
- Apply explanation methods such as SHAP [19] to explain individual fraud alerts.
- Use time-aware validation and incremental retraining to handle concept drift, and test the model under real-time transaction loads.

8. Conclusion

This research compared six machine-learning models, Logistic Regression, Decision Tree, Random Forest, XGBoost, a Voting ensemble and a Stacking ensemble, for credit card fraud detection using imbalance-aware evaluation. The Voting ensemble gave the best balance of precision and recall (F1-score 0.3600), and the Stacking ensemble gave the best ROC-AUC (0.8577), which supports the use of ensemble learning for this problem. At the same time, the individual models showed a clear trade-off between detecting fraud and raising false alarms, and the low precision and recall on the fraud class show that further work is required. Decision-threshold tuning, evaluation on a larger fraud sample with cross-validation, and real-time validation are needed before the approach can be considered for practical use.

Acknowledgement

The author sincerely thanks Dr. S Gnanapriya, Department of Computer Applications, Nehru College of Management, Coimbatore, for her valuable guidance and support throughout the project work and the preparation of this research paper.

Author's Biography

Kavikannan N is a postgraduate (MCA) student in the Department of Computer Applications, Nehru College of Management, Coimbatore, Tamil Nadu, India. His interests include machine learning, cloud computing and cyber security.

References

[1] A. R. Khalid, N. Owoh, O. Uthmani, M. Ashawa, J. Osamor and J. Adejoh, "Enhancing credit card fraud detection: An ensemble machine learning approach," *Big Data and Cognitive Computing*, vol. 8, no. 1, art. 6, 2024.

[2] L. Grinsztajn, E. Oyallon and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?," in *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, vol. 35, 2022.

[3] C. Bentéjac, A. Csörgő and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artificial Intelligence Review*, vol. 54, no. 3, pp. 1937–1967, 2021.

[4] N. A. Azhar, M. S. M. Pozi, A. M. Din and A. Jatowt, "An investigation of SMOTE based methods for imbalanced datasets with data complexity analysis," *IEEE Trans. Knowledge and Data Engineering*, vol. 35, no. 7, pp. 6651–6672, 2023.

[5] E. Ileberi, Y. Sun and Z. Wang, "Performance evaluation of machine learning methods for credit card fraud detection using SMOTE and AdaBoost," *IEEE Access*, vol. 9, pp. 165286–165294, 2021.

[6] M. A. Islam, M. A. Uddin, S. Aryal and G. Stea, "An ensemble learning approach for anomaly detection in credit card data with imbalanced and overlapped classes," *Journal of Information Security and Applications*, vol. 78, art. 103618, 2023.

[7] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer and P. N. Suganthan, "Ensemble deep learning: A review," *Engineering Applications of Artificial Intelligence*, vol. 115, art. 105151, 2022.

[8] A. Q. Abdulghani, O. N. Uçan and K. M. Ali Alheeti, "Credit card fraud detection using XGBoost algorithm," in *Proc. 14th Int. Conf. on Developments in eSystems Engineering (DeSE)*, 2021, pp. 487–492.

[9] E. Richardson, R. Trevizani, J. A. Greenbaum, H. Carter, M. Nielsen and B. Peters, "The receiver operating characteristic curve accurately assesses imbalanced datasets," *Patterns*, vol. 5, no. 6, art. 100994, 2024.

[10] E. Esenogho, I. D. Mienye, T. G. Swart, K. Aruleba and G. Obaido, "A neural network ensemble with feature engineering for improved credit card fraud detection," *IEEE Access*, vol. 10, pp. 16400–16407, 2022.

[11] F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Obleé and G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection," *Information Sciences*, vol. 557, pp. 317–331, 2021.

[12] F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan and M. Ahmed, "Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms," *IEEE Access*, vol. 10, pp. 39700–39715, 2022.

[13] scikit-learn developers, "Scikit-learn: Machine learning in Python – User guide," 2026. [Online]. Available: https://scikit-learn.org/stable/ (accessed Nov. 21, 2026).

[14] I. D. Mienye and Y. Sun, "A survey of ensemble learning: Concepts, algorithms, applications, and prospects," *IEEE Access*, vol. 10, pp. 99129–99149, 2022.

[15] F. Almalki and M. Masud, "Financial fraud detection using explainable AI and stacking ensemble methods," arXiv preprint arXiv:2505.10050, 2025.

[16] D. Chicco and G. Jurman, "The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification," *BioData Mining*, vol. 16, art. 4, 2023.

[17] A. Shahapurkar and R. Patil, "Concept drift and machine learning model for detecting fraudulent transactions in streaming environment," *Int. Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 5, pp. 5560–5568, Oct. 2023.

[18] K. Hayat and B. Magnier, "Data leakage and deceptive performance: A critical examination of credit card fraud detection methodologies," arXiv preprint arXiv:2506.02703, 2025.

[19] C. Kotrachai, P. Chanruangrat, T. Thaipisutikul, W. Kusakunniran, W.-C. Hsu and Y.-C. Sun, "Explainable AI supported evaluation and comparison on credit card fraud detection models," in *Proc. 7th Int. Conf. on Information Technology (InCIT)*, 2023, pp. 86–91.