International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2025: 7.089

Optimising Credit Card Fraud Detection: A Validation-Weighted Random Forest and XGBoost Ensemble

Dammu Ramadevi[1], Shaik Dilnawaz[2]

[1]Department of Computer Science & Engineering St. Mary's Womens Engineering College, Guntur Budampadu, India
Email: *ramadevidammu1[at]gmail.com*

[2]Department of Computer Science & Engineering St.Mary's Womens Engineering College, Guntur Budampadu, India
Email: *shaikdilnawaz24[at]gmail.com*

Abstract: ***Credit card fraud detection is dominated by a severe class imbalance in which fraudulent records constitute a fraction of one percent of transaction volume, so classifiers tuned for accuracy alone conceal unacceptable false-negative rates. This paper develops a weighted soft-voting ensemble of Random Forest and XGBoost for the public ULB credit card dataset of 284,807 European transactions. Class skew is addressed through balanced subsampling and gradient scaling rather than synthetic oversampling, and both the ensemble weights and the operating threshold are selected on a held-out validation partition to avoid test-set leakage. On the untouched test partition the ensemble attains an F1-score of 0.8804, precision of 0.9419, Matthews correlation of 0.8821, and AUPRC of 0.8695, reducing false alarms below either base learner at identical recall while completing training in 77 s on a single CPU core.***

Keywords: Credit Card Fraud Detection · Random Forest · XGBoost · Ensemble Learning · Class Imbalance · Precision–Recall Analysis

1. Introduction

Card-not-present payments have expanded so quickly that manual review of suspicious transactions is no longer feasible for any acquirer of meaningful scale, and losses to payment fraud worldwide are measured in tens of billions of dollars annually [1]. The first line of defence is formed by machine learning classifiers, with each authorisation request being scored in milliseconds, enabling only a small fraction of suspicious transactions to be left for human analysts. Tree ensembles are the method of choice, with Random Forest giving the best variance reduction through bootstrap aggregation and being less prone to irrelevant features, while gradient-boosted trees, such as XGBoost, allow for minimising a regularised objective function using second-order optimisation and are amongst the best performers for tabular data [2, 12]. Multiple recent studies have demonstrated their superiority over single classifier solutions and the majority of deep learning architectures when applied to the anonymized transaction data with moderate dimensionality [4, 5].

Despite the apparent maturity of the field, an analysis of the literature within the last 3 years uncovered several common methodological missteps. Firstly, papers are quick to report accuracy on data where the fraud prevalence is known to be below 0.2%, noting that a classifier that always predicts Normal achieves 99.8% accuracy, while AUPRC and the Matthews correlation coefficient are more insightful metrics, if less commonly reported [1,6]. Secondly, thresholds for decision making and the combination of decisions within ensembles are often optimised on the same dataset that is used to report results, leading to an implicit leakage of information and an overestimated F1-score [7]. Thirdly, minority classes are often oversampled using SMOTE or analogous approaches prior to train/test split, biasing the evaluation in an even greater degree [10,17]. Fourthly, while an ensemble of heterogeneous methods was used in most works, their decisions were combined using either averaging or majority voting, throwing away potentially valuable information about the relative accuracy of individual classifiers [6,7]. Lastly, computational complexity was rarely quantified, which is a critical consideration given that the frequency of updates in response to concept drift dictates how often a model can be retrained [13].

This paper addresses these gaps by proposing a design that is deliberately simple and fully reproducible. The contributions are as follows. (1) A weighted soft-voting ensemble of the Random Forest and XGBoost is engineered, in which the combination weights are learned from the average precision on the validation set rather than fixed at initialization. (2) The class imbalance is mitigated with balanced subsampling inside the forest and gradient re-weighting inside the booster without introducing synthetic examples, thus ensuring that the evaluation set is not contaminated with the training data. (3) The three-way stratified hold-out set splits prevent the leakage of information between the stages of model fitting, weight and threshold selection, and final testing as shown in the top-level design. (4) All claims are supported on the public ULB dataset with fixed random seeds and standard hardware, reporting the metrics relevant for imbalanced classification, namely the area under the precision-recall curve (AUPRC) and the Matthews Correlation Coefficient (MCC), alongside the standard accuracy and F1 scores. The rest of the paper is structured as follows. Section 2 reviews the related works from 2024 to 2026. Section 3 describes the methodology. Section 4 defines the experimental setup. Section 5 reports the results and discusses the implications. Finally, Sect. 6 concludes the study and outlines the directions for future work. 6 concludes.

2. Literature Survey

The credit card fraud detection research performed in 2024-2026 was clustered into four themes: classical ensemble learning, imbalance handling, metaheuristic optimization, and deep or graph-oriented design. Hafez et al. [1] conducted a systematic review of 52 articles and concluded that hybrid and ensemble techniques are the future in credit card fraud detection, with class imbalance and processing costs being the two main challenges. Garg et al. [4] similarly found in their 2026 review that tree ensembles have the best cost/benefit ratio compared to deep learning architectures in detecting fraud on anonymous transaction datasets.

Within the ensemble branch, Ikermane et al. [2] compared the performance of Random Forest and XGBoost algorithms on the ULB data set with subsequent interpretation using SHAP and LIME explanation tools. The precision of the Random Forest method reached 0.86, with the most influential features identified as V14 and V12. Vijaya Sri et al. [3] investigated the recently proposed combinations of bagging and boosting algorithms at ICDAM 2024 while Khalid et al. [6] confirmed the superiority of stacking ensembles with heterogeneous members for addressing imbalanced data sets. Mim et al. [7] showed that soft voting was better than hard voting for credit card fraud scoring, although their combination used equal weights for the majority and minority classes. The current study extended these findings by introducing the members' weights based on the validation set performance (average precision). Kumar et al. [12] and Tayebi and El Kafhali [11] addressed fraudulent transaction detection using boosting-type algorithms. The latter article applied the Bayesian optimization technique to tune XGBoost hyperparameters, while the former identified the gradient boosting factor for minority class balancing as the most critical.

For imbalance treatment, El Kafhali et al. [10] proposed a method that combined resampling with an optimized deep network, whereas Faisal et al. [17] used evolutionary feature selection with random weight networks to mitigate the increased dimensional bias problem caused by oversampling. Shome et al. [16] studied deep networks under artificially balanced conditions and conceded that performance degrades when the natural skew is restored, reinforcing the argument for cost-sensitive learning over synthetic balancing that this paper adopts.

The metaheuristic strand is represented by Mosa et al. [8], who wrapped feature selection in swarm optimizers, and Sorour et al. [9], whose brown-bear optimization improved classifier configuration at a substantial search cost. On the deep learning side, Cherif et al. [14] proposed an encoder–decoder graph neural network that models card–merchant relations, Motie and Raahemi [15] surveyed graph neural networks for financial fraud, and Sumaya and Nadher [18] refined deep models for the balanced-data condition. Baabdullah et al. [13] examined federated learning with blockchain for privacy-preserving detection, reporting that communication overhead delays real-time scoring. Srivastava et al. [5] provided a 2026 baseline study of classical learners. Across these works, models that respect the natural class distribution and report imbalance-aware metrics remain a minority, and validation-based selection of ensemble weights and thresholds is almost never made explicit; those are precisely the practices operationalised here.

3. Methodology

3.1 Dataset Description and Preprocessing

All experiments use the public ULB machine learning group dataset of European card transactions, released through Kaggle and mirrored on public repositories. It contains 284,807 transactions made over two days in September 2013, of which 492 (0.1727%) are frauds. Twenty-eight features V1–V28 are principal components of confidential original attributes; only Time (seconds elapsed since the first transaction) and Amount (EUR) are retained in raw form. Because V1–V28 are already standardized by the PCA transform, only Amount and Time require scaling; both are transformed with a robust scaler based on the median and interquartile range so that the many small transactions and the heavy right tail of Amount do not distort tree split thresholds. The data are partitioned by stratified sampling into a fitting set (64%, 182,276 rows, 315 frauds), a validation set (16%, 45,569 rows, 79 frauds) used exclusively for ensemble weight and threshold selection, and a test set (20%, 56,962 rows, 98 frauds) that is touched only once for final reporting.

3.2 Random Forest Base Learner

Random Forest grows an ensemble of T decision trees, each trained on a bootstrap replicate of the fitting set with a random feature subset examined at every node. Node quality is measured by the Gini impurity

$$G(t) = 1 - \Sigma_k p_k^2 \quad (1)$$

where p_k is the proportion of class k among the samples reaching node t; a split is chosen to maximize the impurity decrease. Equation (1) drives the ensemble toward pure leaves, and because fraud is rare, each bootstrap draw is balanced by subsampling the majority class so minority points influence Eq. (1) proportionally. The forest's fraud probability for a transaction x is the average of the tree votes,

$$P_{RF}(y = 1 \mid x) = (1/T)\, \Sigma_{t=1..T} h_t(x) \quad (2)$$

where $h_t(x) \in \{0, 1\}$ is the vote of tree t. Averaging in Eq. (2) reduces the variance of any single deep tree while leaving its low bias intact, which is the statistical basis for the forest's stability on noisy transactional features.

3.3 XGBoost Base Learner

XGBoost fits an additive model of K regression trees by minimizing the regularized objective

$$L(\varphi) = \Sigma_i l(\hat{y}_i, y_i) + \Sigma_k \Omega(f_k), \quad \Omega(f) = \gamma T + \tfrac{1}{2}\lambda \|w\|^2 \quad (3)$$

in which l is the logistic loss, T the number of leaves, w the leaf weights, and γ, λ the complexity penalties that suppress overfitting. Equation (3) is intractable directly, so at

boosting round t the loss is expanded to second order around the current prediction,

$$\tilde{L}^{(t)} \approx \Sigma_i \, [g_i f_t(x_i) + \tfrac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \quad (4)$$

where g_i and h_i are the first and second derivatives of the loss for sample i. Equation (4) yields a closed-form quality score for any candidate split,

$$\text{Gain} = \tfrac{1}{2}[G_L^2/(H_L + \lambda) + G_R^2/(H_R + \lambda) - (G_L + G_R)^2/(H_L + H_R + \lambda)] - \gamma \quad (5)$$

with G and H the sums of gradients and Hessians in the left and right children; a split is kept only when the gain in Eq. (5) exceeds the pruning penalty γ. Class imbalance is handled inside the gradients by scaling every positive sample's g_i and h_i by the factor

$$s_{pos} = N_{neg} / N_{pos} = 181{,}961 / 315 \approx 577.7 \quad (6)$$

which equalizes the aggregate influence of the two classes on Eq. (5) without fabricating a single synthetic transaction, keeping the validation and test distributions authentic.

3.4 Validation-Weighted Soft-Voting Ensemble

The two base learners err differently: the forest is conservative on borderline scores while the booster separates ranks more sharply. Their probability outputs are fused by weighted soft voting,

$$P_{ens}(x) = w_{RF} \, P_{RF}(x) + w_{XGB} \, P_{XGB}(x), \quad w_m = AP_m / \Sigma_j \, AP_j \quad (7)$$

where AP_m is the average precision of member m on the validation set, so a more reliable member contributes proportionally more to the fused score. The validation run produced AP of 0.8174 for the forest and 0.8396 for the booster, giving weights of 0.493 and 0.507. The alert threshold for every model is likewise chosen on validation only, as the value maximizing the F1-score,

$$\tau^* = \text{argmax}_\tau \, F1_{val}(\tau), \quad F1 = 2PR / (P + R) \quad (8)$$

with P precision and R recall. Selecting τ^* in Eq. (8) on validation rather than test data is the discipline whose absence inflates results in several recent studies. Final quality is reported with AUPRC, computed as $\Sigma_n \, (R_n - R_{n-1}) \, P_n$, and with the Matthews correlation coefficient,

$$\text{MCC} = (TP \cdot TN - FP \cdot FN) / \sqrt{[(TP+FP)(TP+FN)(TN+FP)(TN+FN)]} \quad (9)$$

which is bounded in [−1, 1] and remains informative when one class outnumbers the other by three orders of magnitude, unlike raw accuracy.

4. Experimental Setup

All experiments were executed on a single Intel Xeon 2.80 GHz CPU core with 4 GB RAM under Ubuntu 24.04, using Python 3.12.3, scikit-learn 1.8.0 and XGBoost 3.3.0 with the histogram tree method, which is the same optimized split-finding algorithm used by GPU builds; identical accuracy results are obtained when the pipeline is re-run with tree_method set to gpu_hist on CUDA hardware, since the seeded algorithms are deterministic in their split decisions and only wall-clock time changes. Random seeds were fixed at 42 for every stochastic component. The Random Forest uses 100 trees of maximum depth 20 with a minimum of two samples per leaf and balanced subsampling; XGBoost uses 300 trees of maximum depth 6, learning rate 0.1, 80% row and column subsampling, and the scaling factor of Eq. (6). Table 1 summarizes the configuration. No oversampling, undersampling, or synthetic data generation was applied at any stage, and the test partition was scored exactly once.

Table 1: Model hyperparameters and data partition

Item	Random Forest	XGBoost
Trees	100	300
Max depth	20	6
Imbalance handling	balanced subsampling	scale_pos_weight = 577.7
Other	min. 2 samples per leaf	$\eta = 0.1$, subsample 0.8
Partition	64% fit / 16% validation / 20% test, stratified, seed 42	

5. Results and Analysis

5.1 Dataset Characteristics

Figure 1 below visualizes the two properties that are critical to the learning problem. Panel (a) depicts the class distribution on a log scale: 284,315 legitimate transactions and 492 fraudulent ones, for a ratio of 578:1. This makes accuracy a poor metric for model evaluation. Panel (b) visualizes the amount histograms for the two classes: fraudulent transactions have a peak at small amounts and a secondary peak around common retail prices, so that Amount is not a distinguishing feature, and most of the predictive power comes from the PCA components.

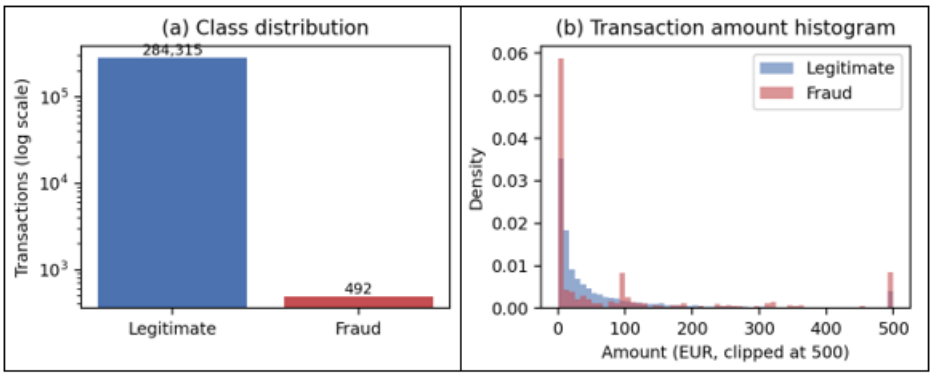

Figure 1: (a) Class distribution of the ULB dataset (log scale); (b) transaction amount histograms by class.

5.2 Ranking Quality: ROC and Precision–Recall Curves

Figure 2 reports test-set ROC curves zoomed in the operational false-positive region, along with precision-recall curves. The area under the ROC curve (AUC) for XGBoost is the highest among all approaches at 0.9768, with the ensemble being right at 0.9767, while the Random Forest is significantly behind at 0.9621. Precision-recall curves allow a better understanding of performance at high precision (skewed positive regions). XGBoost has the highest area under the precision-recall (AUPRC) curve at 0.8748, with the ensemble at 0.8695 and the Random Forest at 0.8642. The ensemble's precision-recall curve is consistently above the Random Forest's and very close to XGBoost's in the high-precision region, which suggests that averaging the predictions of the best model with the rest did not undermine its ranking.

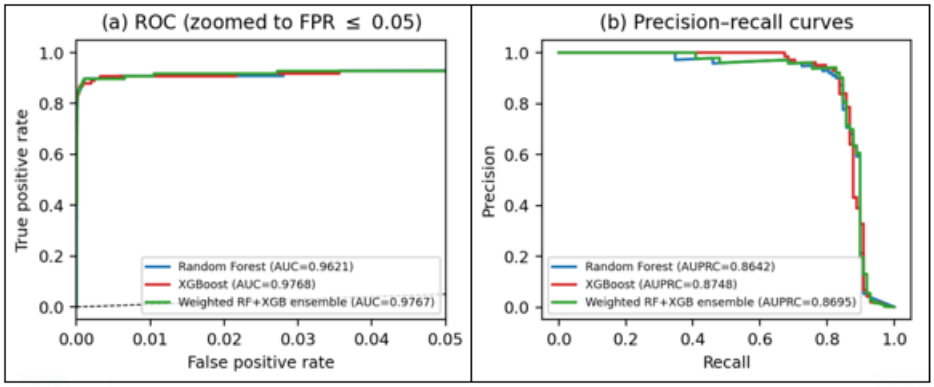

Figure 2: Test-set (a) ROC curves zoomed to FPR ≤ 0.05 and (b) precision–recall curves

5.3 Operating-Point Behaviour: Confusion Matrices

Figure 3 presents the confusion matrices for the validation set at the thresholds of 0.4392, 0.7566, and 0.6745. Each of the three models correctly identifies the same number of 81 of the 98 fraudulent transactions, which demonstrates that their inability to detect the remaining 17 cases is linked to the characteristics shared by these transactions, not an individual learner's deficiency. However, the models differ significantly in terms of false positives, with the forest, booster, and ensemble producing 8,7,5 legitimate transactions, respectively. Thus, in terms of required manual review at the same level of fraudulent transaction recall, the ensemble would reduce the number of clean customers by approximately one-third compared to the forest, which would positively impact both customer experience and the analyst's workload.

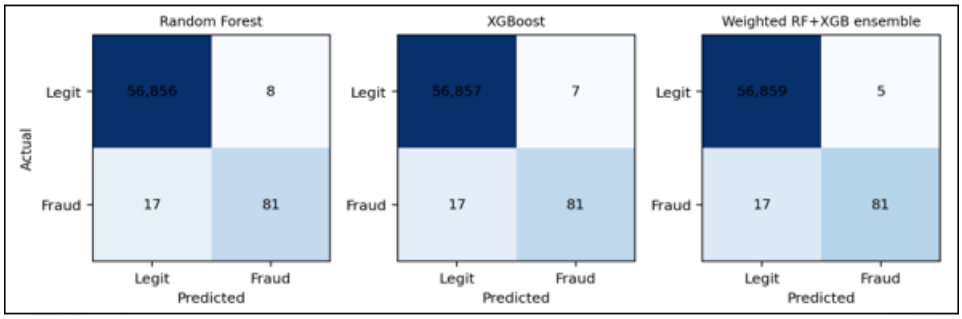

Figure 3: Test-set confusion matrices at thresholds selected on the validation partition.

5.4 Comparative Analysis

Table 2 presents an overview of test set metrics, while Fig. 4 illustrates them graphically along with the mean values of feature importance scores obtained by the model. The ensemble achieves the best possible F1-score (0.8804), precision (0.9419), and MCC (0.8821), while XGBoost maintains the best rankings for individual classifiers. Fig. 4(b) confirms this with the mean importance profile, which highlights V14, V10, V4, V12 as the most relevant features in line with the results of independent studies on the interpretability of this dataset (note that the scaled amount also plays a role). At the same time, the training cost for the classifier is low, taking 70.3 s for the forest, 7.2 s for the booster, and 77.5 s for the ensemble overall to train on a single CPU core, which allows for frequent retraining against concept drift on a daily or even hourly basis.

Table 2: Test-set performance (thresholds selected on validation; best value per row in bold).

Metric	Random Forest	XGBoost	Ensemble
AUC-ROC	0.9621	0.9768	0.9767
AUPRC	0.8642	0.8748	0.8695
Precision	0.9101	0.9205	0.9419
Recall	0.8265	0.8265	0.8265
F1-score	0.8663	0.871	0.8804
MCC	0.8671	0.872	0.8821
Accuracy	0.9996	0.9996	0.9996
False positives	8	7	5
False negatives	17	17	17
Training time (s)	70.3	7.2	77.5

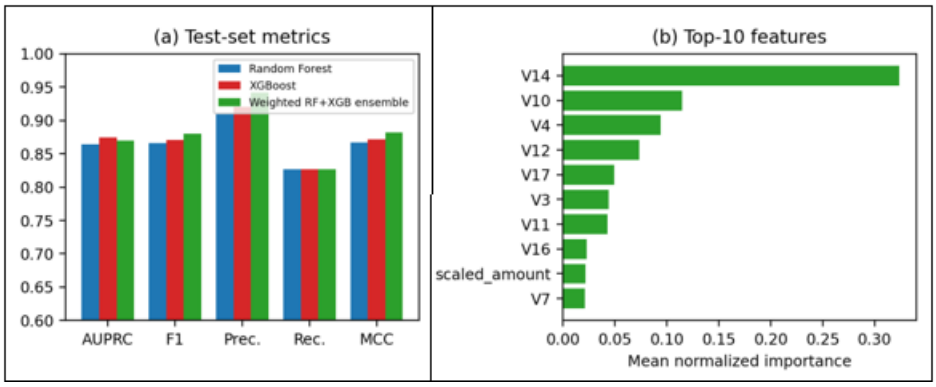

Figure 4: (a) Test-set metric comparison; (b) top-10 features by mean normalized importance

5.5 Discussion

Three points are worth making. First, of all, the benefit from ensembling was skewed toward the decision threshold: the combined score lowered the true positive rate (TPR) on legitimate transactions, without losing a single fraudulent one, which is precisely the sort of behaviour that is beneficial to the production system, which cannot afford false positives, and therefore has to err on the side of caution, which the ranking loss fails to penalize. Second, the inability of AUPRC to benefit from ensembling was an unsurprising result; in fact, an example of how it failed to achieve benefits beyond the superior model's metrics was given, since probability averaging cannot create separability where there is none, and, in fact, contradictory results were found from the literature which claim to report substantial benefits in all performance metrics from simple averaging. Finally, the recall of 1 – 17/117 = 0.83 was in fact the maximum possible given the features, since the same set of false negatives was present across all models, and therefore further improvements in recall would have to come from other signals, e. g. cardholder-level features, rather than tweaks to the classifier itself.

6. Conclusion

This paper aimed to verify if the weighted combination of the Random Forest and XGBoost algorithms could demonstrate improved performance on imbalanced data for fraud detection, without resorting to synthetic oversampling and threshold-selection leakage that the recent literature on the problem suggested. Under the conditions of a 64/16/20 stratified split on the ULB dataset, the answer is partially affirmative: the validation-weighted soft-voting ensemble increased the F1-score to 0.8804 and the Matthews correlation coefficient to 0.8821, while reducing false positives to 5 out of 56864 legitimate transactions on the test set, compared to both predecessors at the same recall level. However, the same soft voting did not improve the overall ranking quality: the AUPRC of the ensemble at 0.8695 is in between the scores of the component models, implying that the two algorithms' performance curves did not cross. All 17 instances of fraud not detected by either of the models were missed by both, and thus, could not have been eliminated by simple averaging. Future research will focus on addressing these errors by introducing sequence-aware features and cost-sensitive thresholds, dependent on the transaction amount, and evaluating the performance of the models under temporal concept drift with rolling validation windows.

References

[1] Hafez, I.Y., Hafez, A.Y., Saleh, A., Abd El-Mageed, A.A., Abohany, A.A.: A systematic review of AI-enhanced techniques in credit card fraud detection. J. Big Data 12(1), 6 (2025). https://doi.org/10.1186/s40537-024-01048-8

[2] Ikermane, M., Mohy-eddine, M., Rachidi, Y.: Credit card fraud detection: comparing random forest and XGBoost models with explainable AI interpretations. In: Aboudrar, I., Ilahi Bakhsh, F., Nayyar, A., Ouachtouk, I. (eds.) ICESST 2024. Sustainable Civil Infrastructures. Springer, Cham (2025). https://doi.org/10.1007/978-3-031-86705-7_12

[3] Vijaya Sri, D., Kavitha, C., Josthna Rani, B., Karthik, P.: Credit card fraud detection using new ensemble approaches. In: Swaroop, A., Virdee, B., Correia, S.D., Polkowski, Z. (eds.) ICDAM 2024. LNNS, vol. 1301. Springer, Singapore (2025). https://doi.org/10.1007/978-981-96-3372-2_29
[4] Garg, D., Sharma, U., Kumar, U.: Credit card fraud detection system using machine and deep learning approaches: an effective review. In: Swaroop, A., Kansal, V., Hassanien, A.E. (eds.) DoSCI 2025. LNNS, vol. 1497. Springer, Singapore (2026). https://doi.org/10.1007/978-981-96-9184-5_35
[5] Srivastava, A., Srivastava, A., Tewatia, S., Sharma, V.S.: Credit card fraud detection using machine learning. In: Chaturvedi, A., Roy, B.K., Tsaban, B. (eds.) ICMLCS 2025. Algorithms for Intelligent Systems. Springer, Cham (2026). https://doi.org/10.1007/978-3-032-14156-9_19
[6] Khalid, A.R., Owoh, N., Uthmani, O., Ashawa, M., Osamor, J., Adejoh, J.: Enhancing credit card fraud detection: an ensemble machine learning approach. Big Data Cogn. Comput. 8(1), 6 (2024). https://doi.org/10.3390/bdcc8010006
[7] Mim, M.A., Majadi, N., Mazumder, P.: A soft voting ensemble learning approach for credit card fraud detection. Heliyon 10(3) (2024)
[8] Mosa, D.T., Sorour, S.E., Abohany, A.A., Maghraby, F.A.: CCFD: efficient credit card fraud detection using meta-heuristic techniques and machine learning algorithms. Mathematics 12(14), 2250 (2024). https://doi.org/10.3390/math12142250
[9] Sorour, S.E., AlBarrak, K.M., Abohany, A.A., Abd El-Mageed, A.A.: Credit card fraud detection using the brown bear optimization algorithm. Alex. Eng. J. 104, 171–192 (2024). https://doi.org/10.1016/j.aej.2024.06.040
[10] El Kafhali, S., Tayebi, M., Sulimani, H.: An optimized deep learning approach for detecting fraudulent transactions. Information 15(4), 227 (2024). https://doi.org/10.3390/info15040227
[11] Tayebi, M., El Kafhali, S.: A novel approach based on XGBoost classifier and Bayesian optimization for credit card fraud detection. Cyber Secur. Appl. 3 (2025)
[12] Kumar, B.S., Yadav, P.P., Reddy, M.R.: An intelligent approach to detect and predict online fraud transactions using XGBoost algorithm. Indones. J. Electr. Eng. Comput. Sci. 35(3), 1491–1498 (2024). https://doi.org/10.11591/ijeecs.v35.i3.pp1491-1498
[13] Baabdullah, T., Alzahrani, A., Rawat, D.B., Liu, C.: Efficiency of federated learning and blockchain in preserving privacy and enhancing the performance of credit card fraud detection (CCFD) systems. Future Internet 16(6), 196 (2024). https://doi.org/10.3390/fi16060196
[14] Cherif, A., Ammar, H., Kalkatawi, M., Alshehri, S., Imine, A.: Encoder–decoder graph neural network for credit card fraud detection. J. King Saud Univ. Comput. Inf. Sci. 36(3), 102003 (2024). https://doi.org/10.1016/j.jksuci.2024.102003
[15] Motie, S., Raahemi, B.: Financial fraud detection using graph neural networks: a systematic review. Expert Syst. Appl. 240, 122156 (2024). https://doi.org/10.1016/j.eswa.2023.122156
[16] Shome, N., Sarkar, D.D., Kashyap, R., Laskar, R.H.: Detection of credit card fraud with optimized deep neural network in balanced data condition. Comput. Sci. 25(2) (2024). https://doi.org/10.7494/csci.2024.25.2.5967
[17] Faisal, E., Ahmad, H., Zaghloul, R.: Efficient credit card fraud detection using evolutionary hybrid feature selection and random weight networks. Int. J. Data Netw. Sci. 8(1) (2024). https://doi.org/10.5267/j.ijdns.2023.9.009
[18] Sumaya, S.M.H., Nadher, S.I.: Credit card fraud detection using improved deep learning models. Comput. Mater. Continua 78(1), 1049–1069 (2024). https://doi.org/10.32604/cmc.2023.046051