Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 2, 1391-1405
2025
Publisher: Learning Gate
DOI: 10.55214/25768484.v9i2.4651

An intelligent credit card fraud detection model using data mining and ensemble learning

Ahmed Samer Ismail AL-Dulaimi[1*], Islam R. Abdelmaksoud[2], Samir Abdelrazek[3], Hazem M. El-Bakry[4]
[1,2,3,4]Dept. of Information Systems, Faculty of Computers and Information, Mansoura University, Egypt;
ahmeddulaimy88@gmail.com (A.S.I.A.) islam-cis@mans.edu.eg (I.R.A.) samir.abdelrazek@mans.edu.eg (S.A.)
elbakry@mans.edu.eg (H.M.E.).

Abstract: The widespread use of credit cards has led to an increase in fraud. Credit card fraud detection involves identifying and preventing fraudulent transactions, either in real-time or post-occurrence. This paper seeks to create an advanced credit card fraud detection model via data mining. The proposed method comprises four essential steps: data acquisition, preprocessing, feature selection, and fraud detection. A recent balanced dataset is acquired, containing 28 anonymized features about the credit card transactions, along with the transaction amount and the transaction label (normal or fraud). The dataset is then explored to clean it and ensure its integrity. Feature selection is executed via the Energy Valley Optimization (EVO) metaheuristic method, employing the accuracy value of the Light Gradient Boosting Machine (LGBM) as the fitness function. This results in a 30% reduction in features. The reduced dataset is then input into the classification step, where an ensemble soft voting model is applied. This model encompasses Extra Trees, eXtreme Gradient Boosting (XGBoost), and Categorical Boosting (CatBoost) classifiers. The proposed model averages the probability of the three classifiers for each label and outputs the label with the highest average probability. The proposed method is assessed using recall, precision, accuracy, and F1-score, attaining 99.89%, 99.58%, 99.74%, and 99.74%, respectively. The proposed approach is evaluated against existing machine learning classifiers and relevant studies using the same dataset, showcasing enhanced performance and confirming its efficacy in identifying credit card fraud.

Keywords: *Credit card fraud detection, Energy valley optimization, Ensemble soft voting, XGBoost.*

1. Introduction

A credit card is a plastic card that holds personal information such as the cardholder's name, card number, and magnetic chip data, allowing the authorized individual to make purchases. Currently, credit card information can be accessed through various means, including automated teller machines, store barcode readers, and banks, and it is extensively utilized in online banking systems. Credit cards feature a distinct Card Verification Value (CCV) number, essential for security. The confidentiality of the card number and the physical security of the card are normally the pillar where the security of creit card rests on [1].

Numerous industries definitley headed toward modifications by E-services to substantially improve customer reach [2]. Employing credit cards for purchases was expedited and made simpler by the introduction of digital payment methods, like mobile wallets, contactless payments, and online payment platforms. Such systems aid in making financial transactions more convienent by providing an easy way to make online and in-store purchases [3].

The strive of the industry towards technological innovation continuously increased credit card acceptability, hence massive growth in credit card transactions has been reported in recent years. According to financial research specialist Raynor de Best, 188 billion Visa payment card purchases were

History: Received: 18 December 2024; Revised: 24 January 2025; Accepted: 27 January 2025; Published: 14 February 2025
* Correspondence: ahmeddulaimy88@gmail.com

made worldwide in 2020 [4]. Figure 1 depicts a surge in credit card transactions related to fraudulent activities. Unfortunately, this has translated to huge losses of almost US$32 billion in 2021, an increase of about 17% over the previous year [4]. Therefore, this increases in fraudulent activities in credit card transactions has gradually presented more difficulties and higher financial costs.

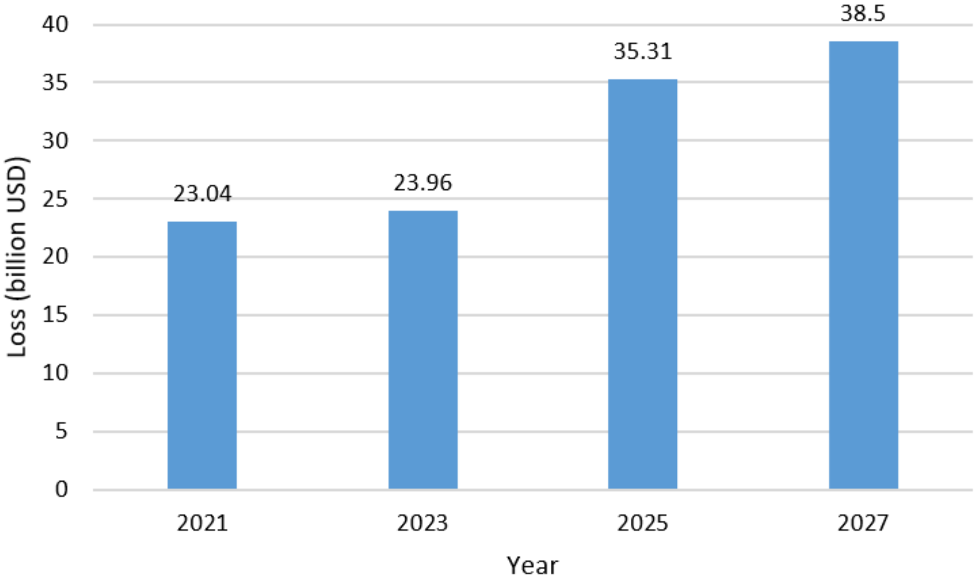


Figure 1.
Fraudulent transaction losses, including past and projected losses.

The main problem in e-commerce transactions is that the physical card and the cardholder are not present during the transaction process [2]. This presents a problem for retailers in verifying the authenticity of the person making the transaction, thus leaving room for fraudsters. Credit card fraud is one of the most common issues faced by consumers, businesses, and financial institutions all over the world. It manifests when unauthorized transactions use stolen or compromised credit card information, leading to economic losses and inconvenience for the victims [3].

To deal with this escalating threat, the domain of Credit Card Fraud Detection (CCFD) [5] has emerged as a vital part of modern finance security. Credit card fraud detection refers to the identification and prevention of fraudulent credit card transactions either in real-time or post-occurrence. It employs data mining methods and machine learning algorithms to look at trends, find out abnormal behavior, and identify other characteristics of fraud. The aim is to reduce the financial loss, protect the cardholder, and retain the trust and integrity in the payment system.

This research tends to enhance a CCFD model by adopting data mining and machine learning (ML) techniques. The study uses the "Credit Card Fraud Detection Dataset 2023 (CCFD2023)", the most recent dataset available. The proposed system can be used in banks to prevent fraudulent transactions and approve only legitimate ones. The study has the following contributions:

- The Energy Valley Optimization (EVO) metaheuristic method is used to identify the most critical features for the fraud detection model, hence augmenting model interpretability and optimizing the efficiency of the detection process.
- The accuracy of the Light Gradient Boosting Machine (LGBM) is utilized as the objective function for the EVO optimization procedure, enhancing the feature selection results and the model robustness.

- An ensemble soft voting classification model is developed, integrating Extra Trees, extreme Gradient Boosting (XG Boost), and Categorical Boosting (Cat Boost) classifiers. The model's efficacy in fraud detection was demonstrated by its 99.74% accuracy.
- The performance of the proposed system was compared to other standard machine learning models and other studies using the same dataset. The results indicated that the proposed system outperforms all alternatives.

The subsequent sections of the paper are structured as follows: The subsequent section examines related research that has employed the same dataset. The third section discusses the proposed system methodology in detail, incorporating data analysis and exploration, feature selection, and classification. The fourth section evaluates the proposed system, compares it with previous ML models, and benchmarks the results against other related studies. The final section concludes the study, which also outlines prospective future directions.

2. Related Work

This paper presents the related work on the CCFD2023 dataset for developing the CCFD models using ML techniques. The work by Zheng [6] adopted the KNN model in credit card fraud prediction. In that work, initial pre-processing was done with repetition and gap testing, data normalizing, and outlier identification on the CCFD2023 dataset. After that, the dimensionality reduction procedure was carried out with PCA to find the contribution rate of every component to the variance. Then, at what point the cumulative contribution rate first exceeded 95%, the number of features was recorded. Lastly, classification was done with the KNN model, and from that, the resultant accuracy obtained was 99.41%.

Iqbal, et al. [7] came up with a fraud detection model that utilized the CCFD2023 dataset. Normalization, cleaning of data, timestamping, and periodic disintegration of patterns were performed. They proposed the best statistical method for feature selection for optimum dimensionality reduction. At the end, they used DAGMM for classification, which yielded an accuracy of 51.85%.

Yadav, et al. [8] performed an application of Apache Spark in credit card fraud detection. The high performance of the Spark framework enabled fast processing and computation over big data that is kept in HDFS format. The monitoring process was done with Prometheus and Grafana: Prometheus collected system metrics and stored them; Grafana was used to visualize the analytics. Various experiments have been conducted by running different thresholds for logistic regression. These experiments indicated that the threshold of 0.6 outperformed the rest. Their overall system achieved 95% accuracy.

Huang, et al. [9] implemented an LSTM model that featured adaptive mechanisms in the form of an Adam optimization method and adaptive regularization method-Dropout. These mechanisms dynamically adjusted the learning rates and dropout probabilities. In the LSTM model, attention was added to dynamically calculate weights of different features, especially the ones that play a significant role in credit fraud. The proposed model was evaluated on the CCFD2023 dataset, which is divided into a 70% training set and a 30% test set. It achieves an accuracy of 95% on the test set.

Eriksson and Jakobsson [10] made a comparison of Python, Scala and Lua for machine learning tasks. They used the CCFD2023 dataset to build a fraud detection model and they concluded that Python is the best. Apache Spark was used for data processing. Logistic regression and multilayer perceptron neural networks were used for classification and the best accuracy of 96.03% and 95.86% were obtained respectively. Han, et al. [11] suggested Any Loss which is a confusion matrix based metric translated into a loss function. To make the confusion matrix differentiable, an approximation function was used to describe the confusion matrix in terms of differentiable elements. To assess their proposed loss function, they used it in single-layer and multilayer perceptron networks for classification on the CCFD2023 dataset. These fraud detection models had the accuracies of 99% and 99.4%, respectively. In each study, Table 1 summarizes the related work, showing dimensionality reduction techniques, classification methods, and achieved performance.

Table 1.
Summary of the related work using the CCFD2023 dataset, showing dimensionality reduction techniques, classification methods, and achieved accuracies.

Study	Dimensionality reduction	Classification	Accuracy (%)
Zheng [6]	PCA	KNN	99.41
Iqbal, et al. [7]	Optimal statistical method	DAGMM	51.85
Yadav, et al. [8]	N/A	Logistic regression	95
Huang, et al. [9]	N/A	Adaptive attention LSTM	95
Eriksson and Jakobsson [10]	N/A	Logistic regression	96.03
		Multilayer perceptron	95.86
Han, et al. [11]	N/A	Single layer perceptron (Anyloss)	99
		Multilayer perceptron (Anyloss)	99.4

3. Materials and Methods

This section offers a comprehensive explanation of the proposed system. The proposed system has four main steps, as shown in Figure 2. In the first step, the CCFD2023 dataset is acquired. The second step involves the exploitation and analysis of the dataset, which includes noise handling, feature distribution examination, and correlation analysis. Feature selection is performed in the third step using the EVO algorithm, optimized by the LGBM accuracy as the objective function. The fourth step is classification, which employs an ensemble soft voting model combining the ET, XG Boost, and Cat Boost classifiers. When the system processes a transaction, it is classified as normal or fraudulent. If the transaction is normal, the system allows it to be executed. However, if the system detects fraud, the transaction is blocked, and the bank is notified to take further action. The following subsections provide a more detailed discussion of each step.

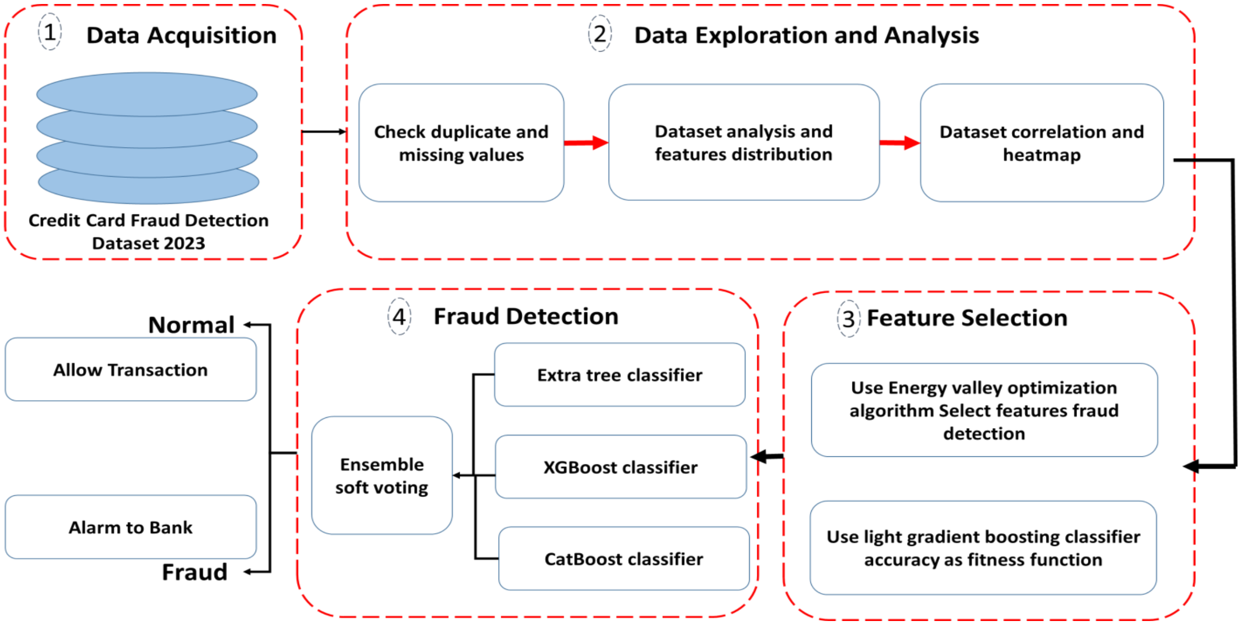
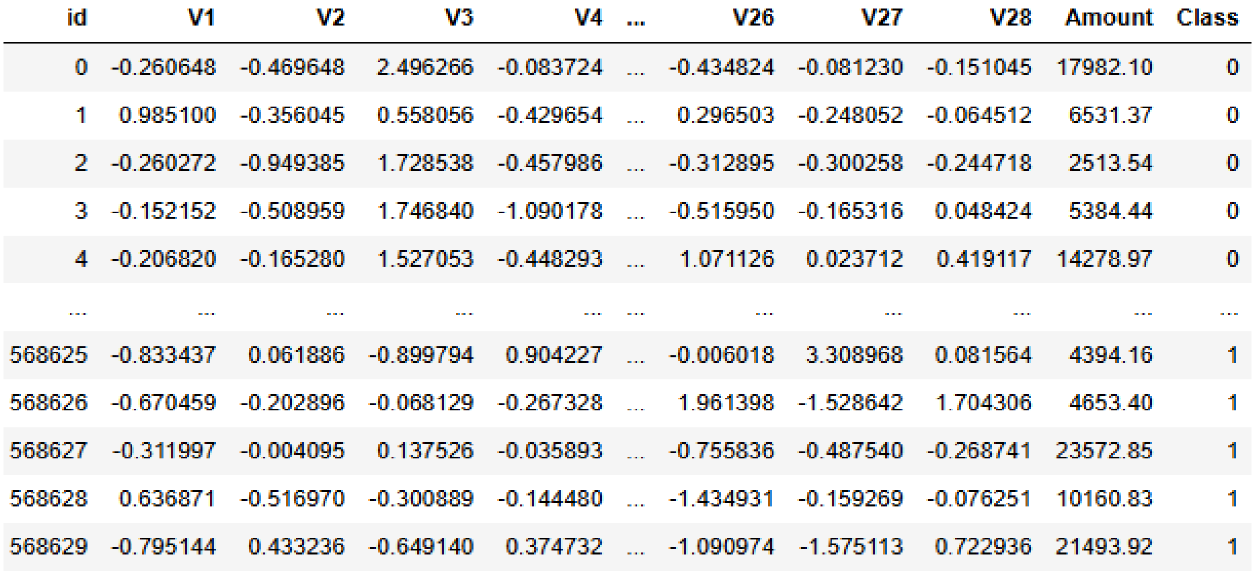

1 Data Acquisition
Credit Card Fraud Detection Dataset 2023
2 Data Exploration and Analysis
Check duplicate and missing values
Dataset analysis and features distribution
Dataset correlation and heatmap
3 Feature Selection
Use Energy valley optimization algorithm Select features fraud detection
Use light gradient boosting classifier accuracy as fitness function
4 Fraud Detection
Extra tree classifier
XGBoost classifier
CatBoost classifier
Ensemble soft voting
Normal
Allow Transaction
Alarm to Bank
Fraud

Figure 2.
The proposed model: Key steps include data acquisition, analysis, feature selection, and classification.

4. Data Acquisition

Figure 3 shows part of the CCFD2023 dataset [12] comprising 2023 European cardholder credit card transactions. The dataset contains 568,630 anonymized records, safeguarding cardholders' information while fulfilling its intended function. Each record has 31 columns, including one binary classification column that indicates if the transaction is fraudulent. The columns comprise four key features: id, V1-V28, amount, and class. These features are described in Table 2.



id	V1	V2	V3	V4	...	V26	V27	V28	Amount	Class
0	-0.260648	-0.469648	2.496266	-0.083724	...	-0.434824	-0.081230	-0.151045	17982.10	0
1	0.985100	-0.356045	0.558056	-0.429654	...	0.296503	-0.248052	-0.064512	6531.37	0
2	-0.260272	-0.949385	1.728538	-0.457986	...	-0.312895	-0.300258	-0.244718	2513.54	0
3	-0.152152	-0.508959	1.746840	-1.090178	...	-0.515950	-0.165316	0.048424	5384.44	0
4	-0.206820	-0.165280	1.527053	-0.448293	...	1.071126	0.023712	0.419117	14278.97	0
...	...	...	...	...	...	...	...	...	...	...
568625	-0.833437	0.061886	-0.899794	0.904227	...	-0.006018	3.308968	0.081564	4394.16	1
568626	-0.670459	-0.202896	-0.068129	-0.267328	...	1.961398	-1.528642	1.704306	4653.40	1
568627	-0.311997	-0.004095	0.137526	-0.035893	...	-0.755836	-0.487540	-0.268741	23572.85	1
568628	0.636871	-0.516970	-0.300889	-0.144480	...	-1.434931	-0.159269	-0.076251	10160.83	1
568629	-0.795144	0.433236	-0.649140	0.374732	...	-1.090974	-1.575113	0.722936	21493.92	1

Figure 3.
Snapshot of the CCFD2023 dataset showing dataset features and values.

Table 2.
Description of the features in the CCFD2023 dataset.

Feature	Description
Id	The identification column serves as the unique identifier for every transaction.
V1- V28	These columns denote diverse transaction characteristics (including time, location, etc.) via 28 anonymous features.
Amount	This column denotes the transaction quantity.
Class	Whether the transaction was fraudulent is indicated by the binary label in the last column (0 for normal transactions and 1 for fraud ones).

The dataset is balanced, with an equal number of fraudulent and normal transactions totaling 284315 records. For faster data processing, the dataset underwent a reduction step, retaining only 5% of the records, resulting in a reduced dataset with 28432 records, where each label contains 14216 records. Table 3 shows this distribution.

Table 3.
Dataset distribution before and after data reduction.

Before data reduction		After data reduction	
Class	**Count**	**Class**	**Count**
Fraud	284315	Fraud	14216
Normal	284315	Normal	14215

5. Data Exploration and Analysis

Duplicate values may introduce bias into statistical analysis, and missing values may distort modeling findings. Maintaining the correctness and integrity of the data is essential for the dependability of the findings. An examination of the dataset showed the absence of duplicate or missing information, indicating good data completeness.

Figure 4 shows the correlation matrix, presented as a heatmap, which displays the extent of linear dependencies among different features of the dataset. The cells in the matrix are color-coded, with shades tending toward blue or red, indicating higher linear dependencies between two features. The figure shows that most features exhibit positive or negative correlations. This justifies the need for a feature selection step to retain only essential features while removing redundant ones.

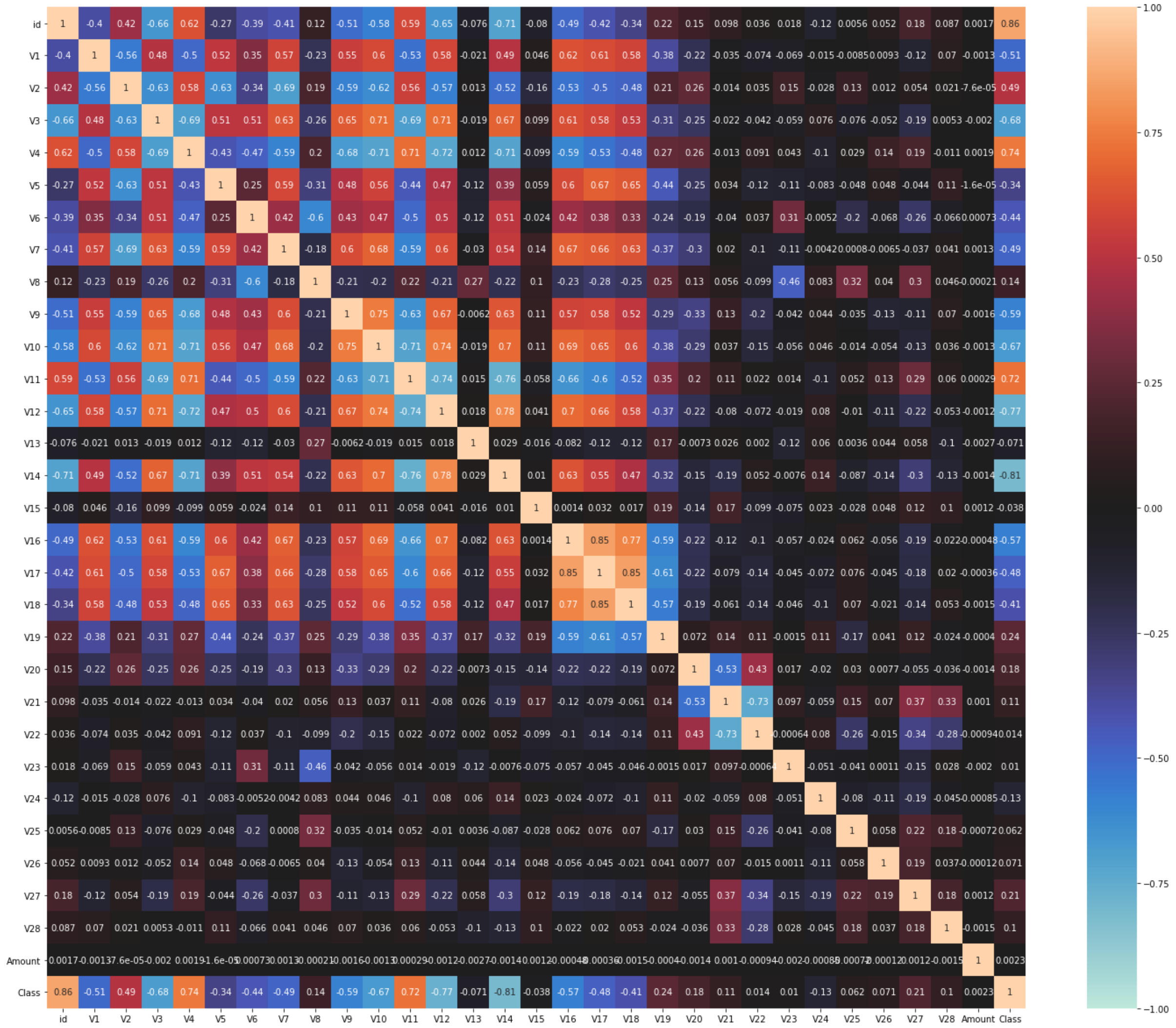


Figure 3.
Heatmap representing feature correlations in the CCFD2023 dataset.

Figure 5 shows the distribution of dataset features. Most features exhibit Gaussian-like distribution, indicating the data is normally distributed and contains minimal outliers. The Amount feature, however, displays the most diverse histogram, reflecting the real-world variability of transaction amounts. This analysis confirms that the dataset is clean and does not require further preprocessing.

DOI: 10.55214/25768484.v9i2.4651

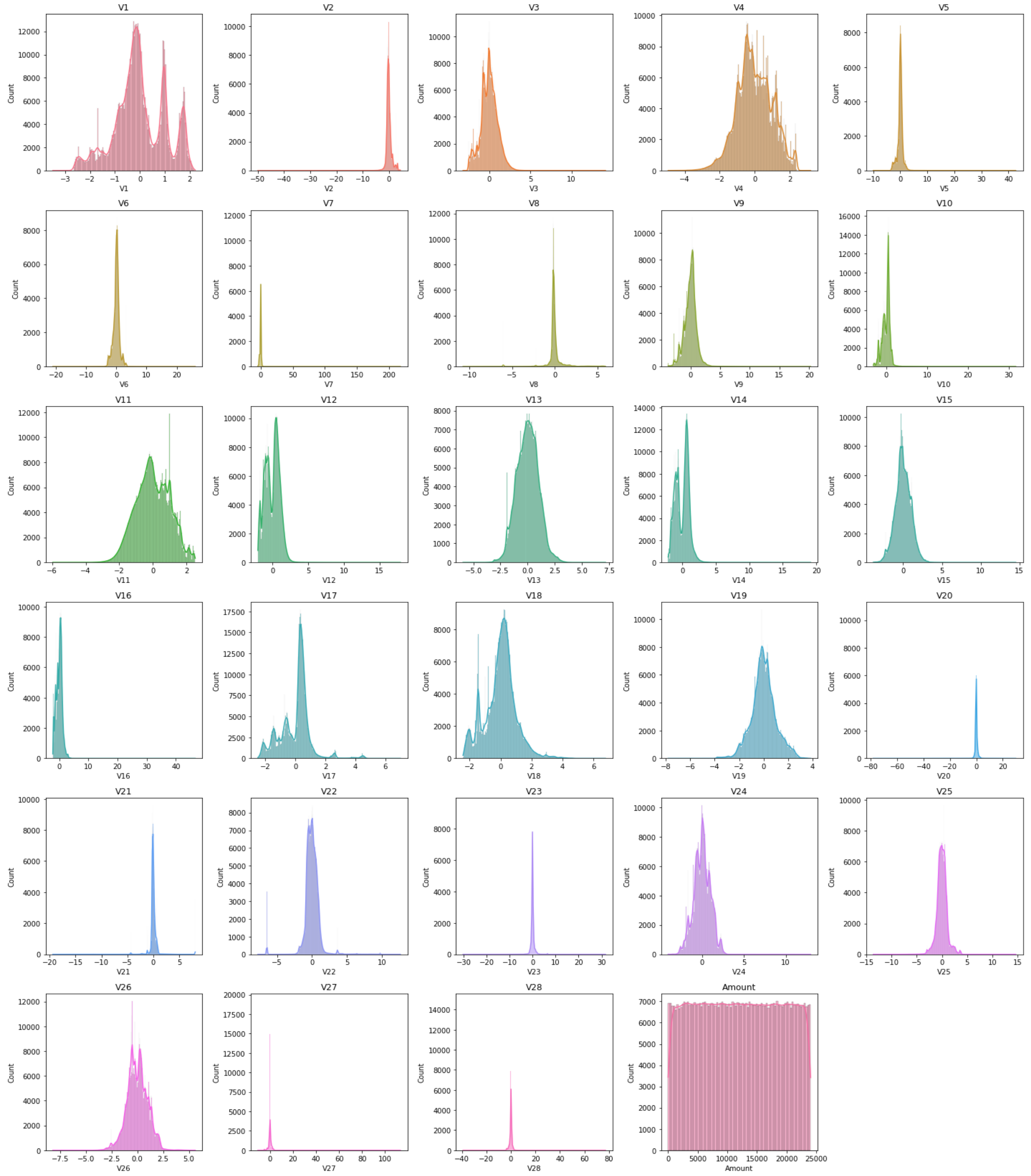


Figure 4.
Distribution of the CCFD2023 dataset features.

6. Feature Selection Using EVO

This study employed the EVO algorithm [13] as a feature selection technique. EVO is a recent metaheuristic optimization algorithm inspired by advanced physics principles. It mimics the degradation process caused by different particles in nature. In EVO, each particle, with varying levels of stability, represents a possible solution. Algorithm 1 shows the step-by-step implementation of the EVO algorithm that takes the dataset as input and outputs a reduced dataset with selected features.

Algorithm 1: EVO Metaheuristic algorithm for feature selection

```
Input: Population of selected feature solutions ($X$), size of the population ($n$), LGBM accuracy function ($F$)
Output: Selected features from the best solution particle
Randomly initialize a new population of features as initial solution particles $X_i$
Compute accuracy that is generated from the LGBM algorithm for each $X_i$ using $F$ as Neutron
While the stop condition is not met
    Determine the enrichment bound ($EB$) of the particle
    Determine the particle with the best stability level ($X_{BS}$)
    For I= 1 to n
        Determine Neutron enrichment level ($NEL_i$) and stability level ($SL_i$) of the ith particle
        If $NEL_i > EB$
            Determine the stability bound of the particles ($SB$)
            Determine a neighboring particle ($X_{Ng}$)
            If $SL_i > SB$
                Generate Alpha Index I and II
                For j= 1 to Alpha index II
                    Generate new position vector: $X_i^{new1} = X_i\left(X_{BS}X_i^j\right)$
                End for
                Generate Gamma Index I and II
                For j= 1 to Gamma index II
                    Generate new position vector: $X_i^{new2} = X_i\left(X_{Ng}X_i^j\right)$
                End for
            Else
                Determine the center of particles $X_{CP}$
                Define random values between 0 and 1 $r_1$, $r_2$, $r_3$, and $r_4$
                Generate new position vector: $X_i^{new1} = X_i + \frac{r_1X_{BS}- r_2X_{CP}}{SL_i}$
                Generate new position vector: $X_i^{new2} = X_i + r_3X_{BS} - r_4X_{NG}$
            End If
        Else
            Define random value between 0 and 1 $r$
            Generate new position vector: $X_i^{new} = X_i + r$
        End If
    End For
End While
Return the particle with the best $X_{BS}$
```

LGBM [14] accuracy is employed as fitness function evaluation in the EVO algorithm. The classification in LGBM is an improved gradient-boosting decision tree (GBDT) algorithm with additional gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB) [15]. GOSS prioritizes training samples with larger gradients to reduce the computational complexity in LGBM and achieve accurate gain estimation. The quantity of features is reduced by EFB, which bundles scarce, mutually exclusive features.

Given training data and loss function, the learning model is defined as a problem of loss function minimization. LGBM adopts leaf-wise growth with depth limitation and histogram strategy based on

GBDT [16]. A leaf-wise strategy involves selecting the leaf node with the highest gain to expand in each node division. In contrast to the conventional GBDT's level-wise approach, leaf-wise strategies achieve greater accuracy within the same division periods, as illustrated in Figure 6.

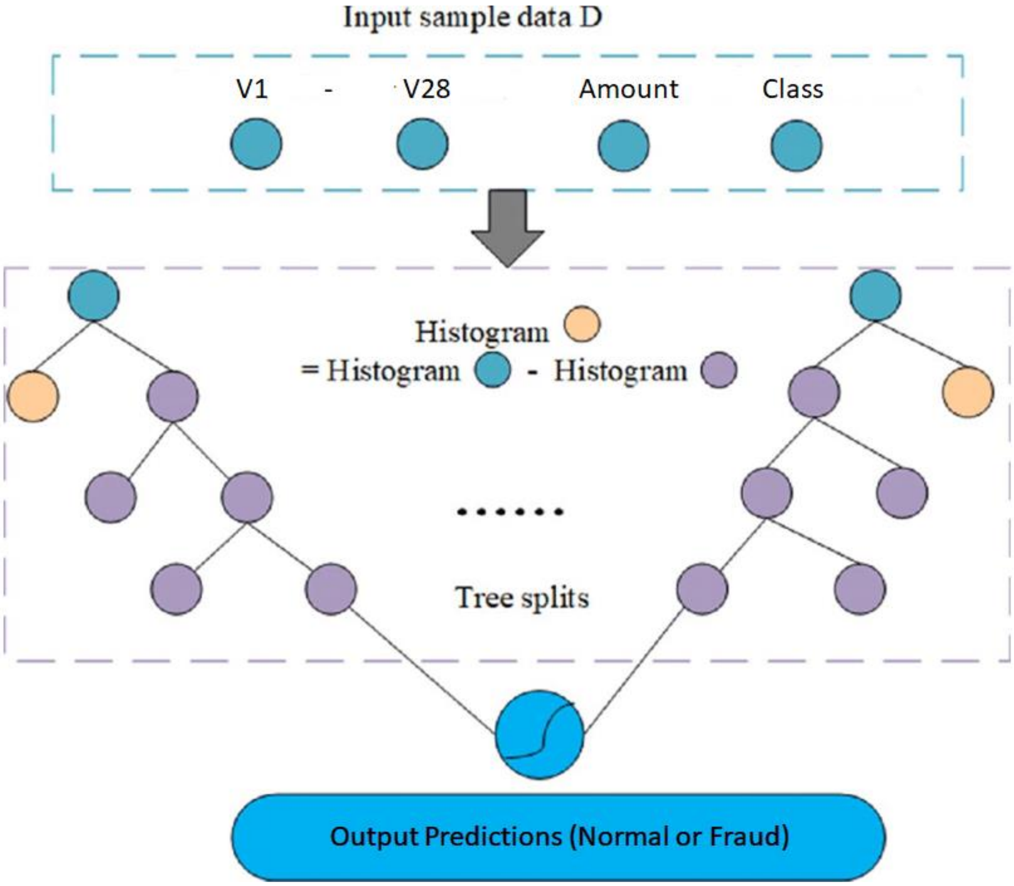


Figure 5.
LGBM leaf-wise method for credit card fraud prediction.

The histogram method [17] accelerates the training process by discretizing a large number of continuous input variables into k unique values and constructing a histogram with width k [18, 19]. The histogram of the parent node can be subtracted from the histogram of the brother node to derive a leaf histogram. In Figure 6, the histogram (orange node) equals the parent node histogram (blue node) minus the sibling node histogram (purple node).

EVO reduced the dataset's dimensionality by selecting 21 out of 29 features, resulting in a 30% reduction. The selected features are: 'V2', 'V3', 'V4', 'V5', 'V8', 'V10', 'V11', 'V12', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V21', 'V22', 'V25', 'V26', 'V27', 'V28', and 'Amount'. Using the designated features, the LGBM model obtained a fitness evaluation of 99.7% accuracy on the CCFD2023 dataset. The dataset was partitioned into 80% training and 20% assessment data. EVO was initialized with a population size of 50 and executed over 10 epochs, converging on the optimal feature set by epoch 5.

7. Fraud Detection

For fraud detection, an ensemble soft voting setup was implemented using three algorithms: Extra Trees [20] XG Boost [20] and Cat Boost [21, 22]. The Algorithm for the classification process is presented in Algorithm 2 and Figure 7. Soft voting uses the class probabilities output by each model rather than direct class predictions. The final prediction result is the category with the highest mean class probability, determined for each category (normal or fraud).

```
Algorithm 2: Ensemble soft voting algorithm for fraud classification
    Input: Training features ($X_{train}$), Training labels ($y_{train}$), testing features ($X_{test}$).
    Output: Prediction $\hat{y}$ for $X_{test}$.
    Initialize models: Extra Trees Classifier $C_{ET}$, XGBoost Classifier $C_{XGB}$, and CatBoost Classifier $C_{Cat}$.
    Train models using training data ($X_{train}, y_{train}$).
    Compute class prediction probabilities from each trained model: $Pr_{ET}$ , $Pr_{XGB}$ , $Pr_{Cat}$.
    Compute the average probability for each class prediction: $Pr = \frac{Pr_{ET} + Pr_{XGB} + Pr_{Cat}}{3}$
    Assign final prediction $\hat{y}$ to the class with the highest probability.
    Return $\hat{y}$.
```

XG Boost is an ensemble learning model designed for machine learning problems. It employs a systematic approach to integrate predictions from multiple decision trees to reduce the objective function θ, as defined in Equation 1.

$$J(\theta) = \sum_{i=1}^{m}\left(L(y_i - \hat{y}_i) + \sum_{k=1}^{K}\Omega(f_k)\right) \qquad (1)$$

where $L(y_i - \hat{y}_i)$ represents the loss function quantifying prediction errors, while $\Omega(f_k)$ represents the regularization component of the kth tree. The parameter m represents the number of samples, and K represents the total number of trees.

Cat Boost utilizes a unique permutation-based methodology in its boosting process, guaranteeing fault rectification in each successive tree while preventing redundant learning from identical samples.

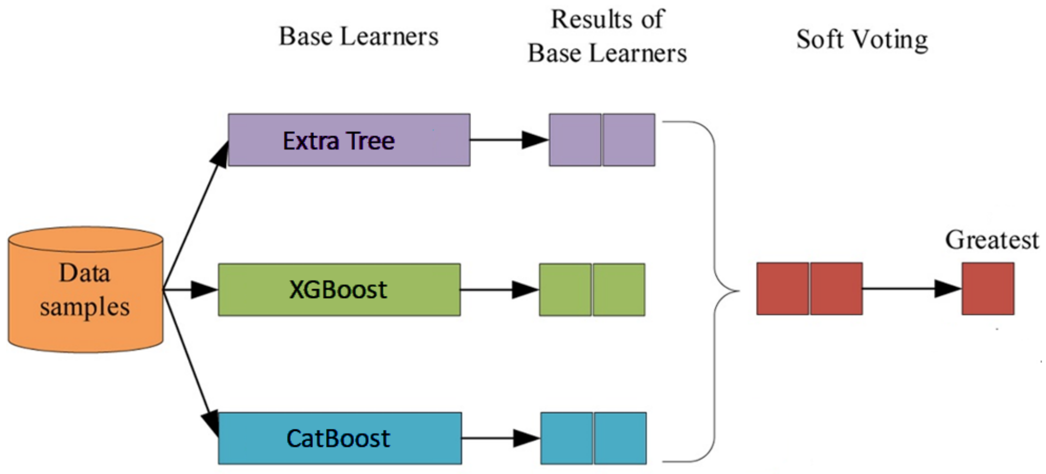


Figure 6.
Proposed ensemble soft voting classification model.

Extra Trees is an ensemble learning methodology defined by the following principal steps:

- Selecting random subsets of data points for each decision tree.
- Independently developing several decision trees using these subgroups.
- Adopting preponderance voting to aggregate predictions.

The ensemble soft voting approach undoubtedly diminishes subjective predictions via adopting the certainty of each model's output probabilities rather than relying on final decisions.

8. Experimental Results

The results of the proposed system are presented, discussed, and contrasted with those of previous studies in this section. The performance of the proposed system is analyzed, and the evaluation metrics employed are discussed. Additionally, this section shows how the proposed system outperformed recent works and other machine learning algorithms, validating its effectiveness.

9. Evaluation Metrics

4 measures are used in the evaluation of the performance of the proposed system: accuracy, F1-score, recall, and precision. The metrics are calculated using the values of True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) from the system's prediction. In this case, TP is the fraudulent transactions that the system was able to identify as fraud. FP is the normal transactions that have been wrongly flagged as fraud. TN is the normal transactions that have been properly identified as normal. Finally, FN is the fraud transactions that have been wrongly flagged as normal.

It is the ratio of correct positive predictions to the total number of predictions made within that class. A low precision value means that there are many false positives.

$$Precision = \frac{TP}{TP\ +\ FP} \tag{2}$$

Recall (Equation 3): The proportion of accurate positive classifications to the total number of samples within that class. High recall signifies a minimal quantity of misclassified instances.

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

F1-score (Equation 4): calculates the harmonic mean of precision and recall. This statistic often indicates the strength of the classification model.

$$F1 - score = \frac{2\ *\ Precison\ *\ Recall}{Precision\ +\ Recall} \tag{4}$$

Accuracy (Equation 5): The ratio of accurately categorized samples to the total number of samples.

$$Accuracy = \frac{TP + TN}{TP\ +\ FP +\ FN\ + TN} \tag{5}$$

These measures thoroughly assess the system's performance, emphasizing its capacity to detect fraudulent transactions accurately.

10. System Results

The experimental results were obtained using a PC with a 2GB Nvidia GeForce MX130 GPU, 32 GB RAM, and an Intel Core i5 CPU. The Python 3.8.8 programming language, as well as the sklearn (version 1.3.2), cat boost (version 1.2), and lightgbm (version 3.3.2) Python libraries are employed to develop and execute the proposed approach. As illustrated in Table 4, the dataset was partitioned into a training set covering 80% of the data and a test set covering 20%. The models were developed using the training set, and the trained model was evaluated using the test set. The TP, FP, TN, and FN values were calculated by comparing the predictions of the proposed model on the test set to the original transaction results. This enabled the calculation of the evaluation metrics.

Table 4.
Training and testing labels count after data split.

Training labels		**Testing labels**	
Class	**Count**	**Class**	**Count**
Fraud	11327	Fraud	2844
Normal	11327	Normal	2843

Before feature selection, the proposed ensemble model was evaluated against ten other machine learning algorithms. The ensemble model achieved superior performance, as shown in Table 5, with an accuracy of 99.77%, precision of 99.65%, recall of 99.89, and F1-score of 99.77%. The results confirm the proposed system's robustness and efficacy.

Table 5.
Results before feature selection.

Algorithm	Precision	Recall	F1-score	Accuracy
Ensemble ET, XGB and Cat	0.9965	0.9989	0.9977	0.9977
Gradient boosting [21]	0.9864	0.9726	0.9795	0.9796
AdaBoost [22]	0.9774	0.9592	0.9682	0.9685
Support vector machine [23]	0.5057	0.4669	0.4856	0.5052
Logistic regression [24]	0.9754	0.948	0.9615	0.962
Bagging [25]	0.9908	0.987	0.9889	0.9889
K- nearest neighbor [26]	0.8305	0.6617	0.7366	0.7633
Gaussian naive bayes [27]	0.9737	0.8579	0.9121	0.9174
Decision trees [28]	0.9733	0.9852	0.9792	0.9791
Quadratic discriminant analysis [29]	0.9847	0.9262	0.9545	0.9559
Linear discriminant analysis [30]	0.9755	0.923	0.9485	0.9499

Among the evaluated algorithms, the K-Nearest Neighbor (KNN) Algorithm performed the worst, with an accuracy of 73.66%. At the same time, the Bagging Classifier achieved the best results among non-ensemble-based algorithms, with an accuracy of 98.89%. Additionally, ensemble-based classifiers such as Gradient Boosting, AdaBoost, and the bagging classifiers performed better than non-ensemble methods, justifying the use of this genre in the proposed ensemble soft voting system.

Feature selection was performed using the EVO algorithm to select the salient features for fraud detection. Several other metaheuristic algorithms were considered, such as Chernobyl Disaster Optimization (CDO) [23] Fick's Law Algorithm (FLA) [24] Physical Phenomenon of RIME-ice (RIME) [25] Coati Optimization Algorithm (CoatiOA) [26] and Osprey Optimization Algorithm (OOA) [27]. However, the EVO algorithm outperformed others in reducing dataset dimensionality while achieving the best fitness function. The EVO algorithm allowed the system to maximize the LGBM accuracy to 99.7%, comparable to the results before feature selection. The EVO algorithm selected the most important features without compromising performance. Figure 8 shows a comparison of the EVO algorithm with other optimization techniques.

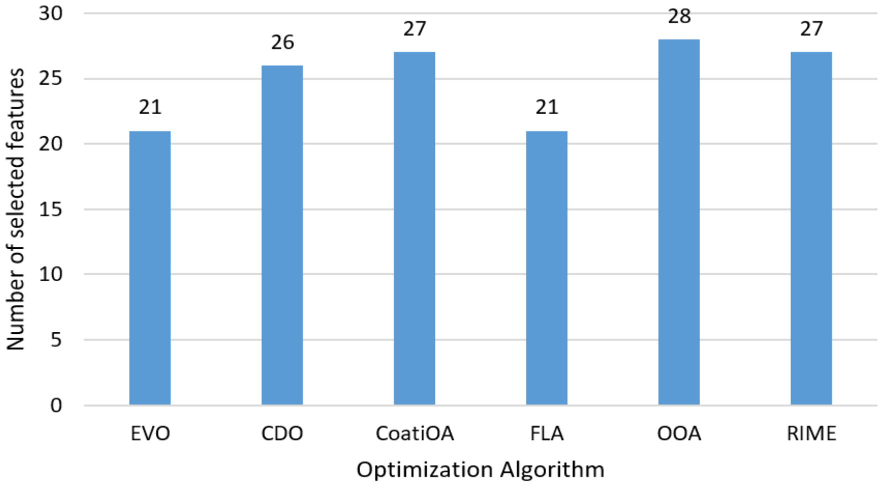


Figure 7.
Comparing EVO to other recent metaheuristic optimization algorithms.

After feature selection, the proposed ensemble model was evaluated against ten other machine learning algorithms. The proposed system retained its high performance, as shown in Table 6, achieving an accuracy of 99.74%, precision of 99.58%, recall of 99.89, and an F1-score of 99.74%. The decrease in the results compared to pre-feature selection was minimal (0% for recall, 0.07% for precision, and 0.03% for accuracy and F1-score). These results validate that feature selection improved system interpretability and efficiency without compromising robustness.

Table 6.
Results after feature selection.

Algorithm	Precision	Recall	F1-score	Accuracy
Ensemble ET, XGB and Cat	0.9958	0.9989	0.9974	0.9974
Gradient Boosting [28]	0.9843	0.9684	0.9762	0.9764
AdaBoost [29]	0.9777	0.9543	0.9658	0.9662
Support Vector Machine [30]	0.5055	0.4655	0.4847	0.505
Logistic Regression [18]	0.9751	0.9219	0.9478	0.9492
Bagging [31]	0.9884	0.9866	0.9875	0.9875
K- Nearest Neighbor [32]	0.8219	0.6343	0.716	0.7484
Gaussian Naive Bayes [33]	0.9781	0.8646	0.9179	0.9226
Decision Trees [34]	0.9737	0.9887	0.9812	0.981
Quadratic Discriminant Analysis [35]	0.9791	0.9205	0.9489	0.9504
Linear Discriminant Analysis [19]	0.9808	0.9153	0.9469	0.9487

Among the evaluated algorithms, SVM performed the worst, with an accuracy of 50.5%, while the Bagging Classifier remained the best performer among these methods, with an accuracy of 98.75%. The proposed system also outperformed previous studies using the CCFD2023 dataset, as demonstrated in Figure 9.

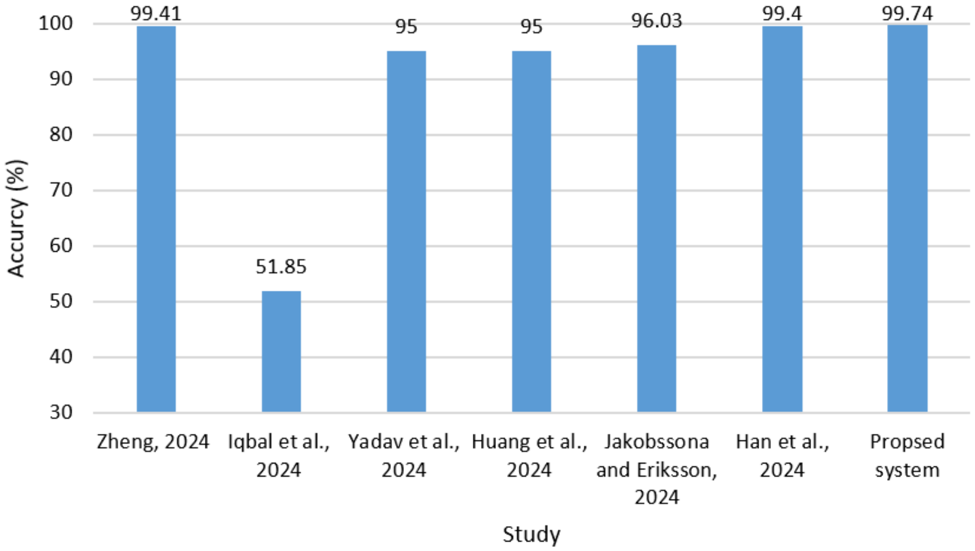


Figure 8.
Comparing the proposed system with previous studies using the CCFD2023 dataset.

11. Conclusion

A novel model for the detection of credit card fraud was suggested in this study. The proposed model comprises Data collection, preprocessing, feature selection, and classification. The proposed framework was trained using a recent balanced dataset of 28 anonymized attributes. To confirm its integrity, the dataset was cleansed. Feature selection was executed with the EVO metaheuristic approach, with the accuracy of the LGBM serving as the fitness function, resulting in a 30% reduction in features. The classification phase utilized an ensemble soft voting model that integrated ET, XGBoost, and CatBoost classifiers, achieving 99.74% accuracy. Comparative analysis showed that the proposed system outperformed existing machine learning classifiers and related studies using the same dataset, demonstrating its efficacy in identifying credit card fraud. Future research should concentrate on integrating explainable AI techniques to enhance model transparency and the development of adaptive models to compensate for the evolving patterns of fraud and concept drift.

Transparency:

The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

Copyright:

References

[1] A. Agrawal, S. Kumar, and A. K. Mishra, "A novel approach for credit card fraud detection," presented at the 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), IEEE, 2015.

[2] L. Borah, B. Saleena, and B. Prakash, "Credit card fraud detection using data mining techniques," *The Seybold Report*, vol. 15, pp. 2431–2436, 2020.

[3] R. Bin Sulaiman, V. Schetinin, and P. Sant, "Review of machine learning approach on credit card fraud detection," *Human-Centric Intelligent Systems*, vol. 2, no. 1, pp. 55-68, 2022. https://doi.org/10.1007/s44230-022-00004-0

[4] A. Mniai, M. Tarik, and K. Jebari, "A novel framework for credit card fraud detection," *IEEE Access*, vol. 11, pp. 112776–112786, 2023. https://doi.org/10.1109/ACCESS.2023.3323842

[5] 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, p. 6, 2024. https://doi.org/10.3390/bdcc8010006

[6] X. Zheng, "Suspicious transaction recognition based on KNN model for credit card fraud prevention," presented at the International Conference on Computational Finance and Business Analytics, Cham: Springer Nature Switzerland, 2024.

[7] A. Iqbal, R. Amin, F. S. Alsubaei, and A. Alzahrani, "Anomaly detection in multivariate time series data using deep ensemble models," *Plos One*, vol. 19, no. 6, p. e0303890, 2024. https://doi.org/10.1371/journal.pone.0303890

[8] N. S. Yadav, G. S. Reddy, A. S. Rao, and G. Sai, "Performance comparison of apache spark and sequential processing on machine learning classification algorithms," *Journal of Theoretical and Applied Information Technology*, vol. 102, no. 1, pp. 45–52, 2024.

[9] Z. Huang, Z. Xu, X. Wang, and Z. Xu, "The analysis of credit governance in the digital economy development under artificial neural networks," *Heliyon*, vol. 10, no. 20, p. e39286, 2024.

[10] J. Eriksson and E. Jakobsson, "Comparative analysis of machine learning libraries: Performance and Availability in Python, Scala, and Lua," Bachelor's Thesis, Linnaeus University, 2024.

[11] D. Han, N. Moniz, and N. V. Chawla, "AnyLoss: Transforming classification metrics into loss functions," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 992-1003.

[12] N. Elgiriyewithana, "Credit card fraud detection dataset 2023. Kaggle," Retrieved: https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detection-dataset-2023. [Accessed 2023.

[13] M. Azizi, U. Aickelin, H. A. Khorshidi, and M. Baghalzadeh Shishehgarkhaneh, "Energy valley optimizer: A novel metaheuristic algorithm for global and engineering optimization," *Scientific Reports*, vol. 13, no. 1, p. 226, 2023. https://doi.org/10.1038/s41598-022-27344-y

[14] G. Ke *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems*, vol. 30, p. 3147, 2017.

[15] Y. Ju, G. Sun, Q. Chen, M. Zhang, H. Zhu, and M. U. Rehman, "A model combining convolutional neural network and LightGBM algorithm for ultra-short-term wind power forecasting," *Ieee Access*, vol. 7, pp. 28309-28318, 2019. https://doi.org/10.1109/ACCESS.2019.2901920.

[16] D. Zhang and Y. Gong, "The comparison of LightGBM and XGBoost coupling factor analysis and prediagnosis of acute liver failure," *IEEE Access*, vol. 8, pp. 220990-221003, 2020. https://doi.org/10.1109/ACCESS.2020.3042848

[17] H. Nhat-Duc and T. Van-Duc, "Comparison of histogram-based gradient boosting classification machine, random Forest, and deep convolutional neural network for pavement raveling severity classification," *Automation in Construction*, vol. 148, p. 104767, 2023. https://doi.org/10.1016/j.autcon.2023.104767.

[18] S. Sperandei, "Understanding logistic regression analysis," *Biochemia Medica*, vol. 24, no. 1, pp. 12-18, 2014. https://doi.org/10.11613/BM.2014.003.

[19] M. Melinda, O. Maulisa, N. H. Nabila, Y. Yunidar, and I. K. A. Enriko, "Classification of EEG signal using independent component analysis and discrete wavelet transform based on linear discriminant analysis," *JOIV: International Journal on Informatics Visualization*, vol. 7, no. 3, pp. 830-838, 2023. https://doi.org/10.30630/joiv.7.3.1219.

[20] Y. A. Alsariera, V. E. Adeyemo, A. O. Balogun, and A. K. Alazzawi, "Ai meta-learners and extra-trees algorithm for the detection of phishing websites," *IEEE Access*, vol. 8, pp. 142532-142542, 2020. https://doi.org/10.1109/ACCESS.2020.3013699

[21] J. T. Hancock and T. M. Khoshgoftaar, "CatBoost for big data: An interdisciplinary review," *Journal of Big Data*, vol. 7, no. 1, p. 94, 2020. https://doi.org/10.1186/s40537-020-00369-8

[22] A. Ilham and A. Achmad, "Ensemble soft-voting model for classification optimization of medicinal plants leaves," presented at the 2023 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT), IEEE, 2023.

[23] H. A. Shehadeh, "Chernobyl disaster optimizer (CDO): A novel meta-heuristic method for global optimization," *Neural Computing and Applications*, vol. 35, no. 15, pp. 10733-10749, 2023. https://doi.org/10.1007/s00521-023-08261-1

[24] F. A. Hashim, R. R. Mostafa, A. G. Hussien, S. Mirjalili, and K. M. Sallam, "Fick's law algorithm: A physical law-based algorithm for numerical optimization," *Knowledge-Based Systems*, vol. 260, p. 110146, 2023. https://doi.org/10.1016/j.knosys.2022.110146

[25] H. Su *et al.*, "RIME: A physics-based optimization," *Neurocomputing*, vol. 532, pp. 183-214, 2023. https://doi.org/10.1016/j.neucom.2023.02.010

[26] M. Dehghani, Z. Montazeri, E. Trojovská, and P. Trojovský, "Coati optimization algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 259, p. 110011, 2023. https://doi.org/10.1016/j.knosys.2022.110011

[27] M. Dehghani and P. Trojovský, "Osprey optimization algorithm: A new bio-inspired metaheuristic algorithm for solving engineering optimization problems," *Frontiers in Mechanical Engineering*, vol. 8, p. 1126450, 2023. https://doi.org/10.3389/fmech.2022.1126450

[28] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers in Neurorobotics*, vol. 7, p. 21, 2013. https://doi.org/10.3389/fnbot.2013.00021

[29] Y. Ding, H. Zhu, R. Chen, and R. Li, "An efficient AdaBoost algorithm with the multiple thresholds classification," *Applied Sciences*, vol. 12, no. 12, p. 5872, 2022. https://doi.org/10.3390/app12125872

[30] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121-167, 1998. https://doi.org/10.1023/A:1009715923555

[31] A. Chandramouli, V. R. Hyma, P. S. Tanmayi, T. G. Santoshi, and B. Priyanka, "Diabetes prediction using hybrid bagging classifier," *Entertainment Computing*, vol. 47, p. 100593, 2023. https://doi.org/10.1016/j.entcom.2023.100593

[32] P. Cunningham and S. J. Delany, "k-Nearest neighbour classifiers: (with Python examples)," *arXiv preprint arXiv:2004.04523*, 2020. https://doi.org/10.1145/3459665.

[33] N. G. Ramadhan and F. D. Adhinata, "Sentiment analysis on vaccine COVID-19 using word count and Gaussian Naïve Bayes," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 26, no. 3, pp. 1765-1772, 2022. https://doi.org/10.11591/ijeecs.v26.i3.pp1765-1772.

[34] H. Blockeel, L. Devos, B. Frénay, G. Nanfack, and S. Nijssen, "Decision trees: From efficient prediction to responsible AI," *Frontiers in Artificial Intelligence*, vol. 6, p. 1124553, 2023. https://doi.org/10.3389/frai.2023.1124553.

[35] C. Minoccheri, O. Alge, J. Gryak, K. Najarian, and H. Derksen, "Quadratic multilinear discriminant analysis for tensorial data classification," *Algorithms*, vol. 16, no. 2, p. 104, 2023. https://doi.org/10.3390/a16020104