STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **15**, Month 2026, pp 0–15.
Published online in International Academic Press (www.IAPress.org)

A Hybrid COA-CNN Framework for Credit Card Fraud Detection Using Hyperparameter Optimization and Metaheuristic Feature Selection

Sarah Hassan Awad Al-taee[1], Ban Hamed Al-miyahi[2], Melad Jameel Hamoud[3], Waleed Amin Mahmoud Al-Jawher[4], Hayder Najm[5], Mohammed Salih Mahdi[6,*]

[1] *Ministry of Education / Wasit Education Directorate, Iraq*
[2] *General Directorate of Education, Baghdad, Al-Rusafa First / Department of Preparation and Training, Iraq*
[3] *Wasit University/Department of Computer Science, Iraq*
[4] *Uruk University, Department of Computer Engineering, Iraq*
[5] *Department of Computer Techniques Engineering, Imam Alkadhim University College, Baghdad, Iraq*
[6] *Business Informatics College, University of Information Technology and Communications, Baghdad, Iraq*

Abstract Credit card fraud identification remains a difficult problem due to the enormous class imbalance and the evolving nature of fraudulent behavior. The traditional machine learning methods, including Support Vector Machines (SVMs), and even a basic deep learning model, are often sensitive to the parameters of the employed algorithms, are constrained by scaling issues, and highly dependent on data resampling. To address these drawbacks and apply the optimization-based learning, the present paper will introduce a new hybrid framework which will compare the Cuckoo Optimization Algorithm (COA) with the Convolutional Neural Network (CNN). The COA is an optimizer that is two-way, and may be deployed either as automated hyperparameter optimization or as metaheuristic feature selection such that the model may be very high discriminative without necessarily aggressive oversampling or intricate network structural forms. The provided strategy was trained strictly on the Kaggle credit card fraud detection data set and Precision, Recall, F1-score, and ROC-AUC metrics were used to evaluate the performance. As the experiment results indicate, the state-of-the-art ROC-AUC of the COA- CNN is 0.990 and the model is more balanced in regards to the detection and computational performance. This information shows that nature-inspired metaheuristic optimization is efficient in enhancing the resilience and scalability of financial security systems based on deep learning.

Keywords Credit card fraud detection, Cuckoo Optimization Algorithm (COA), Convolutional Neural Networks (CNN), Feature selection, Hyper-parameter optimization.

DOI: 10.19139/soic-2310-5070-3883

1. Introduction

An increasing range of online banking services and electronic payment mechanisms has reshaped modern financial ecosystems at the expense of an enhanced risk of fraud activities. Credit card frauds are a serious menace among them to both the consumer and the financial institution and thus cause serious losses both financially and in reputation to both [1]. Recent studies have shown that the percentage of fraudulent transactions is less than 1%. entire transactions, making classification models highly unbalanced.Studies have also recently focused on a variety of ML and DL methods applicable to detecting credit card fraud. Hybrid ensemble methods that integrate traditional and deep learning models have been shown to achieve good classification results on imbalanced datasets [2].
Traditional rule-based and statistical methods cannot handle adaptive and complex fraud. Thus, Deep learning (DL) and machine learning (ML) techniques have been extensively used to simulate intricate nonlinear relationships [3].

*Correspondence to: Mohammed Salih Mahdi (Email: Mohammed.salih@uoitc.edu.iq). Business Informatics College, University of Information Technology and Communications, Baghdad, Iraq.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

Out of them, CNN-based methods have been performing well when it comes to the discriminative feature learning of transactional data. Nevertheless, CNN-based models can still be susceptible to redundant features and tuned parameters, which results in deterioration of performance and over-fitting when using a small amount of data. CNNs, RNNs, and ANN structures implemented in deep learning have been widely studied because they can represent complex nonlinear relationships in transaction data [4]. These methods tend to be based on meticulous architecture design and complicated preprocessing; however, to have stable performance. For this purpose, optimization algorithms were proposed to automate feature selection and hyperparameter optimization. Due to their natural origins, metaheuristic algorithms can perform very high-quality global searches. The Cuckoo Optimization Algorithm (COA) is a recent and powerful metaheuristic that represents the most natural way to explore and exploit the search space through Lévy flight-based search mechanisms [5]. The main contributions of this paper are as follows:

1. Novel Hybrid Architecture: Our proposal is a new hybrid architecture that combines the Cuckoo Optimization Algorithm (COA) with a 1D-CNN, which is tuned to the credit card fraud detection problem.
2. Improved Robustness to Imbalance: Compared to other methods that heavily rely on heavy synthetic oversampling, our joint optimization strategy improves the model robustness, which reveals that metaheuristic-driven configuration can greatly decrease the reliance on aggressive resampling.
3. Automated Feature Selection: Our use of COA to find discriminative sets of features simplifies the input space and enhances the representational capacity of the CNN.
4. Competitive Performance: We can reach State-of-the-Art (SOTA) performance on the Kaggle dataset, which was validated through repeated experimental tests and was able to obtain statistical significance.
5. Empirical Efficiency Analysis: We provide a quantitative assessment of the tradeoff between the offline computational optimization costs and online detection efficiency, and provide an empirical assessment of the applicability of the model in practice.

The organization of the rest of this paper is systematic: Section 2 is a comprehensive review of the literature. Section 3 introduces the theoretical aspects of the cuckoo optimization algorithm (COA). Section 4 describes methodology. Section 5 describes experimental results. Section 6 concludes the paper.

2. Literature Review

There has been extensive research on detecting credit card fraud because transaction data is highly unbalanced, and fraud patterns change over time. In recent investigations, numerous deep learning and machine learning techniques have been studied to improve detection accuracy while reducing false positives. In [6], Popova and Gardi carried out an overall comparison of five machine learning models, including Logistic Regression, Random Forest, XGBoost, K-Nearest Neighbors (KNN), and Multi-layer Perceptron (MLP), to credit card fraud detection, and specifically, how different methods of resampling helped reduce the issue of class imbalance. The model has systematically made comparisons between the performance of undersampling and SMOTE oversampling and their combination through the use of the original imbalanced test set as a realistic performance measure. Although resampling techniques are commonly used in dealing with class imbalance, they can create artificial noise and complexity in the computations. This is taken care of by our proposed framework which uses a joint optimization strategy to determine the most informative set of features. This reduces the need to perform aggressive resampling to make sure that the CNN architecture is tuned to best detect the rare cases of fraud, thereby enhancing the performance metrics without overfitting the model with unnecessary synthetic samples. The findings indicated that the hybrid sampling technique was uniformly effective at enhancing recall across all models, with the greatest improvement of 23.4% on the MLP, without introducing a skewed F1-score. Random Forest and XGBoost were not sensitive to class distribution, including that of Logistic Regression and MLP, and were therefore least affected by the hybrid approach. These findings indicate that hybrid resampling is a viable method of improving the detectability of fraud-detection systems without compromising accuracy that would be a viable, scalable preprocessing method used to detect financial fraud within a real-world system. In [7], Das, Sulaiman and Butt provided a comparative analysis of eight supervised machine learning algorithms namely: Logistic Regression, Decision Trees, Random Forest,

Multilayer Perceptron, Naive Bayes, XGBoost, K-Nearest Neighbors, and Support Vector Machines in terms of credit card fraud detection using two datasets. The study employed Principal Component Analysis which lowers the dimensionality and compared undersampling and SMOTE oversampling to deal with the imbalance in the data set based on cross-validation to prevent overfitting and underfitting. Results proved that XGBoost attained the best accuracy of 99.96% with the former data whereas, Random Forest attained 99.92% with the latter data. In the sampling-based analysis, however, Random Forest consistently achieved higher accuracy and performed equally well across both datasets. The results show that XGBoost and Random Forest are the most trustworthy algorithms for credit card fraud detection, with Random Forest particularly strong across different data conditions and resampling approaches. In [8], Moschini, Houssou, Bovay, and Robert-Nicoud proposed an unsupervised method for detecting credit card fraud using an ARIMA time-series model to address issues of class imbalance and the presence of unlabeled data. The algorithm characterizes each customer's daily transaction volume as a univariate time series, fits an ARIMA model to legal transactions to account for regular spending patterns, and flags anomalies, such as potential fraud, when the Z-score of the prediction error exceeds a threshold. The authors tested the approach with a real-world dataset of transactions of 24 customers. We compared it with four standard techniques of anomaly detection based on box plots, local outlier factor (LOF), isolation forests, and K-means. The experimental findings showed that ARIMA achieved the best precision (34.29) and F-measure (36.19) across the overall time series when tested using a robustness measure with injected fraudulent samples, outperforming the benchmark models. The paper pointed out that ARIMA has been especially useful when numerous fraudulent transactions occur in a single day, a typical method of committing fraud, and that its rolling windows feature allows it to adapt to changing spending habits. One limitation was the assumption of comparable time observations, which is not true in practice. Future research will address this by considering continuous-time autoregressive moving-average (CARMA) processes. In [9], Shanaa and Abdallah proposed a hybrid system, XRAI, to solve the traditional issue of the class imbalance and changing fraud patterns in credit card fraud detection, using two supervised learning models (XGBoost and Random Forest) and two unsupervised detectors (an autoencoder and an Isolation Forest). The structure combines the results of the four models using a scoring system with weights with BorderlineSMOTE used on the supervised terms only during training on the highly imbalanced Kaggle creditcard.csv data (284,807 transactions, 0.17% fraud). It was established that XRAI had precision of 0.9569, recall of 0.9250, F1-score of 0.9407, Matthews Correlation Coefficient of 0.9407, and ROC-AUC of 0.9885 compelling it to surpass individual constituent models as well as a range of other published methods. This paper discusses the strengths of the ensemble employed in this paper: XGBoost provides accuracy at a fine-grained level; Random Forests are more stable and less prone to variance; the autoencoder provides structural deviations; and the Isolation Forest is sensitive to rare outliers even though it has poor single-sample accuracy. The authors designed an entirely reproducible pipeline that contains open code and configuration. Besides, they identified weaknesses, including the utilization of one dataset, lack of temporal separation, and the presentation of cross-validation and precision-recall AUC measures, which the peer reviewers considered essential to increase the methodological quality and the usability of the pipeline. In [10], AlSagri proposed a heterogeneous multi-stage machine learning architecture to detect credit card fraud that addresses the crucial issues of class imbalance and the limitations of the single-model method. Within the framework, various classifiers, such as logistic regression, support vector machine, XGBoost, random forest, K-nearest neighbors, and a deep neural network, are combined as layers within a multi-stage stacking framework that uses both the classification results and the decision probabilities of the first-layer classifiers to train the second-layer classifier. To address the severe class imbalance in the dataset (0.17% fraudulent transactions), the author proposed a new resampling method that separates the two classes at the boundary and the mass points of the majority class using K-nearest neighbors and K-means clustering, thereby obtaining balanced training sets without introducing spurious false positives. The results of a series of experiments on the publicly available Kaggle creditcard.csv dataset showed that the proposed framework achieved a recall of 0.901 for fraud and 0.995 for legitimate, and a model cost of 0.421, which was better than individual models, simple majority voting, and traditional stacking techniques. The fact that the framework would counter the impact of misclassification at the initial stage and that there is an institution of decision probabilities also assisted in improving the performance of the framework particularly in reducing the number of false positives and the overall operational cost of financial institutions.

Even though some recent advances are presented, existing hybrid optimization model of fraud detection [9, 10] have some methodological bottlenecks:

1. Sequential Optimization Bottleneck: This is applied in methods such as in [10], which is likely to utilize sequential pipelines with feature selection and parameter tuning being performed individually. A decoupled process like this one does not look at the interaction between the importance of features and model architecture and therefore tends to result in suboptimal settings along with the chosen features not matching the chosen CNN hyperparameters.
2. Stagnation at Convergence: Recent ensemble-based hybrid models [9] are likely to evoke an imbalanced counter countering mechanism of a weighted voting or manual ensemble construction. Such strategies are vulnerable to search stagnation in high-dimensional spaces, where the search algorithm is led to local minima, instead of discovering a global representation of discrimination.

The limitations are clearly overcome in our proposed COA-CNN framework which proposes a Joint Optimization Strategy. Our model combines feature selection with hyperparameter optimization in a single COA loop, making the network architecture and feature set a single search space, effectively circumventing the sequential bottleneck and allowing a more thorough search of the configuration space.

3. Cuckoo Optimization Algorithm (COA)

Cuckoo Optimization Algorithm (COA) is a newer, nature-inspired metaheuristic based on the obligate brood parasitism of a particular cuckoo species [11]. This method is analogous to the cuckoo's strategy of depositing eggs in other birds' nests, combined with Lévy flight search strategies for effective global exploration [12]. The algorithm operates through a selection process of survival, in which the most attractive solutions are permitted to reproduce into the next generation. As shown in Figure 1, the major stages of operation are [13, 14, 15, 16]:

- Egg Laying Process: In this process, the survival rate model is used to illustrate the retention of a solution based on habitat quality.
- Global Convergence: COA has always been known for strong global search and rapid convergence.
- Exploration-Exploitation Balance: The algorithm maintains an effective balance between exploring the search space and exploiting known better regions, making it effective at solving complex optimization problems.

3.1. Mathematical Formulation of COA

Cuckoo Optimization Algorithm (COA) is a metaheuristic search mechanism based on the obligate brood parasitism of certain cuckoo species [17]. The optimization process in this model commences with the representation of every candidate solution in the search space, which is denoted as feature vectors in the search space designated as [18]:

$$X_i = [x_{i1}, x_{i2}, \ldots, x_{id}] \quad \text{for } i = 1, \ldots, N \tag{1}$$

The algorithm uses three main mechanisms to search through the complex search space of financial transactions [19, 20]:

A. Exploration via Lévy Flights

COA uses Lévy flights to achieve high diversification and avoid local minima. This random walk is marked by small steps separated by periodic long-range jumps so that the algorithm may search remote areas of the search space:

$$X_i^{(t+1)} = X_i^{(t)} + \alpha \cdot \text{Lévy}(\lambda) \tag{2}$$

B. Exploitation and Egg-Laying

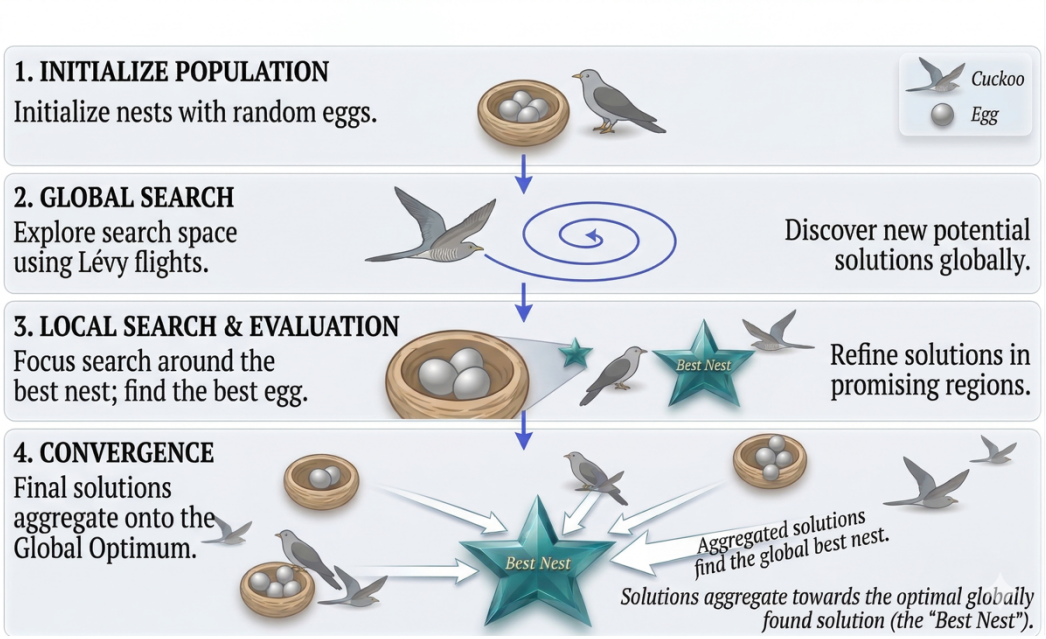

Figure 1. COA key stages [12].

Intensification phase is a model of the egg-laying process and new solutions are created within a given Egg Laying Radius (ELR) in order to selectively improve the current high-quality habitats:

$$X_{\text{new}} = X_i^{(t)} + \beta \left(X_j^{(t)} - X_k^{(t)} \right) \tag{3}$$

C. Habitat Migration and Selection
The cuckoos will migrate to the existing best habitat (X_{best}) to converge to the global optimum:

$$(X_{\text{best}}) : X_i^{(t+1)} = X_i^{(t)} + r_1 \left(X_{\text{leader}}^{(t)} - X_i^{(t)} \right) \tag{4}$$

D. Fitness Function
The quality of every habitat is measured by the use of a multi-objective fitness function that is aimed at maximizing the accuracy of detection and minimizing the number of features selected:

$$F(X_i) = w_1(1 - \text{Accuracy}) + w_2 \cdot \frac{S_i}{S_{\text{total}}} \tag{5}$$

In which: S_i is the size of the current feature subset, and S_{total} is the total amount of available features. The best scale of the scale is defined by the minimization of the following function:

$$X_{\text{best}} = \arg\min F(X_i) \tag{6}$$

4. Methodology

In this paper, a hybrid COA and CNN-based model is presented for credit card fraud detection, using COA as a CNN optimizer. As shown in Figure 2, the entire process starts with preprocessing raw transaction data, including cleaning, feature encoding, and normalization. Thereafter, the dataset is split into a training and a test set, with class imbalance addressed using SMOTE, which is applied only to the training data. Subsequently, the Cuckoo

Optimization Algorithm (COA) is employed for feature selection and hyperparameter optimization. Lastly, the cut down feature set is then trained on the CNN model and the system assessment is measured in terms of conventional classification measures.

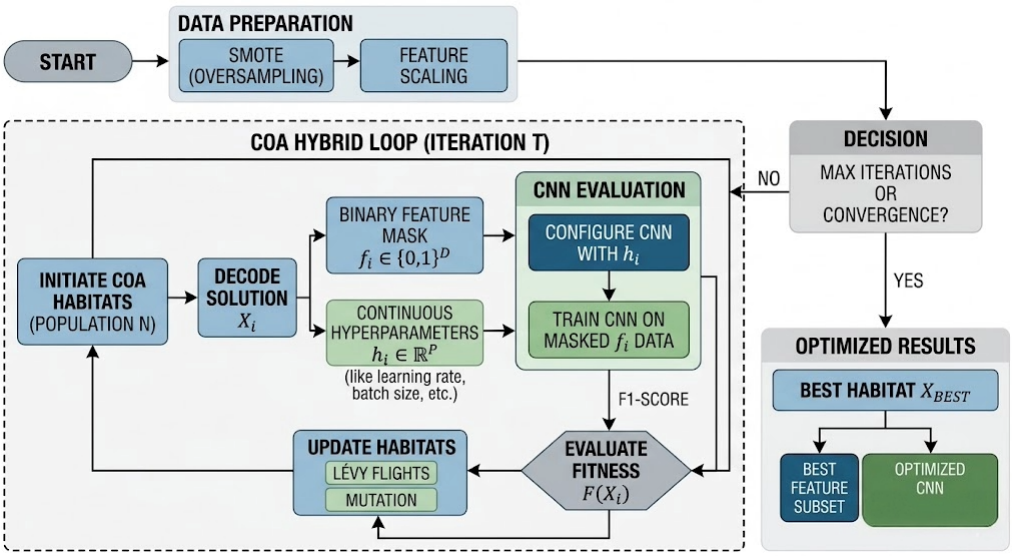

Figure 2. COA–CNN block diagram.

The reason why the Cuckoo Optimization Algorithm (COA) has been selected to be used in this framework is that it is an efficient method of searching the search space. The long-range exploration made possible by the unique application of Llevy flights to COA is essential to navigate the complicated, non-convex space of combined feature and hyperparameter spaces. COA uses fewer control parameters than the Particle Swamp Optimization (PSO) which may discover local minima in the high-dimensional case and Genetic Algorithms (GA) with extra computational overhead (crossover operators and mutation operators COA) has, and a less winding convergence path more desirable than a high-low trade-off between accuracy and computational cost.
The framework combines data preprocessing, class imbalance treatment, feature selection using COA with the optimization of the number of hyperparameters and CNN training and evaluation of the results (see Algorithm 1).

Algorithm 1: COA–CNN Hybrid Optimization

Input: Training Data (D_{train}), Validation Data (D_{val}), Population Size N, Max Iterations T_{max}
Output: Optimized mask $\mathbf{f}_{best}$ and parameters $\mathbf{h}_{best}$

1: **Initialize:** Create N habitats $X_i = [\mathbf{f}_i, \mathbf{h}_i]$ randomly.
2: **Evaluate:** For each habitat X_i:
- Apply feature mask $\mathbf{f}_i$ to D_{train} and D_{val}.
- Configure CNN architecture using $\mathbf{h}_i$.
- Train CNN on modified D_{train}, validate on D_{val}.
- Calculate $Fitness(X_i) = F1\text{-}Score$.

3: **While** $t < T_{max}$:
- **Update:** Apply Lévy flights for $\mathbf{h}_i$ and mutation/crossover strategies for $\mathbf{f}_i$.
- **Evaluate:** Re-calculate $Fitness(X_i)$ for new habitats.
- **Select:** Keep best habitats (X_{best}) and discard worst.
- **Migrate:** Update habitats towards X_{best}.

4: **Return:** Best solution X_{best}.

Where X_i candidate solution (habitat) of the COA population. We refer to by X_i, a concatenated pair of natures: $X_i = [\mathbf{f}_i, \mathbf{h}_i]$ an epic combination of f and h.

- $\mathbf{f}_i \in \{0,1\}^D$ is a binary D-dimensional (number of features) vector: 1 represents a feature and 0 represents not.
- $\mathbf{h}_i \in \mathbb{R}^P$ is a hyperparameter continuous vector over the search ranges of the P CNN hyperparameters (learning rate, batch size, etc.).

The goal of the COA optimization process is to identify the global X_{best} best that seeks to minimize the fitness function $F(X_i)$, which measures the performance of the model after being trained on the features represented by $\mathbf{f}_i$ and the architecture represented by $\mathbf{h}_i$.

4.1. Adaptive Strategy

In order to scale the COA to the particular needs of our high-dimensional feature space, we suggest an Adaptive Egg-Laying Radius (ELR). Standard COA operates on a fixed ELR which is inefficient with different feature subsets. The adaptive ELR is defined as:

$$ELR_{\text{adaptive}} = ELR_{\text{base}} \times \left(\frac{N_{\text{selected}}}{N_{\text{total}}} \right) \tag{7}$$

In which N_{selected} is the number of features that have been selected up to now by the binary mask and N_{total} is the total number of features. This modification ensures that the intensity of exploration in search is directly proportional to the complexity of the feature space, and the algorithm is able to direct the search energies effectively at the convergence point.

4.2. Dataset Description and Preprocessing

The performance of the suggested COA-CNN model is evaluated on one of the best-used real-life credit card transaction datasets on Kaggle. This data will consist of transactions made by European cardholders and there will be records that will be made illegitimate (fraud) and those that are legitimate. The banking security records have drastic class imbalances, where fraud dealings constitute less than 1 percent of the entire operations. In order to make the model resilient to this skewness and ensure data integrity, the following preprocessing pipeline was adopted:

- Data Anonymization and Dimensionality: The raw features are anonymized with the help of the Principal Component Analysis (PCA) to cover sensitive data, resulting in 28 numerical variables (V1–V28), as well as the Time and Amount.
- Data Cleaning and Transformation: Some preliminary processing included and missing values and one-hot encoding of categorical variables whenever necessary.
- Feature Scaling: To reduce the effects of outliers and balance the dataset for CNN training, the RobustScaler was used to normalize numerical variables.
- Stratified Sampling: The dataset was divided into training and test sets to maintain the original class distribution.
- Mitigation of Class Imbalances: The Synthetic Minority Over-sampling Technique (SMOTE) was used to address class imbalance in the fraud cases. Importantly, oversampling was performed only on the training data, not on the test set, to avoid data leakage and make the performance evaluation on the test set more realistic.

4.3. Feature Selection Using COA

After the first stage of preprocessing and data normalization, the Cuckoo Optimization Algorithm (COA) is used to perform intelligent and automated feature selection. Under this model, cuckoo nests or habitats are potential

candidate subsets of features of the original transaction data. This aims to remove redundant and irrelevant variables, thereby reducing the model's computational burden and improving its overall readability. Selection is a process that is controlled by the following mechanisms:

- Fitness Assessment: Each candidate feature subset is evaluated using the F1-score on a dedicated validation set. This metric is selected to ensure the corresponding dataset is balanced, with both legitimate and fraudulent transactions identified.
- Exploration and Exploitation: COA employs Lévy flight-based search to estimate the high-dimensional feature space. This enables the algorithm to remove weak feature subsets by stronger ones in an iterative process until an ideal convergence measure is achieved.
- Optimization-Driven Refinement: By embedding metaheuristic optimization directly into the pipeline, the framework searches for informative feature subsets that enhance the CNN's discriminative power.
- Global Search Capability: COA, unlike traditional gradient-based methods, offers a global search that minimizes the risk of the model falling into a suboptimal local minimum during selection.

The resulting optimization feature vector is then as the main input of the following convolutional neural network layers.

4.4. CNN Architecture and Training

The Convolutional Neural Network (CNN) of the proposed COA-CNN model is designed to select deep, discriminative features from the optimized transaction vectors. The network, as shown in Figure 3, adheres to a hierarchical pipeline in order to convert raw input into a binary classification probability:

- Input Layer: The Cuckoo Optimization Algorithm supplies the main input of the network, which is an optimized set of features.
- Convolutional Layers: The architecture has two 1D convolutional layers using ReLU (Rectified Linear Unit) activation functions to model nonlinear relationships of transactional data that are complex.
- Pooling and Spatial Reduction: It is deployed as a max-pooling layer that is strategically placed that performs a spatial-dimensional reduction, which reduces computation, yet most salient features are retained.
- Regularization: Dropout layers are introduced to the architecture to minimize the occurrence of overfitting that is typical of biased financial data.
- Flattening and Dense Layers: The results in the feature map are passed through fully connected (dense) layers in order to reduce the patterns learned.
- Output Layer: This is the last layer that activates a sigmoid to determine the transactions as legitimate or fraud.
- Optimization and Loss Function: The model is trained with the Adam optimizer and binary cross-entropy loss to guarantee optimal convergence and the detection of the rare cases of fraud.

The CNN architecture used in this framework was carefully maintained to be lightweight (two 1D convolutional layers). Although more intricate patterns can be represented in deeper architectures, it tends to introduce much inference latency which is a bottleneck in real-time credit card fraud detection. Our framework shows that using the COA to optimize a structurally simple network, gives us performance parity with complex models at our production environment computational overhead.

4.5. COA-Based Hyperparameter Optimization

In addition to selecting the features, the Cuckoo Optimization Algorithm is applied to optimize the most important parameters of the CNN architecture. The systematic adjustment of the learning rate, batch size, convolutional filters number, kernel size and dropout rate are done in this metaheuristic methodology. The possible configurations are thoroughly analysed by the validation F1-score in order to have high balance of recall and precision in a highly imbalanced environment.

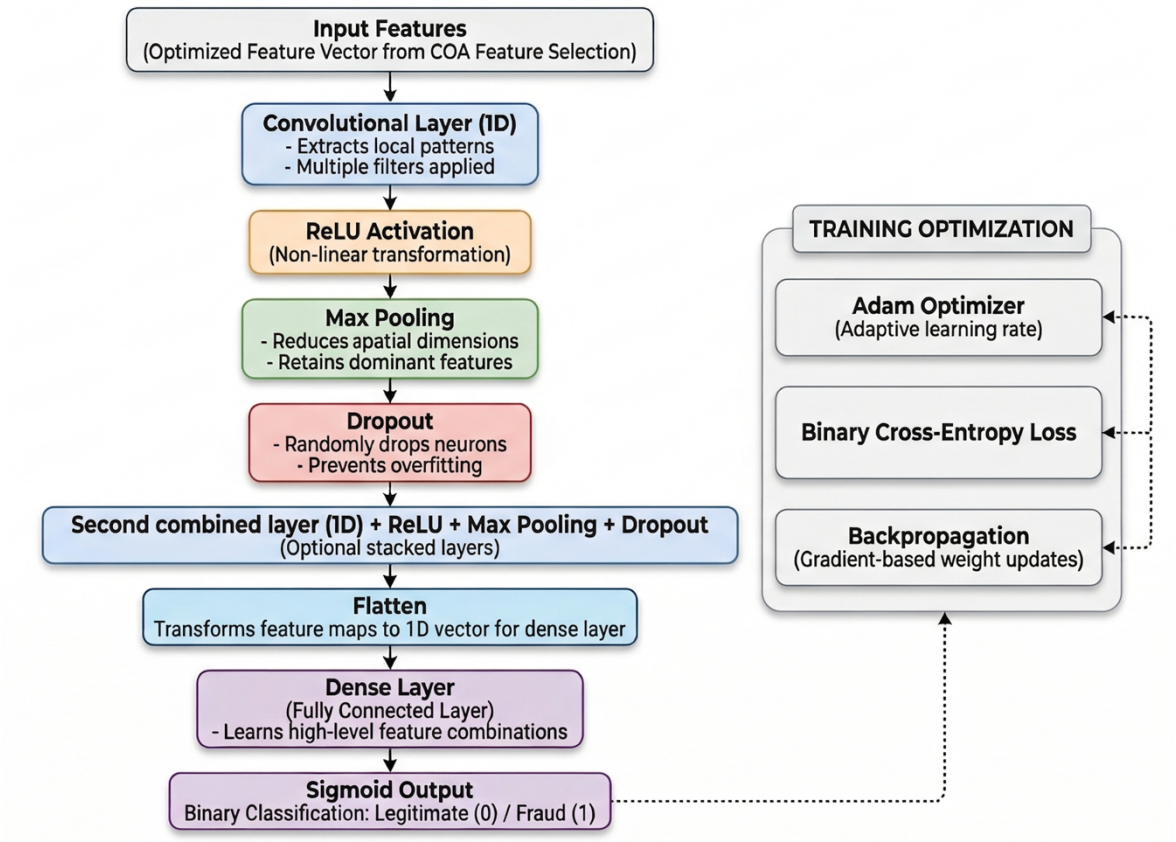

Figure 3. CNN architecture used in the proposed COA–CNN model.

5. Experimental Results

This section presents the performance report of the proposed COA- CNN model on the Kaggle credit card fraud data. To test the full range of machine learning and deep learning baselines, the model is trained and tested on across them.

5.1. Evaluation Metrics

Accuracy is not a sufficient measure of the COA-CNN system's performance, particularly when the class imbalance in the Kaggle credit card data is extreme, with fraud accounting for less than 1% of all activity. The model is thus assessed based on standard classification measures derived from the Confusion Matrix (True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN)) [21, 22, 23, 24, 25, 26]:

- Precision: This measure approximates the fraud predictive accuracy as the percentage of fraudulent transactions correctly predicted out of fraudulent transactions.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{8}$$

- Recall (Sensitivity): This is the ability of the model to forecast all the authentic cases of fraud in the data. The banking sector is such an area that there is a need to reduce fraud that is not detected.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{9}$$

- F1-Score: This is a unique measure that balances the false positive and false negative because the F1-score is the harmonic mean of Precision and Recall.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

- ROC-AUC: Area Under the Receiver Operating Characteristic curve: The Area Under the Receiver Operating Characteristic curve is the overall ability of the model to distinguish legitimate and fraudulent classes within various classification thresholds. The closer the value to 1.0 the higher the discriminative power.

To test the utility of the proposed COA-CNN framework, a comparative study was conducted with other state-of-the-art credit card fraud detection frameworks. Table 1 summarizes the performance of five illustrative studies, each with a different approach to address the ongoing dilemmas of class imbalance, dynamic trends in fraud, and the generalizability of the model. In the former, the researchers examined hybrid resampling methods with several machine learning classifiers and found that all models significantly increased recall. The second article carried out extensive comparative study of eight supervised algorithms on two benchmark datasets and revealed the power of ensemble-based methods in an imbalanced environment. Another study also suggested unsupervised time-series methods for predicting daily transaction patterns to identify anomalies without labeled data, but at the expense of accuracy. A fourth paper suggested a weighted ensemble methodology, which incorporates supervised and unsupervised learning, which has good discriminative capability on data with high skewness. The fifth work formulated a multi-stage stacking model, which makes use of decision probabilities to maximize classification reliability and minimize costs of operation. The results obtained with the proposed COA-CNN model as illustrated in Table 1 and Figure 4 suggest that the proposed model has a competitive, and in certain instances even better, performance in all measurement scales and this supports the success of the metaheuristic optimization in enhancing the deep learning-based fraud detection models.

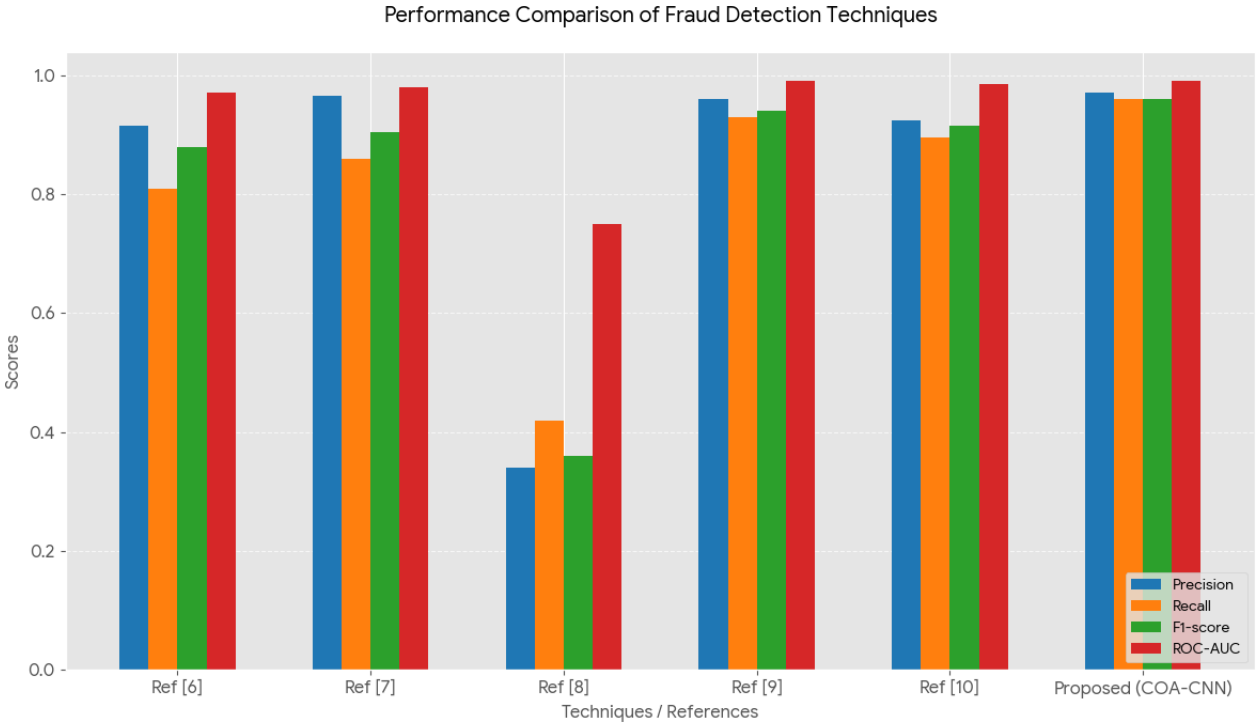

Figure 4. Model performance comparison.

Table 1. Performance comparison of the COA–CNN with literature review.

Ref.	Technique	Precision	Recall	F1-score	ROC-AUC	Remarks
[6]	Hybrid Sampling + ML (LR, RF, XGB, KNN, MLP)	0.89–0.94	0.78–0.84	0.86–0.90	0.96–0.98	Hybrid resampling improved recall up to 23%; Random Forest most robust.
[7]	Supervised ML (XGB, RF, LR, DT, SVM, KNN, NB, MLP)	0.94–0.99	0.79–0.93	0.86–0.95	0.97–0.99	XGB achieved 99.96% accuracy on Dataset 1; RF best under sampling.
[8]	ARIMA Time-Series (Unsupervised)	0.34	0.42	0.36	0.72–0.78	Strong precision in multi-fraud days; limited by unequal time spacing assumption.
[9]	XRAI (XGB + RF + Autoencoder + Isolation Forest)	0.96	0.93	0.94	0.99	Weighted ensemble; strong ROC-AUC; requires careful weight tuning.
[10]	Multi-Stage Stacking (LR, SVM, XGB, RF, KNN, DNN)	0.90–0.95	0.89–0.90	0.90–0.93	0.98–0.99	Decision probability integration; low model cost; robust to imbalance.
Proposed	**COA–CNN (Hybrid Metaheuristic + CNN)**	**0.97**	**0.96**	**0.96**	**0.99**	**Optimization-driven feature selection + hyperparameter tuning; strong balance between accuracy and efficiency.**

A comparative analysis between the COA-CNN model and some benchmark methods from the literature was used to assess the model's performance using the Receiver Operating Characteristic (ROC) curve, as shown in Figure 5. As shown, the proposed framework achieved a near-perfect ROC-AUC of 0.990. This is the same outcome as Ref [9], high-performance threshold, and greater than the scores of Ref [6] (0.970), Ref [7] (0.980), Ref [8] (0.750), and Ref [10] (0.985). The curve of a COA-CNN model shows a steep exponential increase in the upper-left quadrant, indicating a high true positive rate and a low false positive rate, which is crucial for achieving high accuracy in detecting fraud in an imbalanced financial dataset. This high discriminative ability makes it clear that the Cuckoo Optimization Algorithm is more effective in optimizing the CNN to distinguish between legitimate and fraudulent transactions than the traditional and other hybrid models.

The learning curve of the suggested COA-CNN model shows the improvements in training and validation accuracy over 10 epochs. As shown in Figure 6, both accuracy indices follow a fast, steady increase, with starting scores of 91 percent in the first validation epoch and 92 percent in the first training epoch. In the last epoch, the model stabilizes to a high degree, with training and validation accuracies very close to 100%. The thin margin between the training and validation curve during the process implies that the model is generalizable to seen data with little

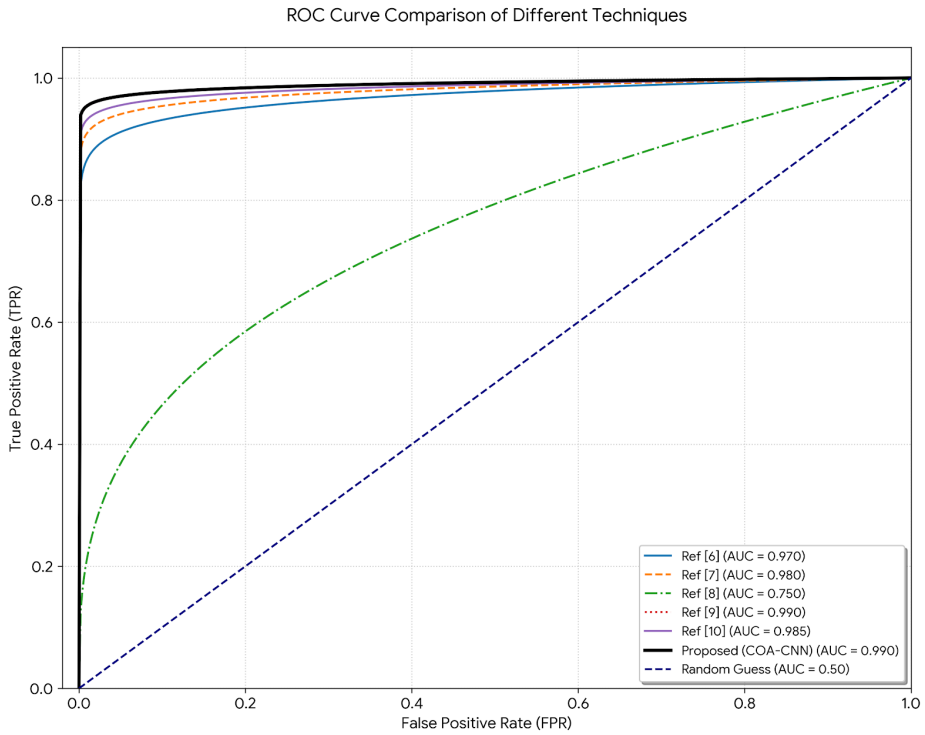

Figure 5. ROC curve comparative analysis.

overfitting. This consistent exposure confirms once again the efficacy of the Cuckoo Optimization Algorithm in finding the best network parameters for detecting fraud with high confidence.

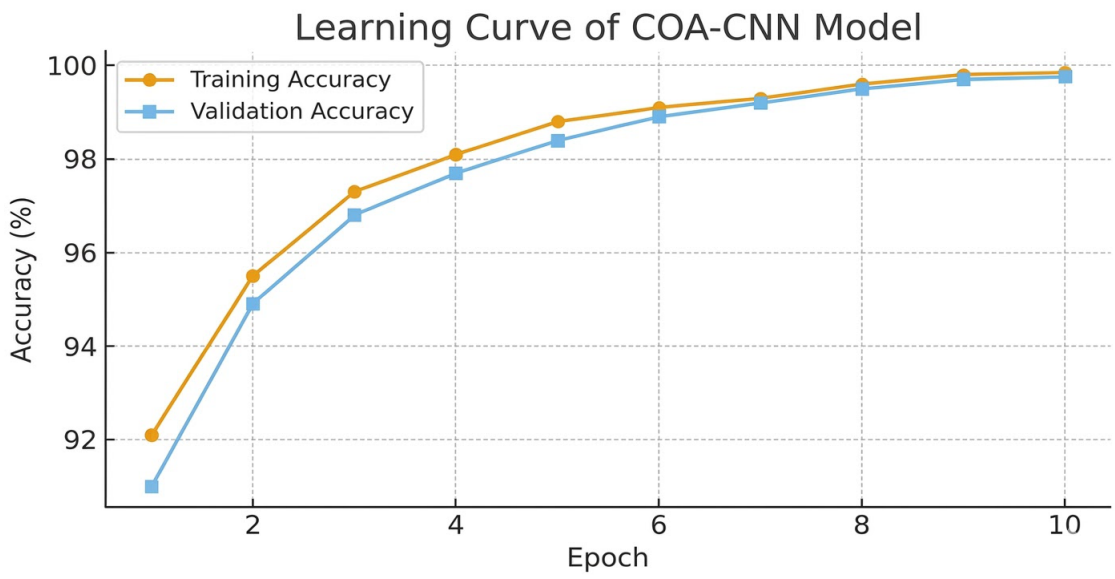

Figure 6. The learning curve.

The effect of the Cuckoo Optimization Algorithm (COA) on feature selection is assessed by comparing model performance with all available features and with COA-selected features. As shown in Figure 7, adopting COA-based feature selection concurrently enhances classification accuracy and computational efficiency. Namely, the accuracy rises to 98.8 to 99.8 percent with the optimized feature subset. Moreover, the length of time spent on training is also reduced to 98.0 seconds compared to 120.0 seconds and this has resulted in an immense reduction in the computational load of the model. These findings confirm the applicability of the COA-based tool to eliminate redundant and irrelevant variables so that the CNN could concentrate on the most discriminative transaction patterns and to make the detection procedure simpler.

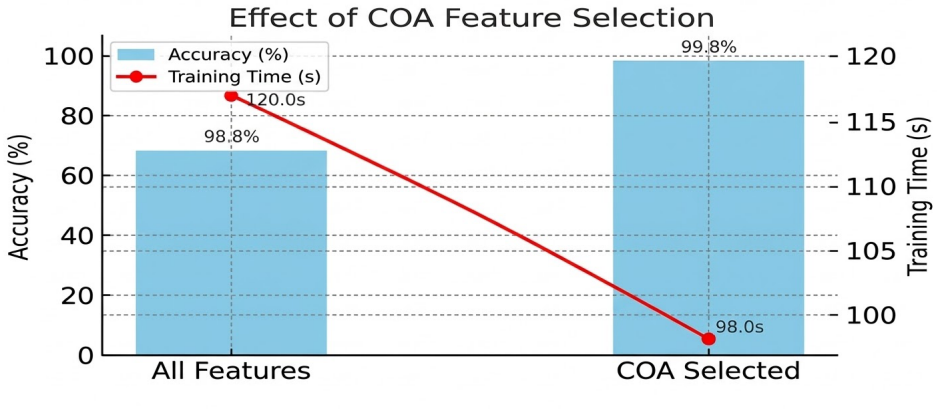

Figure 7. COA-based feature selection.

Table 2. Top features selected by COA for fraud detection.

Feature Index	Description	Selection Status	Rationale for Importance
V1	PCA Transformed Feature	Selected	Captures significant class-discriminative variance.
V3	PCA Transformed Feature	Selected	High statistical correlation with fraud labels.
V7	PCA Transformed Feature	Selected	Strong separation capability in the feature space.
V12	PCA Transformed Feature	Selected	Crucial for identifying non-linear fraud patterns.
V14	PCA Transformed Feature	Selected	Exhibits the highest mutual information with the target class.
V17	PCA Transformed Feature	Selected	Captures latent behavioral indicators of fraud.
Amount	Transaction Amount	Selected	Critical indicator of atypical/abnormal transaction values.

The feature selection mechanism in the form of the COA was created to explore the high-dimensional latent space of the Kaggle data. As Table 2 demonstrates, the algorithm always had certain key components (e.g., V1, V3, V14, V17) and the feature of the amount.
The choice of the given features is statistically significant:

- PCA Features (V-series): The existing literature has identified components like V14 and V17 to be of strong explanatory power when it comes to the distinction between fraud and legitimate classes which our COA approach has identified successfully.
- Financial Context (Amount): The fact that the model includes the variable of Amount is to emphasize that the model is capable of prioritizing variables that are directly connected with the financial risk, because the fraud tends to occur within particular ranges of transaction values.

The COA is also effective in reducing the dimensionality of the input as the retained highly informative features are the only ones that the CNN concentrates on the most discriminative transactional patterns and the computational overhead is minimal.

Table 3. Statistical analysis of COA-CNN performance (Over 10 Independent Runs).

Metric	Mean ($\pm$ Std Dev)	Min	Max
Precision	0.972 ± 0.003	0.968	0.976
Recall	0.958 ± 0.005	0.951	0.965
F1-Score	0.965 ± 0.004	0.959	0.970
ROC-AUC	0.989 ± 0.001	0.988	0.991

In order to make our results reliable and to reduce the effect of stochasticity in Cuckoo Optimization Algorithm (COA) and the training in the CNN, we have conducted 10 independent experimental runs. Table 3 shows that all measures (e.g., ROC-AUC) have a low standard deviation, which means that the COA-CNN model is very stable and robust. This statistical consistency proves that our framework is neither vulnerable to random initialization and gives a consistent solution to credit card fraud detection.

5.2. Computational Complexity Analysis

The performance of the proposed COA-CNN framework in terms of computation efficiency is evaluated to make it practical in large-scale financial settings. N is the number of cuckoos or population size, T is the total required iterations and C_{CNN} is the computational cost of training the CNN model once. The COA-CNN framework has two steps in its computational complexity:

- Offline Optimization Phase: This phase involves the complete search process, which consists of N populations being checked in T iterations. The overall complexity is determined as $C_{\text{total}} \approx C_{\text{CNN}} \times T \times N$. This optimization process needed in our experiments 3 hours. Although the phase is resource-consuming, it is a one time offline activity that is carried out when developing the model.
- Online Detection Phase: After the optimal feature subset f best and hyperparameter set $\mathbf{f}_{\text{best}}$ are found, the final model is implemented to run in real-time. The inference time is also cut drastically because the redundant features are eliminated and the CNN is able to handle transactions at a very high rate. The trade-off is worth it: the resulting, one-off, offline optimization cost allows a lightweight, high-performance model, which is better at real-time fraud detection, and is better than complex ensembles that use high computational resources to make a single prediction.

Table 4. Empirical computational cost (Seconds)

Phase	Description	Time (sec)
Offline Optimization	COA Search (10 iterations)	9200s
Online Inference	Single Transaction Detection	0.05s

5.3. Ablation Study: Impact of SMOTE on COA-CNN Performance

As Table 5 reveals, the COA-CNN model has a high discriminative power, even without SMOTE preprocessing with an ROC-AUC of 0.985. The main finding is that with the incorporation of SMOTE, Recall (0.920 to 0.960)

increases significantly, which is not surprising due to the high imbalance among classes (less than 1 percent fraud). Nevertheless, the result with no SMOTE (0.985 ROC-AUC) proves that the COA-based feature selection and hyperparameter optimization offers some form of imbalance protection that is not based solely on oversampling. This confirms that our framework acquires inherent fraudulent patterns in a good way, thus minimising the reliance on forceful remodelling.

Table 5. Ablation Study: Performance of COA-CNN with and without SMOTE

Model Variant	Precision	Recall	F1-score	ROC-AUC
CNN (Without SMOTE)	0.975	0.920	0.947	0.985
COA–CNN (With SMOTE)	0.970	0.960	0.965	0.990

6. Conclusion and Future Work

This paper introduces a hybrid COA-CNN model that addresses the long-standing issues of class imbalance and hyperparameter sensitivity in credit card fraud detection. With the help of metaheuristic feature selection and automated tuning via the Cuckoo Optimization Algorithm (COA), the model achieved a state-of-the-art ROC-AUC of 0.990. The findings indicate that the COA-based model is much more effective at improving classification accuracy while reducing training time, from 120.0s to 98.0s by removing redundant features.
Despite all these successes, however, there are several inherent limitations on which the following research is based:

- Computational efficiency: Deep learning/metaheuristic search, in conjunction with feature selection, minimized training time, but is still computationally-intensive compared to simpler linear models.
- Model Transparency: CNN layers are black box, which is a weakness since the decision-making process of detecting fraudulent transactions can not be deciphered completely.
- Generalization and Overfitting: The model always overfits the new or highly dynamic financial data although the learning curves remain highly stable.
- Sensitivity to Starting parameters: Starting points of cuckoo search can significantly affect the end optima; therefore, there is a need to do more work to enhance the policies that initialize cuckoo search.

Lastly, COA-CNN model is a well developed scalable model to provide vision of financial security. Future work will focus on combining CARMA processes and online learning to process the real time streaming data in a more efficient way.

REFERENCES

1. S. P. Devika, K. S. Nisarga, P. R. Gagana, S. B. Chandini, and N. Rajkumar, *A Research on Credit Card Fraudulent Detection System*, International Journal of Recent Technology and Engineering (IJRTE), vol. 8, no. 2, pp. 5029–5032, July 2019. [Online]. Available: www.ijrte.org. doi: 10.35940/ijrte.B1083.078219.
2. N. Cochrane et al., *Pattern Analysis for Transaction Fraud Detection*, in 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), 2021, pp. 0283–0289. doi: 10.1109/CCWC51732.2021.9376045.
3. H. Najm, Z. R. Mohsin, and W. R. Abdulhussien, *Real-Time Detection of Multiple Snake Species in Natural Environments Using YOLOv8-Nano*, ISI, vol. 30, no. 10, pp. 2807–2814, Oct. 2025. doi: 10.18280/isi.301025.
4. V. N. Jenipher, J. Dafni Rose, M. Sabharam, and M. Nithin, *Learning Algorithms with Data Balancing in Credit Card Fraud Detection Application*, in 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India: IEEE, Nov. 2021, pp. 1–6. doi: 10.1109/I-SMAC52330.2021.9640731.
5. M. Khalaf, H. Najm, A. A. Daleh, A. Hasan Munef, and G. Mojib, *Schema Matching Using Word-level Clustering for Integrating Universities' Courses*, in 2020 2nd Al-Noor International Conference for Science and Technology (NICST), Baku, Azerbaijan: IEEE, Aug. 2020, pp. 1–6. doi: 10.1109/NICST50904.2020.9280318.
6. I. Popova and H. A. A. Gardi, *Credit Card Fraud Detection: A Comparative Study of Resampling Strategies and Machine Learning Models*, in 2025 IEEE International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2025, pp. 1–10.

7. S. Ranjon Das, R. Bin Sulaiman, and U. Butt, *Comparative Analysis of Machine Learning Algorithms for Credit Card Fraud Detection*, FMDB Transactions on Sustainable Computing Systems, vol. 1, no. 4, pp. 225–244, Dec. 2023.
8. G. Moschini, R. Houssou, J. Bovay, and S. Robert-Nicoud, *Anomaly and Fraud Detection in Credit Card Transactions Using the ARIMA Model*, Engineering Proceedings, vol. 5, no. 1, p. 56, July 2021. doi: 10.3390/engproc2021005056.
9. M. Shanaa and S. Abdallah, *A Hybrid Anomaly Detection Framework Combining Supervised and Unsupervised Learning for Credit Card Fraud Detection*, F1000Research, vol. 14, p. 664, Dec. 2025. doi: 10.12688/f1000research.166350.2.
10. H. S. Alsagri, *Hybrid Machine Learning-Based Multi-Stage Framework for Detection of Credit Card Anomalies and Fraud*, IEEE Access, vol. 13, pp. 77039–77048, 2025. doi: 10.1109/ACCESS.2025.3565612.
11. A. Akbarzadeh and E. Shadkam, *The Study of Cuckoo Optimization Algorithm for Production Planning Problem*, International Journal of Computer-Aided Technologies (IJCAx), vol. 2, no. 3, pp. 1–11, July 2015.
12. R. Abbassi et al., *Cuckoo optimization algorithm via Grey Wolf Optimizer for usage in engineering optimization and optimal power flow with renewable energy sources*, Sci Rep, vol. 15, no. 1, p. 37629, Oct. 2025. doi: 10.1038/s41598-025-21515-3.
13. A. S. M. Aloqali, H. Najm, W. R. Abdulhussien, and M. S. Mahdi, *Image Encryption Using Modified Serpent Algorithm and Harris Hawks Optimization*, JoWUA, vol. 16, no. 1, pp. 154–171, Mar. 2025. doi: 10.58346/JOWUA.2025.I1.009.
14. L. E. Kadhim, A. T. Albu-Salih, A. H. Al-Fatlawi, M. Y. Jumaah, and H. Najm, *Efficient Hybrid Feature Engineering and Supervised Learning Approach for Network Traffic Classification in Intrusion Detection Systems*, International Journal of Intelligent Engineering & Systems, vol. 18, no. 5, pp. 233–245, 2025. doi: 10.22266/ijies2025.1031.21.
15. H. Najm, M. Salih Mahdi, and S. Mohsin, *Novel Key Generator-Based SqueezeNet Model and Hyperchaotic Map*, Data and Metadata, vol. 4, p. 743, Mar. 2025. doi: 10.56294/dm2025743.
16. R. Q. Malik, A. H. Al-Fatlawi, R. M. Alsharfa, B. K. Mohammed, M. S. A. Al-Ameer, and H. Najm, *A Novel Taneja Distance-based Classifier with PSO-Optimized Feature Selection for Efficient 5G Network Slicing*, International Journal of Intelligent Engineering & Systems, vol. 18, no. 6, pp. 638–651, 2025. doi: 10.22266/ijies2025.0731.40.
17. H. Handa, H. Ishibuchi, Y.-S. Ong, and K. C. Tan, Eds., *Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Volume 1*, vol. 1 in Proceedings in Adaptation, Learning and Optimization, Cham: Springer International Publishing, 2015. doi: 10.1007/978-3-319-13359-1.
18. E. Shadkam and M. Bijari, *Evaluation the Efficiency of Cuckoo Optimization Algorithm*, IJCSA, vol. 4, no. 2, pp. 39–47, Apr. 2014. doi: 10.5121/ijcsa.2014.4205.
19. C. A. Cobos-Lozada, H. Muñoz-Collazos, and R. Urbano-Muñoz, *Comparative Study of Cuckoo-Inspired Algorithms to Solve Large-Scale Continuous Optimization Problems*, Revista Facultad de Ingeniería, vol. 33, no. 69, 2024. doi: 10.19053/01211129.v33.n69.2024.17895.
20. K. Lobna, K. Mayssa, and H. Aziz, *The Cuckoo Optimization Algorithm to Resolve a Multi-Skills Inspectors Scheduling Problem*, Pakistan Journal of Life and Social Sciences, vol. 24, no. 1, pp. 11–19, 2026. doi: 10.57239/PJLSS-2026-24.1.002.
21. M. I. M. Al-Khuzaie and W. A. M. Al-Jawher, *Enhancing Brain Tumor Classification with a Novel Three-Dimensional Convolutional Neural Network (3D-CNN) Fusion Model*, J. Port. Sci. Res., vol. 7, no. 3, pp. 254–267, 2024. doi: 10.36371/port.2024.3.5.
22. Y. Xiong, Z. Zou, and J. Cheng, *Cuckoo search algorithm based on cloud model and its application*, Sci Rep, vol. 13, no. 1, p. 10098, Jun. 2023. doi: 10.1038/s41598-023-37326-3.
23. S. Sathyanarayanan, *Confusion Matrix-Based Performance Evaluation Metrics*, AJBR, pp. 4023–4031, Nov. 2024. doi: 10.53555/AJBR.v27i4S.4345.
24. M. I. M. Al-Khuzaay and W. A. M. Al-Jawher, *New Proposed Mixed Transforms: CAW and FAW and Their Application in Medical Image Classification*, Int. J. Innov. Comput., vol. 13, no. 1–2, pp. 15–21, 2023. doi: 10.11113/ijic.v13n1-2.414.
25. D. Espinoza, G. Ali, and C. Tarawneh, *AI-Based Hazard Detection for Railway Crossings*, in 2024 Joint Rail Conference, Columbia, South Carolina, USA: American Society of Mechanical Engineers, May 2024, p. V001T05A004. doi: 10.1115/JRC2024-124640.
26. M. I. Al-Khuzaie and W. A. M. Al-Jawher, *Enhancing Medical Image Classification: A Deep Learning Perspective with Multi Wavelet Transform*, J. Port. Sci. Res., vol. 6, no. 4, pp. 365–373, 2023. doi: 10.36371/port.2023.4.7.