International Journal of Data and Network Science 8 (2024) 463–472

Contents lists available at GrowingScience

International Journal of Data and Network Science

homepage: www.GrowingScience.com/ijds

# Efficient credit card fraud detection using evolutionary hybrid feature selection and random weight networks

**Enas Rawashdeh[a*], Nancy Al-Ramahi[b], Hadeel Ahmad[c] and Rawan Zaghloul[a]**

[a]*Management Information Systems, Albalqa' Applied University, Jordan*
[b]*Computer Science, AlZaytoonah Univeraity of Jordan, Jordan*
[c]*Computer Science, Applied Science Private University, Jordan*

C H R O N I C L E

*Article history*:
Received: July 6, 2023
Received in revised format: August 4, 2023
Accepted: September 10, 2023
Available online: September 10, 2023

*Keywords:*
*Feature Selection*
*Fraud Detection*
*Machine Learning*
*Classification*
*Credit Card*
*Random weight network*

A B S T R A C T

In the realm of financial security, the detection and prevention of credit card fraud has become paramount. With the ever-increasing reliance on digital transactions, the risk of fraudulent activities targeting credit card systems has grown significantly. To combat this, sophisticated techniques are required to swiftly identify and mitigate potential threats. Machine learning, a cornerstone of modern data analysis, has emerged as a powerful tool in this pursuit. By leveraging vast datasets and employing advanced algorithms, machine learning enables the automated scrutiny of transactions, distinguishing between legitimate and fraudulent activities with remarkable precision. This paper introduces an intelligent method for credit card fraud detection that relies on Competitive Swarm Optimization (CSO) and Random Weight Network (RWN). Additionally, the system includes an automated hybrid feature selection capability to identify the most pertinent features during the detection process. The experimental outcomes validate that this system can attain outstanding results in G-Mean, RUC, and Recall values.



## 1. Introduction

Credit card fraud detection has emerged as a critical challenge in modern financial transactions due to the increasing prevalence of online transactions and digital payment methods. The unauthorized use of credit cards for fraudulent activities poses substantial financial risks to individuals, businesses, and financial institutions. To combat this issue, sophisticated fraud detection systems are required to swiftly identify and prevent fraudulent transactions, ensuring the security and trustworthiness of financial operations (Cherif et al., 2023; Abdallah et al., 2016). Machine learning algorithms have revolutionized the field of fraud detection by offering automated data analysis and pattern recognition capabilities (Bin-Sulaiman et al., 2022; Masoud et al., 2021). These algorithms are capable of learning from historical transaction data, detecting unusual behaviors that may indicate fraudulent activities. Techniques such as decision trees, support vector machines, random forests, and neural networks have been harnessed to create predictive models that can efficiently classify transactions into legitimate and fraudulent categories (Shirgave et al., 2019; Jovanovic et al., 2022). An essential aspect of building effective fraud detection models is the selection of relevant features from the transaction data. Not all attributes contribute equally to the task of differentiating between legitimate and fraudulent transactions (Lima & Pereira, 2017). Feature selection aims to identify and retain the most influential attributes while discarding irrelevant or redundant ones. This process enhances the model's performance by improving its ability to capture intricate patterns associated with fraudulent activities. Methods like filter and wrapper approaches are commonly employed to carry out feature selection, enabling the system to achieve higher accuracy, reduced false positives and better generalization across various fraud scenarios (Rtayli & Enneya, 2020; Malik et al., 2022).

* Corresponding author.
E-mail address: enasfaisal@bau.edu.jo (E. Rawashdeh)

ISSN 2561-8156 (Online) - ISSN 2561-8148 (Print)

doi: 10.5267/j.ijdns.2023.9.009

In the filter approach, features are evaluated independently based on statistical measures such as information gain or chi-square (Mienye et al., 2023). This approach is computationally efficient and works well for large datasets. For instance, in the context of credit card fraud detection, the filter approach might entail ranking features based on their individual information gain values, selecting those with the highest scores. However, this method overlooks potential interactions between features and may lead to suboptimal selections when complex relationships exist in the data. On the other hand, the wrapper approach considers feature selection as an integrated part of the machine learning process. It utilizes the classification algorithm itself to evaluate subsets of features. This method takes feature interactions into account, offering a more holistic understanding of the data. For example, in fraud detection, a wrapper approach might involve utilizing a machine learning algorithm like Support Vector Machines (SVM) to iteratively select feature subsets that maximize the algorithm's performance (Rtayli & Enneya, 2020). Despite its effectiveness, the wrapper approach tends to be computationally intensive due to the need for repeated model training and evaluation. The filter approach is advantageous for its speed and simplicity, making it suitable for large-scale datasets. On the other hand, the wrapper approach leverages feature interactions to capture complex patterns, potentially leading to superior selections (Ileberi et al., 2022). However, the wrapper approach demands more computational resources. The choice between these methods depends on the dataset's complexity, available computational power, and the specific fraud detection goals (Esenogho et al., 2022).
In this study, we introduce a credit card fraud detection system that offers an automatic hybrid approach that combines filter and wrapper feature selection methods, while utilizing RWN as the underlying classifier. This innovative approach allows fraud detection system designers to identify the most influential features for detection, enhancing the robustness and efficiency of such systems. The key contributions of this work can be summarized as follows:

- Develop an efficient fraud detection model named HybridIG-CSO.
- Introduction of an automatic feature selection mechanism that identifies significant features using two techniques.
- Utilization of RWN as the foundational classifier, leveraging its potential for improved generalization.
- Automatically optimizing the number of neurons and weights of RWN, reducing the requirement for manual tuning.

The paper is organized as follows: Section 2 discusses important methods proposed for credit card fraud detection models. Section 3 provides an overview of the study's fundamentals. Section 4 presents a detailed explanation of the methodology and the proposed approach. Section 5 elaborates on the conducted experiments. Finally, Section 6 summarizes the key findings of this research and outlines potential future directions.

## 2. Related work

Numerous studies in literature have recognized Machine Learning (ML) as a pivotal tool for addressing credit card fraud detection issues (Verma et al., 2022; Karthika et al., 2022). These methods employ either traditional ML algorithms or delve into the realm of deep learning techniques (Zioviris et al., 2022; Shenvi et al., 2019). With the continuous growth of online financial transactions, effective fraud detection becomes increasingly crucial. However, this task comes with its own set of challenges, such as dealing with imbalanced data and ensuring scalability. ML techniques have emerged as a critical player in overcoming these challenges, addressing data imbalances through approaches like oversampling (Kasasbeh et al., 2023; Biswas & Debbarma, 2023) or undersampling (Zhang et al., 2019). Moreover, cost-sensitive learning strategies have been employed to assign varying misclassification costs to different classes, with a focus on improving the detection of fraudulent cases (Thai-Nghe et al., 2010).

In managing high-dimensional data, ML harnesses the power of feature selection to enhance model performance, reduce computational burdens, improve interpretability, and bolster generalization and robustness. Feature-selection methods encompass a range of techniques, including filter methods (Song et al., 2017), wrapper methods (Kohavi et al., 1997), and embedded methods (Liu et al., 2019), each selected based on the specific dataset characteristics and machine learning task at hand. The wrapper-based approach relies heavily on the choice of learning classifier and the optimization of the search strategy (Mienye et al., 2023). By meticulously selecting the appropriate classifier and refining the search strategy, the wrapper approach seeks to identify an optimal feature subset that maximizes the performance of the chosen ML model (Espinosa et al., 2023). This process, while promising, may entail substantial computational resources, underscoring the importance of thoughtful component selection for efficient and effective feature selection (Habibi et al., 2023). Metaheuristic techniques have significantly enhanced feature selection within wrapper approaches by effectively navigating vast feature spaces, fine-tuning model performance, adapting to diverse datasets and objectives, and tackling intricate optimization challenges, including fraud detection (Singh & Jain, 2020; Ahmad et al., 2022; Zhu et al., 2020; Abdel-Basset et al., 2018). Some prominent metaheuristics employed in machine learning for feature selection in fraud detection encompass genetic algorithms (GA) (Ileberi et al., 2022), particle swarm optimization (PSO) (Rawashdeh et al., 2021), the Ant Colony optimization (ACO) (Liu et al., 2009), the whale optimization algorithm (WOA) (Majhi, 2021), and differential evolution (DE) (Rakesh & Jana, 2023). Nevertheless, the surge in complex problems and practical applications has spurred interest in even more potent optimization algorithms. Finally, the literature demonstrates a notable inclination towards utilizing machine learning methods for feature selection in fraud detection, primarily due to their formidable learning capabilities (Bin-Sulaiman et al., 2022). Consequently, a hybrid wrapper approach addressing the aforementioned considerations is introduced. This approach leverages hybrid feature selection via IG and CSO, with RWN as the classifier. RWN offers rapid learning and superior test performance compared to gradient descent techniques used in training Single-Layer Feedforward Networks (SLFN) and traditional training methods. Furthermore, RWN optimization encompasses connection weights, hidden biases, and the number of hidden neurons, all without requiring manual parameter tuning.

## 3. Preliminaries

In the next section, every algorithm which has been employed in this kind of research is detailed.

### *3.1 Information Gain*

The information gain technique is a mathematical method used in feature selection for machine learning and data analysis. It measures the reduction in uncertainty, or entropy, about a target variable when a specific feature is known. This reduction in uncertainty is a key concept in information theory. Information gain is particularly useful to help select features that lead to the most informative splits in the data (Prasetiyowati et al., 2021). Mathematically, the information gain (IG) for a feature $X$ with respect to a target variable $Y$ can be calculated using the formula in Eq. (1):

$$IG(X|Y) = H(X) - H(X|Y) \quad (1)$$

where $H(Y)$ represents the entropy of the target variable $Y$ before considering feature $X$, and $H(Y|X)$ represents the conditional entropy of $Y$ given the values of feature $X$. Additionally, the calculations for entropy $H(X)$ and conditional entropy $H(X \mid Y)$ are outlined as follows in Eq. (2) and Eq. (3):

$$H(X) = -\sum_{x \in X} P(x) log_2 (x) \quad (2)$$

$$H(X|Y) = -\sum_{x \in X} P(x) \sum_{y \in Y} P(x|y)\, log_2 (P(x|y)) \quad (3)$$

where $p(x)$ is the proportion of instances with value $x$ for feature $X$, and $H(Y|X=x)$ is the entropy of $Y$ for instances where feature $X$ has value $x$. The information gain value measures how much knowing feature $X$ reduces the uncertainty in predicting the target variable $Y$. Features with higher information gain are preferred for splitting in decision trees, as they provide more valuable information for classification tasks.

### *3.2 Competitive Swarm Optimization*

CSO is an algorithm rooted in the original PSO technique, devised to tackle the issue of premature convergence that often arises when applying PSO to complex search spaces containing numerous local optima (Cheng et al., 2015).. Despite various proposed PSO modifications aiming to enhance its search capabilities in different problems, these often lead to increased complexity without effectively addressing the problem of premature convergence caused by gbest.

The distinctive advantage of CSO lies in its ability to counteract premature convergence by removing the influence of gbest and pbest associated with each particle. In traditional PSO, particle updates hinge on the particle's pbest and the global best (gbest). In contrast, CSO employs pairwise comparisons between particles from different swarms for updates. Within each comparison, one particle prevails as the winner, and the other becomes the loser. The winner integrates into the next generation population, while the loser incorporates essential insights gleaned from the winner. The fundamental distinction between CSO and PSO rests in CSO's lack of memory regarding prior generation evaluations, unlike PSO which relies on gbest and pbest. Consequently, CSO solely navigates its search through the competitive comparison process. Hence, with the assumption of having $k$ particles within the swarm (population), the CSO procedure initiates with a population consisting of randomly initialized particles denoted as $P(t)$, where '$t$' signifies the generation. Every potential solution is depicted by one of the swarm's particles. Each particle can be considered as a point represented by a position X within an n-dimensional space, represented as $X_i(t) = (x_{i1}(t), x_{i2}(t),..., x_{in}(t)))$ combined with a velocity V in n-dimensional space, expressed as $V_i(t) = (v_{i1}(t), v_{i2}(t),... , v_{in}(t)))$.

During each iteration, the swarm $P(t)$ is divided into two equal and randomly selected groups. Subsequently, CSO selects two particles, one from each group, and initiates a comparison or contest solely between these two particles. The outcome of this competition designates the 'winner' who is directly carried over to the next swarm generation, $P(t + 1)$, without any alterations. Meanwhile, the 'loser' particle undergoes an update procedure by assimilating information derived from the winner and is then also shifted to the next generation. This sequential process continues until no more particles remain to be compared.

The positions and velocities of the particles that emerged as winners and losers in the $i^{th}$ pairwise competition, relative to generation t, can be expressed as follows: the winning particle's position is denoted as $X_{wi}(t)$ and its velocity $V_{wi}$(t ; similarly, the losing particle's position is $X_{li}(t)$ and its velocity $V_{li}(t)$. Here, I fall within the range [1, $k$/2 ] , where $k$ denotes the total number of particles within the swarm. The adjustment of the losing particle is carried out utilizing Eq. (4) and Eq. (5).

$$V_{li}(t+1) = R_1(i,t)V_{li}(t) + R_2(i,t)\big(X_{Wi}(t) - X_{li}(t)\big) + \varphi R_3(i,t)\big(\bar{X}_{Wi}(t) - X_{li}(t)\big) \tag{4}$$

$$X_{li}(t+1) = X_{li}(t) + V_{li}(t+1) \tag{5}$$

Here, $R_1(i,t)$, $R_2(i,t)$ and $R_3(i,t)$ represent three vectors of randomly generated numbers sampled from the range [0,1]. $X_i(t)$ denotes the average position of the pertinent particles. These pertinent particles could encompass the entire swarm of particles or a predefined group of neighboring particles. The parameter φ governs the extent to which $\bar{X}_i$(t) influences the process.

### *3.3 Random Weight Network*

The RWN (Random Weight Network) network, originally introduced by Schmidt and his team in 1992 (Schmidt et al., 1992), aimed to enhance the computational efficiency of Single-Layer Feedforward Network (SLFN) learning algorithms, as elaborated upon in a study by Cao et al. in 2018 (2018). RWN was subsequently extended to generalize SLFNs into multi-hidden-layer feedforward networks, where each node can be seen as a subnetwork encompassing an extra group of hidden nodes. The fundamental architecture of the RWN network follows a completely connected architecture with only one hidden layer. In contrast to conventional gradient-descent techniques that require the configuration of various parameters such as learning rates and the number of training epochs, RWN simplifies this process by focusing on just one parameter: the count of hidden neurons. Furthermore, RWN begins by initializing random input weights and hidden layer biases, and it utilizes $N$ training samples to construct the hidden layer output matrix. Afterward, the output weights are determined using the Moore-Penrose (MP) generalized inverse. Considering a dataset of $N$ training samples, where each sample is represented as$(x_i, t_i)$, where: $x_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in \mathbb{R}^n$ and $t_i = [t_{i1}, t_{i2}, \dots, t_{im}]^T \in \mathbb{R}^m$, the expression for the output of SLFN comprising $L$ hidden layer nodes is formulated as shown in Eq. (6) (Huang et al., 2004):

$$\sum_{i=1}^{L} \boldsymbol{\beta}_j . g(w_i . x_j + b_i) = O_j\, , j = 1, \dots, N] \tag{6}$$

where $g(x)$ is the activation function, $w_j = [w_{j1}, w_{j2}, \dots, w_{jn}]^T$ is the weight vector connecting the $j^{th}$ hidden neuron to the $n$ input nodes, $\beta_j = [\beta_{j1}, \beta_{j2}, \dots, \beta_{jm}]^T$ is a set of output weights values which connects the $j^{th}$ hidden neurons with $m$ the output nodes (Huang et al., 2004). An SLFN, equipped with an activation function denoted as $g(x)$, and featuring $L$ hidden neurons, demonstrates its ability to perfectly approximate $N$ samples with zero error, implying that the summation of the absolute differences between the predicted outputs ($o_j$) and the actual targets ($t_j$) from $j$ = $1$ to $L$ results in zero $\sum_{j=1}^{L} \|o_j - t_j\| = 0$, i.e., In other words, there exist parameters $w_i$, $\beta_i$, *and* $b_i$ , which satisfy this condition, as described in Eq.(7) (Huang et al., 2004):

$$\sum_{i=1}^{L} \boldsymbol{\beta}_j . g(w_i . x_j + b_i) = t_j\, , j = 1, \dots, N] \tag{7}$$

The set of N rules mentioned above is defined as shown in Eq. (8).

$$H\beta = T \tag{8}$$

Where

$$H = \begin{Bmatrix} g(w_1 . x_1 + b_1) & \dots & g(w_L . x_1 + b_L) \\ . & & . \\ . & \dots & . \\ . & & . \\ g(w_1 . x_N + b_1) & \dots & g(w_L . x_N + b_L) \end{Bmatrix}_{N \times L} \tag{9}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ . \\ \beta_L^T \end{bmatrix}_{L \times m} \quad and \quad T = \begin{bmatrix} t_1^T \\ . \\ t_N^T \end{bmatrix}_{N \times m} \tag{10}$$

where $H$ is the hidden layer output matrix, $\beta$ determines the output weight matrix, and $T$ is the target matrix (Huang et al., 2004). Description of a simple learning algorithm of RWN can be provided in Algorithm 1.

| Algorithm 1: Pseudo-code of RWN |
|---|
| ***Input:*** *Training dataset* $N = \{ (x_j, t_j) \mid x_j \in R^n (1 \leq j \leq N)\}$; |
| *Activation function g();* |
| *Number of hidden neurons L;* |
| ***Output:*** *Output weights β;* |
| *for (i= 1 to L) do* |
| *Initialize weights* $w_i$ *and biases* $b_i$ *randomly;* |
| *Calculate the hidden layer output matrix H;* |
| *Return output weights β* |

## 4. Methodology

This research proposes the integration of both a filter-based and a wrapper-based approach within a hybrid method, aiming to eliminate irrelevant features and enhance the detection of credit card fraud. In our hybrid approach, the initial step involves utilizing the Information Gain (IG) technique as a filter-based method to rank the features within the credit card dataset. From this ranking, only the highest-ranked features are selected and passed to the subsequent wrapper algorithm. The wrapper-based technique employed here is CSO, chosen for its ability to explore the complex search space of potential feature subsets. CSO has demonstrated promising results in enhancing classification performance for various combinatorial problems. The learning algorithm utilized in this method is the RWN.

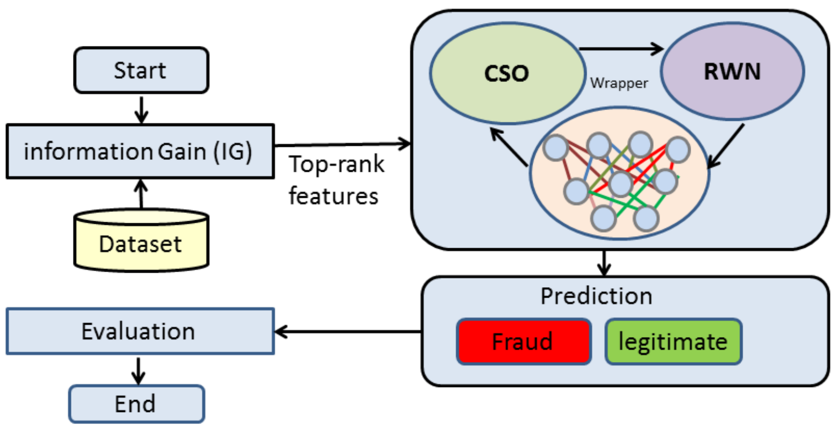


**Fig. 2.** Flowchart of proposed method

### *4.1 Design issues*

To optimize and resolve the issue at hand, there are several crucial matters that need to be tackled, including the representation of the solution and the definition of the fitness function. These are elaborated upon below:

- **Solution representation:** The representation of individuals within the metaheuristic algorithm is carefully crafted to symbolize the solution for the specific problem at hand as depicted in Fig. 2. In the context of our study, the CSO particle is encoded as a real vector encompassing the subsequent components: Firstly, a set of binary flags are included, signifying whether the corresponding features are chosen or not. Second part, a set of binary flags is incorporated to dictate the number of neurons in the hidden layer of the RWN. The third part, represent the RWN parameters, which encapsulate the values of input weights and hidden biases. Therefore, the size of the individual in the proposed approach can be determined using Eq. (11):

$$Length = (D \times K) + (2 \times K) + D \tag{11}$$

where $D$ signifies the number of features in the dataset, and $K$ represents the maximum number of hidden neurons. The elements $[W_{11}, \ldots, W_{DK}]$ within the individual correspond to the weights of the RWN network, with $K$ denoting the biases of the hidden layer.

- **Fitness function:** Formulation of the fitness function of the proposed approach in Eq. (12):

$$Fit = \sigma CLErr + \beta \frac{ft}{FT} + \gamma \frac{hd}{HD} \tag{12}$$

*CLErr* represents the error rate in classifying the RWN network, *fd* indicates the number of features identified using our method, *FT* represents the overall count of features in the dataset, *hd* denotes the count of hidden neurons set by the optimizer, and *HD* is the maximum allowable number of neurons in the RWN. The parameters $\alpha$, $\beta$, and $\gamma$ manage the impact of weights, aiming to enhance the reduction rate of features, curtail RWN complexity, and diminish the quantity of chosen features.

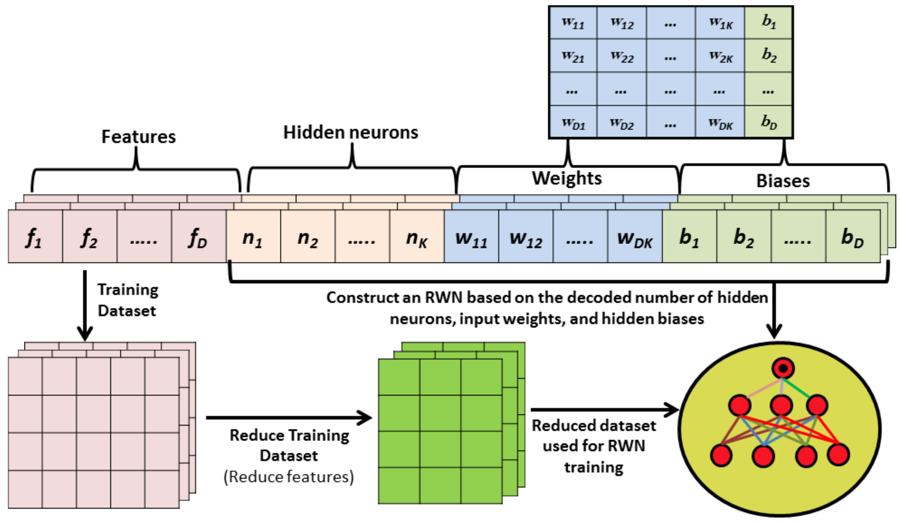


**Fig. 2.** Solution representation of proposed method.

*4.2 Proposed method procedure*

The process of the HybridIG-CSO algorithm proposed in this study can be outlined through the following steps:

- *Attribute Ranking using IG Technique:* In this initial stage, we leverage the Information Gain (IG) technique to assess attribute significance. This involves ranking attributes based on their contribution to the desired outcome. To establish a suitable threshold, we compute the standard deviation of IG values, a common practice for precise threshold determination (Roseline et al., 2022; Prasetiyowati et al., 2021). Attributes exceeding or meeting this threshold are retained, while those falling below it are discarded. This step ensures that only the most impactful attributes move forward.
- *Wrapper CSO Algorithm with RWN Learning:* Building on the initial attribute ranking, we introduce a powerful strategy. The top-ranked attributes, identified through IG, are integrated into the wrapper CSO algorithm. Within this framework, the RWN serves as the learning algorithm. The CSO's primary goal is to iteratively discover optimal subsets of features, achieved through a sequence of generations (Cheng et al., 2015).
- *Fitness Calculation:* At this stage, we initialize a swarm of CSO particles. Each particle encompasses elements to be optimized, such hidden biases, input weights, and the number of hidden neurons. The fitness of each particle is calculated, representing its ability to contribute effectively to the desired outcomes. This fitness calculation guides the subsequent steps towards optimal configuration.
- *CSO Particle Initialization and Competition:* The CSO Randomly initializes the number of individuals for each population, where candidate feature subsets are encoded as particles. The population is divided into two equal parts, each comprising k/2 individuals. The ensuing pairwise competition determines winners and losers among particles. Winners advance to the next generation, while losers undergo updates before moved to the next generation. The subsequent stage entails training diverse RWNs using each particle, followed by the computation of the fitness value for each feature subset. Aims to refine the particle population iteratively.
- *Iterative Refinement:* The methodology persists iteratively until a predefined maximum iteration count is reached. Throughout this process, the wrapper CSO algorithm continues its pursuit of the best feature subset and corresponding RWN configuration. The culmination of these efforts aims to achieve significantly enhanced prediction performance.

**5. Experimental Results and Discussion**

In this section, we delve into a comprehensive analysis to present the effectiveness of HybridIG-CSO method in classification tasks. All evaluations and comparisons were conducted on a computer equipped with an Intel (R) Core (TM) i7-5500U 2.40GHz processor with 8.0GB of RAM. All the algorithms were implemented using Python.

We established a population size of 50, with a maximum of 100 iterations conducted during the experimentation process. The training and testing stages employed a 10-fold cross-validation approach. Simultaneously, the proposed approach compared against several fundamental classifiers, which include Naïve Bayes (NB) (Kaur & Kumar, 2019), Random Forest (RF) (Xuan et al., 2018), and Support Vector Machine (SVM) (Rtayli & Enneya, 2020). Table 1 provides an overview of the dataset characteristics. In this table, "Abb." denotes the assigned dataset code, "#S" represents the sample count, "#PS" signifies the positive samples in each dataset, and "Data link" contains the link for dataset access.

**Table 1**
The dataset characteristics.

| Dataset | Abb. | #S | #PS | #Att | Data link |
|---|---|---|---|---|---|
| Loan Prediction | $D_1$ | 614 | 192 | 13 | https://github.com/Paliking/ML_examples/blob/master/LoanPrediction/train_u6lujuX_CVtuZ9i.csv |
| Creditcardcsvpresent | $D_2$ | 3075 | 448 | 12 | https://github.com/gksj7/creditcardcsvpresent |
| Default ofCreditCardClients | $D_3$ | 30000 | 6636 | 24 | http://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients. |
| European cardholders | $D_4$ | 284807 | 482 | 31 | https://kaggle.com/mlg-ulb/creditcardfraud |

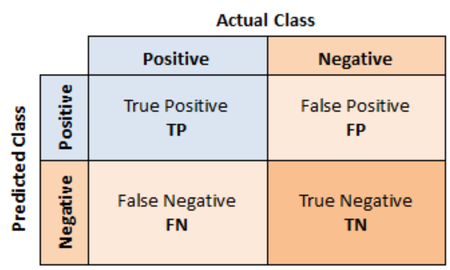


**Fig. 3.** Confusion matrix.

### *5.1 Performance Metrics*

The evaluation of the proposed method's performance involves a range of metrics derived from the confusion matrix shown in Fig. 3. These metrics are constructed using parameters like *TP* for true positive cases, *TN* for true negative cases, *FP* for false positive cases, and *FN* for false negative cases. The ensuing metrics can be calculated using the following equations:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \quad (13)$$

$$SE = \frac{TP}{TP + FN}, \quad (14)$$

$$SF = \frac{TN}{TN + FP}, \quad (15)$$

$$G - mean = \sqrt{SE \times SF}, \quad (16)$$

$$Percision = \frac{TP}{TP + FP}, \quad (17)$$

$$Recall = \frac{TP}{TP + FN}, \quad (18)$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (19)$$

Accuracy gauges overall classification accuracy; G-mean, calculated as the geometric mean of correct classification rates for both positive and negative classes; Sensitivity (SE) and Specificity (SP), representing correctly identified positive and negative cases; AUC (Area Under the Curve), assessing the model's differentiation capability via the ROC curve; Precision, indicating correctly predicted positive samples; Recall, reflecting accurately predicted positive samples among all actual positives; and F1 Score, a balanced assessment combining Precision and Recall's harmonic mean to effectively measure the model's performance.

### *5.2 Experiment I: Comparisons performance between HybridIG-CSO, RWN with filter approach, and RWN with CSO*

In this experiment, we have evaluated the HybridIG-CSO method by comparing it with three different techniques: the classical RWN, RWN with a filter-based approach (IG-RWN), and manually tuned CSO-RWN. This extensive comparison was conducted across four diverse datasets. The performance of the HybridIG-CSO method was assessed in comparison to the other approaches using six distinct criteria: Accuracy, Precision, Recall, AUC, F1, and G-mean. Best-performing results highlighted in bold. The results for HybridIG-CSO and the other methods are presented in Table 2. As depicted, for $D_1$ dataset, CSO-

RWN achieves the highest Accuracy, while IG-RWN leads in Precision. HybridIG-CSO excels in Recall, AUC, F1, and G-mean. Moving to $D_2$ dataset, CSO-RWN achieves top Precision, while classic-RWN takes the lead in Accuracy. Once again, HybridIG-CSO outperforms in Recall, AUC, F1, and G-mean. In $D_3$ dataset, HybridIG-CSO claims the top spot in Recall, AUC, F1, and G-mean, while CSO-RWN dominates Accuracy and Precision. Finally, for $D_4$ dataset, HybridIG-CSO secures the best results across all metrics. Generally, CSO-RWN performs admirably in Accuracy and Precision, while HybridIG-CSO shines in the remaining metrics, making them comparable options for different aspects of the problem.

**Table 2**
Performance of the Proposed method, classic RWN, IG-RWN, and CSO-RWN

| Dataset | Algorithm | Accuracy | Precision | Recall | AUC | F1 | G-Mean |
|---|---|---|---|---|---|---|---|
| $D_1$ | Classic-RWN | 0.901 | 0.841 | 0.660 | 0.725 | 0.740 | 0.808 |
| | IG-RWN | 0.903 | **0.955** | 0.634 | 0.722 | 0.762 | 0.800 |
| | CSO-RWN | **0.914** | 0.948 | 0.556 | 0.698 | 0.701 | 0.766 |
| | **HybridIG-CSO** | 0.885 | 0.799 | **0.681** | **0.726** | **0.735** | **0.810** |
| $D_2$ | Classic-RWN | **0.994** | 0.888 | 0.996 | 0.995 | 0.939 | 0.995 |
| | IG-RWN | 0.980 | 0.858 | 0.943 | 0.964 | 0.898 | 0.964 |
| | CSO-RWN | 0.970 | **0.963** | 0.739 | 0.874 | 0.835 | 0.863 |
| | **HybridIG-CSO** | 0.993 | 0.907 | **0.997** | **0.997** | **0.947** | **0.996** |
| $D_3$ | Classic-RWN | 0.856 | 0.652 | 0.617 | 0.631 | 0.634 | 0.757 |
| | IG-RWN | 0.780 | 0.518 | 0.644 | 0.571 | 0.574 | 0.727 |
| | CSO-RWN | **0.873** | **0.715** | 0.657 | 0.515 | **0.685** | 0.662 |
| | **HybridIG-CSO** | 0.837 | 0.609 | **0.672** | **0.637** | 0.639 | **0.773** |
| $D_4$ | Classic-RWN | 0.887 | 0.854 | 0.890 | 0.914 | 0.872 | 0.902 |
| | IG-RWN | 0.935 | 0.908 | 0.926 | 0.930 | 0.917 | 0.938 |
| | CSO-RWN | 0.964 | 0.953 | 0.967 | 0.957 | 0.960 | 0.957 |
| | **HybridIG-CSO** | **0.993** | **0.995** | **0.989** | **0.992** | **0.992** | **0.994** |

**Table 3**
Performance of proposed method with other classifiers.

| Dataset | Classifier | Accuracy | Precision | Recall | AUC | F1 | G-Mean |
|---|---|---|---|---|---|---|---|
| $D_1$ | NB | 0.897 | 0.882 | 0.618 | 0.708 | 0.727 | 0.785 |
| | RF | 0.900 | 0.869 | 0.623 | 0.720 | 0.726 | 0.794 |
| | SVM | **0.905** | **0.934** | 0.592 | 0.712 | 0.724 | 0.780 |
| | **HybridIG-CSO** | 0.885 | 0.799 | **0.681** | **0.726** | **0.735** | **0.810** |
| $D_2$ | NB | 0.990 | 0.889 | 0.995 | 0.996 | 0.940 | **0.996** |
| | RF | 0.978 | 0.859 | 0.944 | 0.965 | 0.899 | 0.965 |
| | SVM | 0.968 | **0.964** | 0.740 | 0.875 | 0.836 | 0.864 |
| | **HybridIG-CSO** | **0.993** | 0.907 | **0.997** | **0.997** | **0.947** | **0.996** |
| $D_3$ | NB | **0.844** | **0.627** | 0.636 | 0.629 | 0.631 | 0.760 |
| | RF | 0.842 | 0.625 | 0.635 | 0.626 | 0.630 | 0.758 |
| | SVM | 0.836 | 0.606 | 0.639 | 0.621 | 0.622 | 0.757 |
| | **HybridIG-CSO** | 0.837 | 0.609 | **0.672** | **0.637** | **0.639** | **0.773** |
| $D_4$ | NB | 0.925 | 0.919 | 0.925 | 0.905 | 0.922 | 0.917 |
| | RF | 0.946 | 0.937 | 0.944 | 0.884 | 0.940 | 0.945 |
| | SVM | 0.915 | 0.850 | 0.861 | 0.859 | 0.855 | 0.903 |
| | **HybridIG-CSO** | **0.993** | **0.995** | **0.989** | **0.992** | **0.992** | **0.994** |

### *5.3 Experiment II: Comparison with other classifiers*

In this experimental evaluation, we assess the effectiveness of HybridIG-CSO in the context of fraud classification using four distinct credit card datasets. Furthermore, we juxtapose its performance against other widely employed algorithms typically used as induction techniques in feature selection wrapper-based methods, namely Naïve Bayes (NB), Random Forest (RF), and Support Vector Machine (SVM). Table 3 offers a comparative overview of the outcomes achieved by various foundational classifiers across the four datasets.

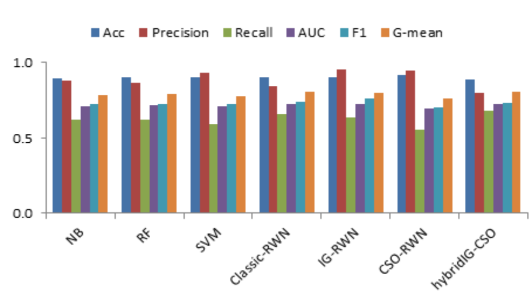


**Fig. 4.** Comparative analysis using the $D_1$.

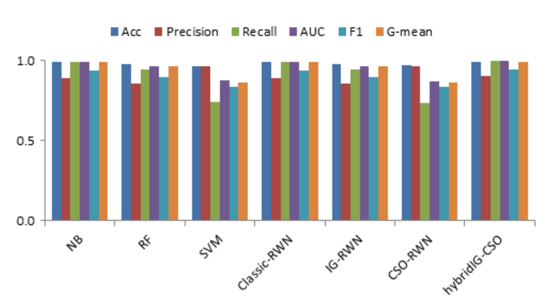


**Fig. 5.** Comparative analysis using the $D_2$.

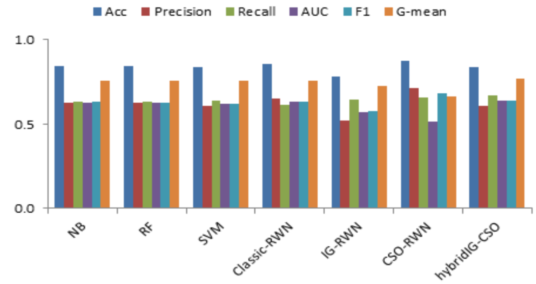


**Fig. 6.** Comparative analysis using the $D_3$.

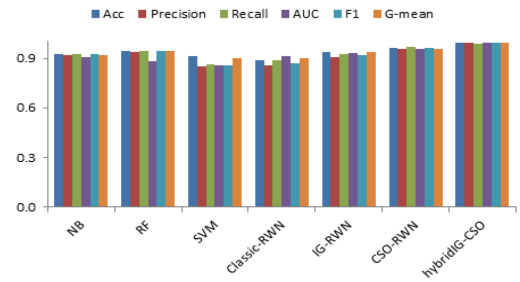


**Fig. 7.** Comparative analysis using the $D_4$.

For $D_1$, it becomes evident that HybridIG-CSO stands out, securing the top results in Recall, AUC, F1, and G-mean. Conversely, SVM attains the best results in terms of Accuracy and Precision. Shifting our focus to the $D_2$ dataset, HybridIG-CSO boasting the highest Accuracy, Recall, AUC, and F1. SVM, on the other hand, excels in terms of Precision, while NB and HybridIG-CSO share the same G-mean value. Transitioning to $D_3$, it's notable that NB achieves the highest scores in Accuracy and Precision, whereas HybridIG-CSO dominates in the remaining metrics. Finally, with respect to $D_4$, HybridIG-CSO achieved the best results across all the evaluation metrics. Figs. 4-7 depict the comparison of the different techniques across the four datasets.

## 6. Conclusion and future works

In this study, we introduced HybridIG-CSO for imbalanced classification problems, an innovative approach for credit card fraud detection that effectively combines IG and CSO within a framework utilizing RWN. The hybridization of filter and wrapper feature selection techniques empowers the model to identify and leverage the most critical attributes in fraud detection. The experimental results showcased HybridIG-CSO's consistent superiority over conventional classifiers like NB, RF, and SVM across multiple datasets. The HybridIG-CSO approach excelled particularly in metrics such as Recall, AUC, F1, and G-mean, demonstrating its potential in enhancing fraud detection. Future research will involve comparing our proposed method with alternative machine learning techniques and exploring opportunities for optimization and ensemble methods to enhance model performance.

## References

Abdallah, A., Maarof, M. A., & Zainal, A. (2016). Fraud detection system: a survey. *Journal of Network and Computer Applications, 58*, 90-113.

Abdel-Basset, M., Abdel-Fatah, L., & Sangaiah, A. K. (2018). Metaheuristic algorithms: A comprehensive review. *Computational intelligence for multimedia big data on the cloud with engineering applications*, 185-231.

Ahmad, H., Kasasbeh, B., Aldabaybah, B., & Rawashdeh, E. (2023). Class balancing framework for credit card fraud detection based on clustering and similarity-based selection (SBS). *International Journal of Information Technology*, *15*(1), 325-333.

Ahmad, H., Kasasbeh, B., AL-Dabaybah, B., & Rawashdeh, E. (2023). EFN-SMOTE: An effective oversampling technique for credit card fraud detection by utilizing noise filtering and fuzzy c-means clustering. *International Journal of Data and Network Science*, *7*(3), 1025-1032.

Bin Sulaiman, R., Schetinin, V., & Sant, P. (2022). Review of machine learning approach on credit card fraud detection. *Human-Centric Intelligent Systems*, *2*(1-2), 55-68.

Biswas, M., & Debbarma, S. (2022, June). An Efficient Approach for Credit Card Fraud Identification with the Oversampling Method. In *International Conference on Frontiers of Intelligent Computing: Theory and Applications* (pp. 273-286). Singapore: Springer Nature Singapore.

Cao, W., Wang, X., Ming, Z., & Gao, J. (2018). A review on neural networks with random weights. *Neurocomputing, 275*, 278-287.

Cheng, R., & Jin, Y. (2015). A Competitive Swarm Optimizer for Large Scale Optimization. IEEE Transactions on Cybernetics, 45(2), 191 - 204.

Cherif, A., Badhib, A., Ammar, H., Alshehri , S., Kalkatawi, M., & Imine, A. (2023). Credit card fraud detection in the era of disruptive technologies: A systematic review. *Journal of King Saud University - Computer and Information Sciences, 35*(1), 145-174.

Esenogho, E., Mienye, I. D., Swart, T. G., Aruleba, K., & Obaido, G. (2022). A Neural Network Ensemble With Feature Engineering for Improved Credit Card Fraud Detection. *IEEE Access, 10*, 16400 - 16407.

Habibi, A., Delavar, M. R., Sadeghian, M. S., Nazari, B., & Pirasteh, S. (2023). A hybrid of ensemble machine learning models with RFE and Boruta wrapper-based algorithms for flash flood susceptibility assessment. *International Journal of Applied Earth Observation and Geoinformation*, *122*, 103401.

Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2004, July). Extreme learning machine: a new learning scheme of feedforward neural networks. In *2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541)* (Vol. 2, pp. 985-990). Ieee.

Ileberi, E., Sun, Y., & Wang, Z. (2022). A machine learning based credit card fraud detection using the GA algorithm for feature selection. *Journal of Big Data*, *9*(1), 1-17.

Jiao, R., Nguyen, B. H., Xue, B., & Zhang, M. (2023). A Survey on Evolutionary Multiobjective Feature Selection in Classification: Approaches, Applications, and Challenges. *IEEE Transactions on Evolutionary Computation*.

Jovanovic, D., Antonijevic, M., Stankovic, M., Zivkovic, M., Tanaskovic, M., & Bacanin, N. (2022). Tuning machine learning models using a group search firefly algorithm for credit card fraud detection. *Mathematics*, *10*(13), 2272.

Karthika, J., & Senthilselvi, A. (2022). Credit Card Fraud Detection based on Ensemble Machine Learning Classifiers. *3rd International Conference on Electronics and Sustainable Communication Systems (ICESC).* Coimbatore: IEEE.

Kaur, B. J., & Kumar, R. (2020). A hybrid approach for credit card fraud detection using naive Bayes and voting classifier. In *Proceeding of the International Conference on Computer Networks, Big Data and IoT (ICCBI-2019)* (pp. 731-740). Springer International Publishing.

Kohavi, R., & John, G. (1997). Wrappers for feature subset selection. *Artificial Intelligence, 97*.

Lima, R. F., & Pereira, A. (2017). Feature Selection Approaches to Fraud Detection in e-Payment Systems. *International Conference on Electronic Commerce and Web Technologies (pp. 111-126).* Springer.

Liu, H., Zhou, M., & Liu, Q. (2019). An embedded feature selection method for imbalanced data classification. *IEEE/CAA Journal of Automatica Sinica*, *6*(3), 703-715.

Liu, O., Ma, J., Poon, P.-L., & Zhang, J. (2009). On an Ant Colony-Based Approach for Business Fraud Detection. *International Conference on Intelligent Computing. 5754,* pp. 1104–1111. Springer.

Majhi, S. K. (2021). Fuzzy clustering algorithm based on modified whale optimization algorithm for automobile insurance fraud detection*. Evolutionary Intelligence, 14*, 35-46.

Malik, E. F., Khaw, K. W., Belaton, B., Wong, W. P., & Chew, X. (2022). Credit Card Fraud Detection Using a New Hybrid Machine Learning Architecture. *Mathematics, 10*(9).

Masoud, M., Jaradat, Y., Rababa, E., & Manasrah, A. (2021). Turnover Prediction using Machine Learning: Empirical Study. *International Journal of Advances in Soft Computing & Its Applications, 13*(1).

Mienye, I. D., & Sun, Y. (2023). A Machine Learning Method with Hybrid Feature Selection for Improved Credit Card Fraud Detection. *Applied Sciences*, *13*(12), 7254.

Prasetiyowati, M. I., Maulidevi, N. U., & Surendro, K. (2021). Determining threshold value on information gain feature selection to increase speed and prediction accuracy of random forest. *Journal of Big Data*, *8*(1), 84.

Rakesh, D. K., & Jana, P. (2023). An improved differential evolution algorithm for quantifying fraudulent transactions. *Pattern Recognition, 141.*

Rawashdeh, E., Aljarah , I., & Faris, H. (2021). A cooperative coevolutionary method for optimizing random weight networks and its application for medical classification problems. *Journal of Ambient Intelligence and Humanized Computing, 12*, 321–342.

Roseline, J. F., Naidu, G. B. S. R., Pandi, V. S., alias Rajasree, S. A., & Mageswari, N. (2022). Autonomous credit card fraud detection using machine learning approach☆. *Computers and Electrical Engineering*, *102*, 108132.

Rtayli, N., & Enneya, N. (2020). Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization. Journal of Information Security and Applications, 55.

Schmidt, W. F., Kraaijveld, M. A., & Duin, R. P. (1992). Feedforward neural networks with random weights. *Conference B: Pattern Recognition Methodology and Systems (pp. 1-4). IEEE*.

Shenvi, P., Samant, N., Kumar, S., & Kulkarni, V. (2019). Credit Card Fraud Detection using Deep Learning. *IEEE 5th International Conference for Convergence in Technology (I2CT).* Bombay: IEEE.

Shirgave, S., Awati, C., More, R., & Patil, S. (2019). A Review On Credit Card Fraud Detection Using Machine Learning. *International Journal Of Scientific & Technology Research, 8*(10).

Singh, A., & Jain, A. (2020). Cost-sensitive metaheuristic technique for credit card fraud detection. *Journal of Information and Optimization Sciences*, *41*(6), 1319-1331.

Song, Q., Jiang, H., & Liu, J. (2017). Feature selection based on FDA and F-score for multi-class classification. *Expert Systems with Applications*, *81*, 22-27.

Thai-Nghe, N., Gantner, Z., & Schmidt-Thieme, L. (2010). Cost-sensitive learning methods for imbalanced data*. International Joint Conference on Neural Networks (IJCNN). Barcelona: IEEE*.

Verma, B. P., Verma, V., & Badholia, A. (2022). Hyper-Tuned Ensemble Machine Learning Model for Credit Card Fraud Detection. *International Conference on Inventive Computation Technologies (ICICT). IEEE*.

Xuan, S., Liu, G., Li, Z., Zheng, L., Wang, S., & Jiang, C. (2018, March). Random forest for credit card fraud detection. In *2018 IEEE 15th international conference on networking, sensing and control (ICNSC)* (pp. 1-6). IEEE.

Zhang, F., Liu, G., Li, Z., Yan, C., & Jiang, C. (2019). GMM-based Undersampling and Its Application for Credit Card Fraud Detection*. International Joint Conference on Neural Networks (IJCNN). Budapest: IEEE.*

Zhu, H., Liu, G., Zhou, M., Xie, Y., Abusorrah, A., & Kang, Q. (2020). Optimizing Weighted Extreme Learning Machines for imbalanced classification and application to credit card fraud detection. *Neurocomputing, 407,* 50-62.

Zioviris, G., Kolomvatsos, K., & Stamoulis, G. (2022, April 06). Credit card fraud detection using a deep learning multistage model. *The Journal of Supercomputing, 78*, 14571–14596.