



Paper Type: Original Article

Machine Learning Based Credit Card Fraud Detection Using the Binary Dragonfly Algorithm

Ali Ashkivar¹ , Rouzbeh Ghousi^{1,*} 

¹ Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran; Ghousi@iust.ac.ir.

Citation:

Received: 14 March 2024

Revised: 22 May 2024

Accepted: 22 June 2024

Ashkivar, A., & Ghousi, R. (2025). Machine learning based credit card fraud detection using the binary dragonfly algorithm. *Annals of optimization with applications*, 1(3), 153-166.

Abstract

Credit card fraud detection is crucial for financial institutions to prevent unauthorized transactions; however, it is hindered by challenges such as high-dimensional data and class imbalance. This study proposes a novel approach that integrates the Binary Dragonfly Algorithm (BDA) for Feature Selection (FS) with K-Nearest Neighbors (K-NN) for classification. Applied to a credit card fraud dataset, the method achieves 99.14% accuracy, 98.52% recall, 99.78% precision, and 99.15% F1-score, outperforming existing techniques. This approach provides an effective solution for fraud detection, not only enhancing the precision of fraud detection but also optimizing the model's efficiency. Future work could explore combining BDA with other metaheuristic algorithms or advanced classifiers to enhance performance further.

Keywords: Credit card fraud, Machine learning, Feature selection, Binary dragonfly algorithm, K-nearest neighbors.


1 | Introduction

The rapid growth of e-commerce and mobile internet technologies has significantly increased credit card transactions, driven by digital payment systems, online shopping, mobile banking, and enhanced security measures [1].

However, this rise has amplified credit card fraud, an unauthorized activity in electronic payment systems that is illegal and targets financial institutions [2]. Detecting such fraud using traditional methods is challenging, making advanced fraud detection models critical for academia and industry [3].

Real-world datasets for fraud detection often contain numerous features, many of which are irrelevant or redundant, degrading model performance [4]. Feature Selection (FS) addresses this by reducing dataset size

 Corresponding Author: Ghousi@iust.ac.ir

 <https://doi.org/10.48314/anowa.v1i3.47>



Licensee System Analytics. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0>).

while preserving accuracy, unlike feature construction, which creates new features. This paper focuses on FS, a preprocessing step that boosts classification performance by eliminating noisy or redundant data. FS methods include filters, which rely on data properties, and wrappers, which use a learning algorithm to evaluate subsets [5].

This study proposes a novel wrapper approach for credit card fraud detection, employing the Binary Dragonfly Algorithm (BDA) for FS and the K-Nearest Neighbors (KNN) classifier for evaluation. BDA, a swarm intelligence meta-heuristic, optimizes feature subsets, enhancing detection efficiency. The paper is organized as follows: Section 2 reviews the literature, Section 3 defines the problem, Section 4 is about the methodology, Section 5 presents experimental results, and Section 6 concludes the study.

2 | Literature Review

Credit card fraud detection has advanced considerably with the integration of machine learning and meta-heuristic algorithms, particularly in optimizing FS to improve classification performance. Early work by Duman and Ozcelik [6] in 2011 introduced Genetic Algorithm (GA) and Scatter Search (SS) alongside Logistic Regression and Neural Networks, achieving a better performance gain over traditional methods. This foundational study highlighted the potential of metaheuristics to enhance fraud detection.

In 2012, Ramakalyani and Umadevi [2] extended GA's application with a rule-based classifier, demonstrating its adaptability for FS across diverse domains. Vats et al. [7] in 2013 further refined GA by selecting feature subsets that maximized correlation with fraud labels, boosting accuracy while minimizing False Positives (FP).

A persistent challenge in this field is managing imbalanced datasets, where legitimate ones significantly outnumber fraudulent transactions. Benchaji et al. [8] in 2019 tackled this by integrating GA with K-means and Synthetic Minority Oversampling Technique (SMOTE), enhancing Random Forest's accuracy through balanced FS. Similarly, Saheed et al. [9] in 2020 applied GA across multiple classifiers, achieving an Area Under the Curve (AUC) of 1 and 100% accuracy with GA-Decision Tree, underscoring its versatility.

Recent advancements have shifted toward the application of swarm intelligence. Uma Rani et al. [10] in 2023 utilized Grey Wolf Optimization (GWO) with Random Forest and SMOTE-Edited Nearest Neighbors, improving AUC, Matthews Correlation Coefficient, and Kappa scores while reducing Mean Squared Error. Furlanetto et al. [11] in 2023 employed Artificial Bee Colony (ABC) with Random Forest, reducing features from 30 to 15 and increasing accuracy from 0.948 to 0.963.

Prabhakaran and Nedunchelian [1] in 2023 proposed a hybrid of Particle Swarm Optimization (PSO) and Opposition-based Cat Swarm Optimization, enhancing FS efficiency. Finally, Sikkandar et al. [12] in 2023 applied the Bat Optimization Algorithm for anomaly detection, reinforcing swarm intelligence's growing role in fraud detection.

The literature on credit card fraud detection highlights the growing use of meta-heuristic algorithms, such as GA, PSO, GWO, and ABC, to tackle fraud and the challenges posed by imbalanced datasets. Despite these advancements, a notable research gap persists: no prior study has investigated the BDA for FS in this context. Existing approaches often struggle with computational inefficiency or scalability when applied to high-dimensional fraud datasets and tend to pair FS with conventional classifiers like Random Forest or Decision Trees, limiting their adaptability.

This study fills this gap by leveraging the BDA, a meta-heuristic inspired by the dynamic swarming patterns of dragonflies, to optimize FS. BDA's strength lies in its ability to balance exploration (global search) and exploitation (local search), enabling it to pinpoint optimal feature subsets in complex datasets efficiently. Notably, this research combines BDA with KNN, a classifier less commonly employed in fraud detection, to improve detection accuracy and computational efficiency.

This novel integration not only addresses the limitations of prior methods but also offers a scalable and practical solution for real-time fraud detection systems. By pioneering the application of BDA in this domain,

this work expands the repertoire of meta-heuristic techniques and delivers a robust framework for improving fraud detection performance.

3 | Problem Definition

Credit card fraud detection is a significant challenge in the financial industry, as fraudulent transactions can result in substantial financial losses and erode consumer trust. Detecting such anomalies is particularly difficult due to the highly imbalanced nature of transaction datasets, where fraudulent activities constitute only a small fraction of the total transactions. Moreover, the presence of high-dimensional data with redundant or irrelevant features further complicates the task of building accurate and efficient detection models. To address these challenges, this study focuses on FS, a preprocessing step that identifies the most relevant features, thereby improving model performance and reducing computational complexity.

3.1 | Dataset

In this study, we utilize a credit card fraud detection dataset that comprises credit card transactions made by European cardholders over two days in September 2013. This dataset is widely employed in data mining and machine learning research, particularly for tasks such as anomaly detection, classification, FS, and addressing imbalanced learning challenges. It is publicly available on the Kaggle website [13] and other platforms.

The dataset comprises 284,807 transactions, of which only 492 are fraudulent, representing approximately 0.172% of the total transactions. This severe class imbalance poses a significant challenge for fraud detection models. The dataset includes 30 features: 28 anonymized features (V1 through V28), Time, and Amount. All features are numerical. The features V1 through V28 are the result of a Principal Component Analysis (PCA) transformation applied to the original features to ensure confidentiality. The Time feature represents the seconds elapsed between each transaction and the first transaction in the dataset, while the Amount feature indicates the transaction amount. The target variable, Class, is a binary label where 1 denotes a fraudulent transaction and 0 denotes a genuine one.

3.2 | Feature Selection

FS plays a vital role in machine learning by addressing the challenges posed by expansive feature spaces, which can hinder model performance. The selection of an appropriate FS technique varies depending on the specific problem at hand. FS entails identifying the most pertinent features in a dataset to eliminate noise and redundancy, thereby enhancing model effectiveness. Key benefits of applying FS include streamlining models, shortening training duration, and circumventing the curse of dimensionality. Finding an optimal feature subset remains a significant hurdle in FS tasks. The primary aim is to extract a group of M features from an initial set of N features (where $M < N$) while preserving essential information. Evaluating all possible combinations is impractical, as a dataset with N features yields 2^N subsets that is truly a computationally intensive process [5].

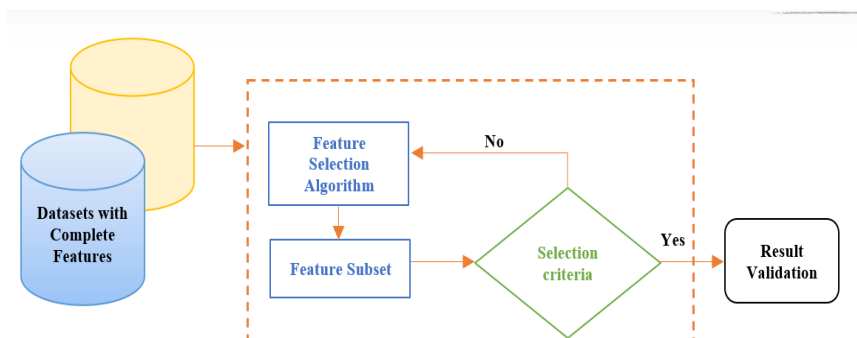


Fig. 1. Overall FS process [4].

The imbalanced nature of credit card transaction datasets and the computational complexity of FS present significant barriers to accurate fraud detection. This study addresses these challenges through an optimized FS strategy, outlined in the methodology that follows.

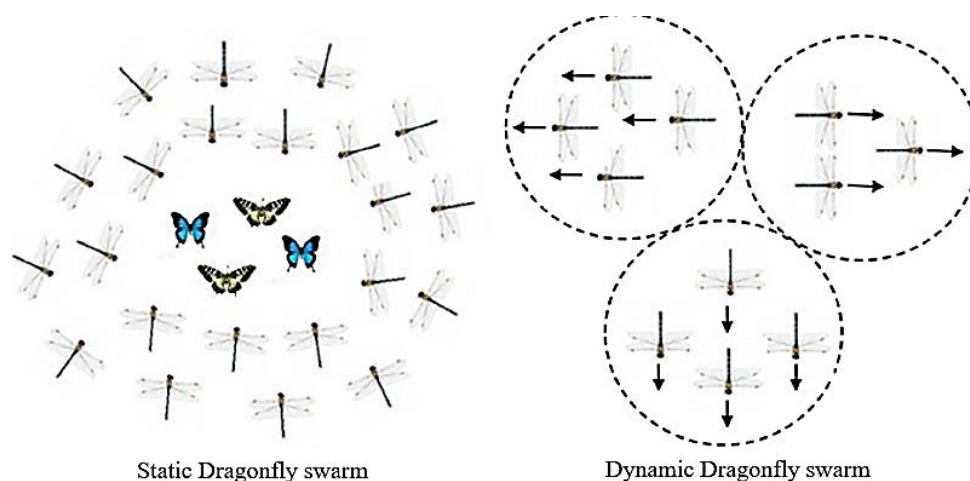
4 | Methodology

In this study, credit card fraud detection through a machine learning framework is addressed that optimizes FS using the BDA and employs the KNN classifier for classification. The section begins with an overview of the Dragonfly Algorithm (DA), a swarm intelligence optimization technique inspired by dragonfly swarming behaviors, known for balancing exploration and exploitation. Then, the BDA was adapted for binary optimization tasks, such as FS. Next, we detail how features are represented in BDA, followed by an explanation of the KNN classifier and its role in our study. Finally, we outline the fitness functions used to evaluate feature subsets, ensuring both accuracy and computational efficiency.

4.1 | Dragonfly Algorithm

DA is a swarm intelligence optimization method introduced by Mirjalili [14] in 2016, drawing inspiration from the swarming patterns observed in dragonflies. This nature-inspired meta-heuristic is designed to tackle a wide range of optimization challenges spanning various fields, including engineering, robotics, and image processing. In the context of this study, DA serves as the foundation for the BDA, which is employed for FS in this research.

DA replicates the dual swarming behaviors of dragonflies: static swarms, where dragonflies hover in small groups to feed, and dynamic swarms, where they migrate over long distances in larger formations. These behaviors mirror the algorithm's two primary phases: exploitation, which refines solutions within a local area, and exploration, which searches broadly across the solution space. The interplay between these phases enables

**Fig. 2. The static and dynamic swarming behaviors of dragonflies [15].**

Five basic primitive principles, presented in Fig. 3, are utilized to model the swarm behaviors of dragonflies as follows [15]. In the following equations, P represents the position of the current individual, P_j the position of the j th neighboring individual, and M the number of neighboring individuals.

The algorithm models dragonfly swarming through five fundamental principles that dictate their movement:

- I. Separation: Ensures individuals avoid collisions by maintaining distance from nearby dragonflies, calculated as the negative sum of differences between the current dragonfly's position and its neighbors' positions.

$$S_i = - \sum_{j=1}^M P - P_j. \quad (1)$$

- II. Alignment: Encourages dragonflies to synchronize their velocities with those of neighboring individuals, derived as the average velocity of the neighbors.

$$A_i = - \frac{\sum_{j=1}^M V_j}{M}. \quad (2)$$

- III. Cohesion: Drives dragonflies toward the swarm's center, determined by the difference between the average position of neighbors and the current individual's position.

$$C_i = \frac{\sum_{j=1}^M P_j}{M} - P. \quad (3)$$

- IV. Attraction: Guides dragonflies toward food sources, represented by the distance between the dragonfly's position and the food source's location. F_i represents the food source of the i th individual and F_p is the position of the food source in this formula.

$$F_i = F_p - P. \quad (4)$$

- V. Distraction: Steers dragonflies away from enemies, modeled as the position offset from an enemy's location. E_i denotes the position of the enemy of the i th individual and E_p denotes the enemy's position.

$$E_i = E_p + P. \quad (5)$$

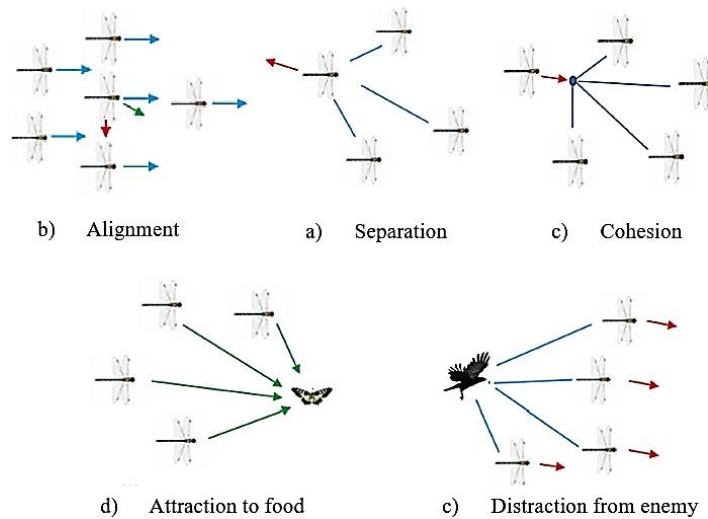


Fig. 3. Primitive corrective patterns between dragonflies in a swarm (different steps of the DA) [15].

These principles are mathematically formulated to update the positions of artificial dragonflies within the search space. Two key vectors govern the movement: ΔP is the step vector, akin to velocity in PSO, and P is the position vector. The step vector is adjusted using a weighted combination of the five behaviors, plus an inertia term from the previous iteration, as shown below:

$$\Delta P_i^{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + \omega \Delta P_i^t, \quad (6)$$

Where s , a , c , f , and e are weights for separation, alignment, cohesion, attraction, and distraction, respectively, while ω is the inertia weight, and t denotes the iteration. The position is then updated as:

$$P_i^{t+1} = P_i^t + \Delta P_i^{t+1}. \quad (7)$$

To promote exploration when no neighboring solutions are available, DA incorporates a random walk mechanism known as a Levy flight, updating the position as:

$$P_i^{t+1} = P_i^t + \text{Levy}(d) \times P_i^t, \quad (8)$$

Where d is the dimensionality of the search space, the Levy flight introduces controlled randomness to enhance the algorithm's ability to escape local optima.

The Levy flight is calculated by:

$$\text{Levy}(d) = 0.01 \times \frac{r_1 \times \sigma}{|r_2|^{\frac{1}{\beta}}}, \quad (9)$$

Where r_1 and r_2 are random vectors uniformly distributed in the range $[0,1]$, β is a constant, and σ is calculated as follows:

$$\sigma = \left(\frac{\Gamma(1 + \beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma \times 2^{(\frac{\beta-1}{2})} \times \beta \times (\frac{1+\beta}{2})} \right)^{\frac{1}{\beta}}. \quad (10)$$

$$\Gamma(x) = \int_0^{\infty} (t^{x-1} e^{-t}) dt. \quad (11)$$

When x is an integer:

$$\Gamma(x) = (x - 1)! \quad (12)$$

4.2 | Binary Dragonfly Algorithm

The BDA is an adapted form of the DA, tailored specifically for optimization tasks involving binary decision variables. Unlike the standard DA, which navigates continuous search spaces, BDA modifies this framework to operate within a discrete binary domain, making it an effective tool for FS. In this study, BDA is applied to the credit card fraud detection dataset to determine an optimal subset of features, thereby enhancing the performance of the machine learning model by reducing dataset dimensionality and filtering out irrelevant or redundant attributes.

Introduced by Abdel-Basset et al. [16], BDA was originally developed to tackle the 0-1 knapsack problem, which is a well-known NP-hard combinatorial optimization task. The algorithm employs a V-shaped transfer function to convert the continuous position updates of dragonflies into binary outcomes, enabling it to function in a binary search space. This transformation makes BDA well-suited for problems where solutions are represented as binary choices, such as selecting or discarding features.

In BDA, each dragonfly's position is encoded as a binary vector, with each bit corresponding to a feature in the dataset (1 signifying selection and 0 indicating exclusion). The algorithm iteratively refines these binary vectors by leveraging the five core principles of DA mentioned in the previous subsection. These principles guide the swarm's behavior, with a transfer function ensuring that position updates remain binary.

A key component of BDA is its V-shaped transfer function, which translates the continuous step vector into a probability that dictates whether a bit in the position vector changes. The function is expressed as:

$$T(\Delta P_i) = \left\lfloor \frac{2}{\pi} \arctan \left(\frac{\pi}{2} \Delta P_i \right) \right\rfloor. \quad (13)$$

Here, ΔP_i represents the step vector for the i th dragonfly, and the output ranges between 0 and 1. The position update follows this rule:

$$P_i^{t+1} = \begin{cases} 1 - P_i^t, & \text{if } r < T(\Delta P_i^{t+1}), \\ P_i^t, & \text{otherwise,} \end{cases} \quad (14)$$

Where r is a random value uniformly distributed between 0 and 1, and P_i^t is the current position of the i th dragonfly at iteration t . This probabilistic approach introduces randomness, allowing BDA to effectively explore the binary solution space while preventing premature convergence to suboptimal solutions.

In this research, BDA is implemented to select an optimal feature subset from the dataset's 28 features. The goal is to minimize the number of features while maximizing classification accuracy, addressing issues such as high dimensionality and class imbalance. The binary representation inherent to BDA aligns seamlessly with the FS process, where each feature is either included or excluded, streamlining the optimization task.

4.3 | K-Nearest Neighbor

The KNN algorithm is a straightforward yet practical supervised machine learning approach used for classification tasks. It operates by assigning a class label to a new data point based on the labels of its closest neighbors in the training dataset. As a non-parametric and instance-based method, KNN makes no assumptions about the underlying data distribution and defers computation until prediction time, relying solely on stored training data [17]. This learning strategy leverages a distance metric (Typically Euclidean distance) to measure similarity between data points, aiding in the identification of fraudulent credit card transactions while minimizing FP.

In this research, KNN serves as the classification component following FS by the BDA. By evaluating the reduced feature subset, KNN predicts whether a transaction is fraudulent or genuine, leveraging its proximity-based logic to enhance detection accuracy. The integration of BDA and KNN aims to mitigate some of KNN's limitations (such as sensitivity to irrelevant features) by ensuring only the most informative features are used, thus improving efficiency and effectiveness in credit card fraud detection.

4.4 | Fitness Function

The primary goal of this study is to identify an optimal, minimal subset of features that effectively captures the essential characteristics of the credit card fraud detection dataset, enabling efficient training of a machine learning model. The BDA is employed to search for this subset, necessitating a fitness function to assess the quality of each candidate feature combination. A well-designed fitness function must strike a balance between two competing objectives: maximizing classification accuracy (how well the subset predicts fraud) and minimizing complexity (the number of features selected). This balance ensures that high-performing, concise feature sets are favored, while poor accuracy or overly complex sets are penalized. The formulation of a fitness function varies based on the specific problem and desired trade-offs.

Among all design considerations and approaches, the linear weighted combination fitness function is proposed for this FS task. This approach combines the error rate and feature count linearly, assigning adjustable weights to each component. The function is defined as:

$$f(x) = \alpha \times \text{error}(x) + \beta \frac{\text{num_feat}(x)}{\text{Max_feat}}. \quad (15)$$

Here, x represents the feature subset, $\text{error}(x)$ is the classification error rate (estimated using a classifier like KNN), $\text{num_features}(x)$ is the number of selected features, and Max_features is the total number of features in the dataset (28 in this case). The parameters α and β are weights that prioritize accuracy and simplicity,

respectively. A higher α emphasizes reducing errors, while a higher β favors fewer features. This function is minimized, with lower values indicating superior subsets.

4.5 | Parameters Tuning

To ensure the efficacy of the proposed methodology, careful tuning of key parameters is essential. This study employs the Taguchi method, a statistical optimization technique, to systematically adjust the parameters of BDA, enhancing its convergence and stability in selecting an optimal feature subset from the credit card fraud dataset. This approach aims to make the FS process robust against variations in the dataset, thereby improving the overall performance of the subsequent KNN classifier.

The Taguchi method optimizes system performance by identifying parameter settings that minimize variability and align outcomes with desired targets [18]. It follows a two-step process of experimental design and analysis. Firstly, an orthogonal array is used to conduct a minimal set of experiments, efficiently testing multiple parameters and their interactions. This reduces the experimental burden while capturing the effects of each factor. After this step, results are evaluated using a quality loss function and signal-to-noise (S/N) ratio. The quality loss function is calculated by:

$$L(y) = k(y - T)^2. \quad (16)$$

This function measures the deviation of the output (y) from a target value (T), where k is a cost-related constant. The formula of S/N ratio with the “smaller-is-better” form is:

$$\frac{S}{N} \text{ Ratio} = -10 \log\left(\frac{\sum y^2}{n}\right). \quad (17)$$

Eqn. (17) is applied here to minimize the fitness value, where n is the number of observations.

Table 1. Parameters levels of the Taguchi method for BDA.

Parameter	Description	Level 1	Level 2	Level 3	Level 4	Level 5
N	Number of dragonflies	10	20	30	40	50
S	Separation weight	0.1	0.2	0.3	0.4	0.5
A	Alignment weight	0.1	0.2	0.3	0.4	0.5
C	Cohesion weight	0.1	0.2	0.3	0.4	0.5
F	Attraction weight	0.1	0.2	0.3	0.4	0.5
E	Distraction weight	0.1	0.2	0.3	0.4	0.5

Table 1 outlines the parameter settings tested to optimize the BDA's performance in FS. It includes six parameters, each evaluated across five levels. The parameters reflect BDA's swarm dynamics and computational aspects, while the weights influence the balance between exploration and exploitation. The range of levels ensures comprehensive coverage, accommodating the dataset's complexity (28 features, 284,807 transactions).

4.6 | Performance Metrics

In this study, the effectiveness of the proposed methodology is evaluated using a set of well-established performance metrics. These metrics serve as critical benchmarks to guide the optimization of feature subsets and the tuning of the KNN classifier, ensuring the model accurately detects fraudulent credit card transactions while managing the challenges of an imbalanced dataset. The following performance metrics are utilized throughout the methodology to evaluate feature subsets and the final classifier:

Accuracy: This metric measures the overall correctness of the model, calculated as the ratio of correctly predicted transactions (both fraudulent and genuine) to the total number of transactions. It is defined as:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}. \quad (16)$$

True Positives (TP) is the number of correctly identified frauds, True Negatives (TN) is the number of correctly identified genuine transactions, FP is the number of genuine transactions misclassified as frauds,

and False Negatives (FN) is the number of frauds missed. Accuracy is a primary indicator of model performance, but it can be misleading in imbalanced datasets, necessitating the use of additional metrics.

Precision: This metric quantifies the proportion of predicted frauds that are actually fraudulent, providing insight into the model's reliability when flagging transactions. It is expressed as:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (17)$$

High precision is crucial in fraud detection to minimize false alarms and reduce unnecessary investigations by financial institutions.

Recall (sensitivity)

Also known as the true positive rate, recall measures the proportion of actual frauds correctly identified by the model, calculated as:

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

Given the severe consequences of missing fraudulent transactions, recall is prioritized to ensure the model captures as many frauds as possible, particularly in the context of the dataset's extreme imbalance (0.1728% frauds).

F1-score

The F1-score provides a balanced evaluation by computing the harmonic mean of precision and recall, defined as:

$$F1_{\text{score}} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (19)$$

This metric is particularly valuable in this study, as it accounts for the trade-off between precision and recall, offering a single score to optimize when dealing with imbalanced classes.

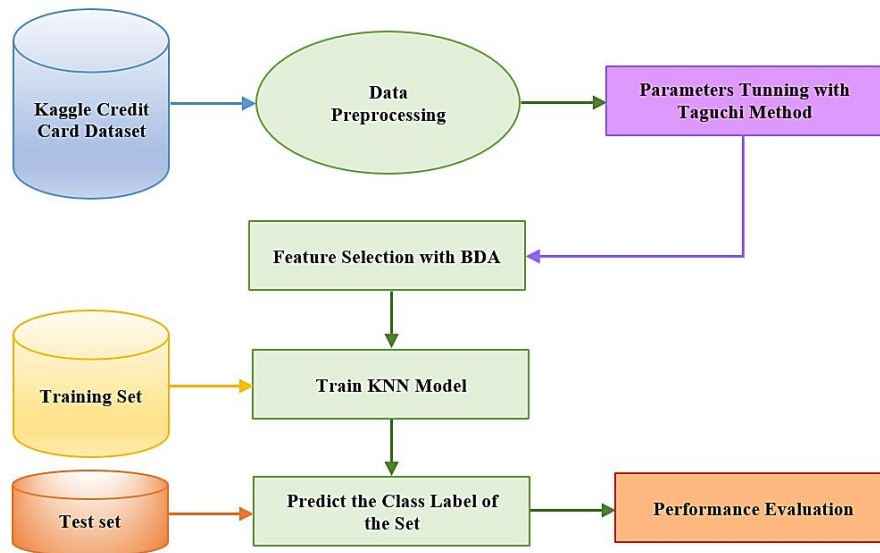


Fig. 4. Framework of the proposed credit card fraud detection.

5 | Results and Discussions

This section presents the experimental outcomes of the proposed methodology, which leverages the BDA for FS and KNN for classification to detect fraudulent credit card transactions. To tackle the class imbalance in the Kaggle Credit Card dataset, five oversampled datasets (named OS1 to OS5) are generated, each balanced with an equal number of fraud and non-fraud samples. For each dataset, we ran the BDA-KNN

model five times and selected the best-performing run based on accuracy, precision, recall, and F1-score. This process yielded five best runs (one per dataset). The performance metrics for the best run of each oversampled dataset are presented in *Table 2*.

These metrics reflect the model's effectiveness in classifying fraudulent transactions. The results show consistently high performance across all datasets. OS1 achieved the highest accuracy (99.20%) and F1-score (99.19%), reflecting a strong balance between precision and recall. OS2 stands out with a recall of 99.90%, though its precision (98.23%) is slightly lower than others. These variations suggest differences in how each oversampled dataset influenced the model's decision boundaries.

Table 2. Performance metrics for the best run of each oversampled dataset.

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
OS1	99.20	98.68	99.69	99.19
OS2	99.05	98.23	99.90	99.06
OS3	99.15	98.56	99.81	99.18
OS4	99.15	98.63	99.70	99.16
OS5	99.15	98.49	99.80	99.14

To explore the optimization dynamics of the BDA-KNN model, we examined the convergence behavior for each dataset. *Fig. 5 to 9* illustrate the convergence plots for the best-performing run of OS1 to OS5, respectively, with iterations on the x-axis and fitness values on the y-axis. All of them are tested within 100 iterations.

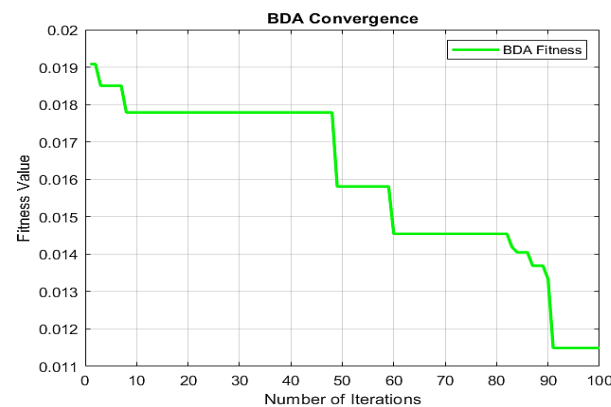


Fig. 5. Best run of OS1 convergence plot.

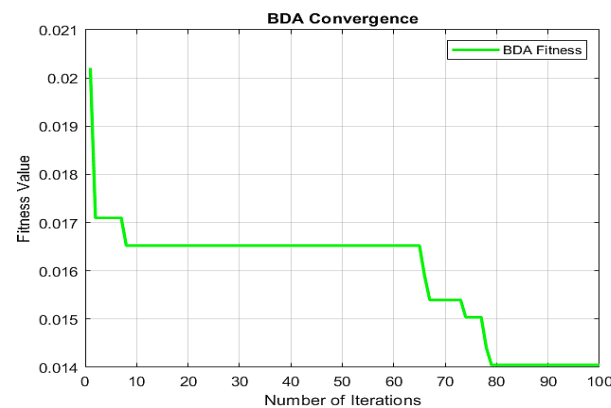


Fig. 6. Best run of OS2 convergence Plot.

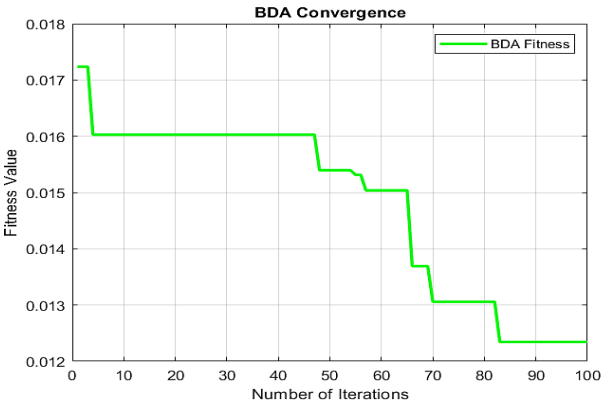


Fig. 7. Best run of OS3 convergence Plot.

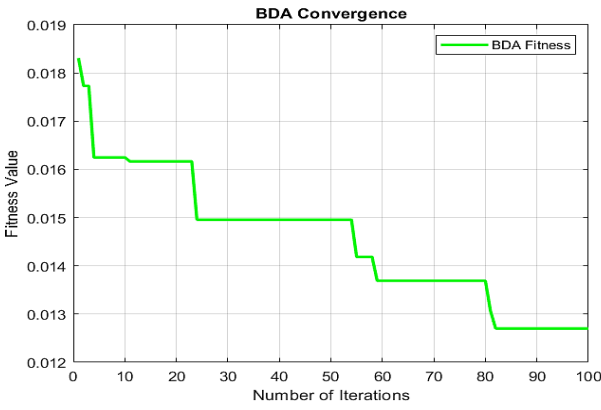


Fig. 8. Best run of OS4 convergence Plot.

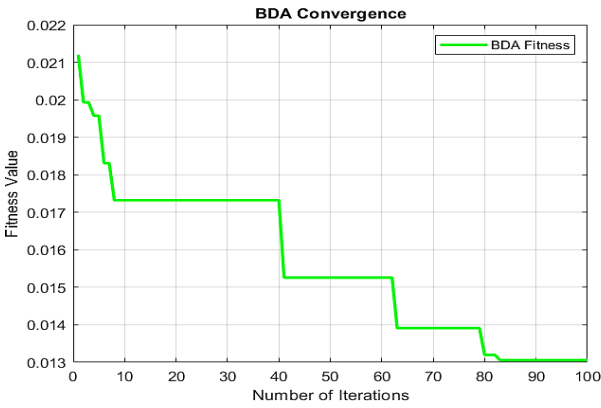


Fig. 9. Best run of OS5 convergence Plot.

These plots reveal that while all datasets achieved stable solutions, the speed and pattern of convergence differed, likely due to variations in oversampling or feature distributions.

Also, the results demonstrate that BDA, when tuned with the Taguchi method, effectively reduces dimensionality while maintaining high detection rates, validated by the plots' convergence trends. As shown in the figures, the best fitness value can be seen in Fig. 1, which is near 0.011.

As the final results, the average of each metric across the five best runs is computed to summarize the BDA-KNN model's overall performance. These averages are reported in Table 3.

Table 3. Average performance metrics for the BDA-KNN model.

Metric	Final Value (%)
Accuracy	99.14
Precision	98.52
Recall	99.78
F1-Score	99.15

The average accuracy of 99.14% highlights the model's reliability in classifying transactions. The precision of 98.52% indicates exceptional performance in minimizing FP, while the recall of 99.78% ensures most fraud cases are detected. The F1-score of 99.15% reflects a well-balanced model, making it highly effective for fraud detection.

6 | Conclusion

Imbalanced datasets and high-dimensional feature spaces pose significant challenges for credit card fraud detection. This paper introduces a novel machine learning approach combining the BDA for FS and KNN for classification. BDA, a metaheuristic inspired by dragonfly swarming behavior, excels at optimizing feature subsets in large datasets, enhancing detection efficiency. This study addresses the challenges posed by imbalanced datasets and high-dimensional feature spaces in credit card transaction data, aiming to improve detection accuracy while minimizing computational complexity.

The proposed BDA-KNN method achieved an impressive average accuracy of 99.14%, with a precision of 98.52%, a recall of 99.78%, and an F1 score of 99.15%, demonstrating its effectiveness for credit card fraud detection. These results highlight the capability of the BDA to optimize FS, paired with the KNN classifier, in addressing challenges such as imbalanced datasets and high-dimensional feature spaces.

Future work could refine BDA parameters (e.g., swarm size, behavioral weights) to boost performance. Comparing BDA with methods such as GA or PSO, or integrating hybrid models, may further optimize the FS. Testing alternative classifiers (e.g., Random Forests, Support Vector Machines) could enhance the performance.

Addressing class imbalance through oversampling or cost-sensitive learning, and exploring feature interpretability, offer additional avenues to strengthen fraud detection and prevention. Given the challenge of class imbalance in the dataset, applying advanced techniques like implementing cost-sensitive learning could enhance the model's ability to detect fraudulent transactions.

Finally, delving deeper into the interpretability of the selected features might provide valuable insights into the transaction attributes most indicative of fraud, paving the way for more effective fraud prevention strategies.

Author Contribution

The author was solely responsible for the conception and design of the study, development of the methodology, implementation of the computational framework, validation of the results, sensitivity analyses, and preparation of the manuscript.

Funding

This work was conducted without any financial support from funding agencies in the public, commercial, or non-profit sectors.

Data Availability

All data generated or analyzed during this study are included in this published article.

Conflicts of Interest

The author declares that there are no conflicts of interest relevant to the content of this article.

References

- [1] Prabhakaran, N., & Nedunchelian, R. (2023). Oppositional cat swarm optimization-based feature selection approach for credit card fraud detection. *Computational intelligence and neuroscience*, 2023(1), 2693022.
- [2] RamaKalyani, K., & UmaDevi, D. (2012). Fraud detection of credit card payment system by genetic algorithm. *International journal of scientific & engineering research*, 3(7), 1–6. <https://b2n.ir/qw1814>
- [3] Chaudhary, K., Yadav, J., & Mallick, B. (2012). A review of fraud detection techniques: Credit card. *International journal of computer applications*, 45(1), 39–44. <https://www.academia.edu/download/74530838/pxc3878991.pdf>
- [4] Agrawal, P., Abutarboush, H. F., Ganesh, T., & Mohamed, A. W. (2021). Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019). *Ieee access*, 9, 26766–26791. <https://doi.org/10.1109/ACCESS.2021.3056407>
- [5] Mafarja, M. M., Eleyan, D., Jaber, I., Hammouri, A., & Mirjalili, S. (2017). Binary dragonfly algorithm for feature selection. *2017 international conference on new trends in computing sciences (ICTCS)* (pp. 12–17). IEEE. <https://doi.org/10.1109/ICTCS.2017.43>
- [6] Duman, E., & Ozcelik, M. H. (2011). Detecting credit card fraud by genetic algorithm and scatter search. *Expert systems with applications*, 38(10), 13057–13063. <https://doi.org/10.1016/j.eswa.2011.04.110>
- [7] Vats, S. A. T. V. I. K., Dubey, S. K., & Pandey, N. K. (2013). Genetic algorithms for credit card fraud detection. In *International conference on education and educational technologies* (pp. 42–53). Institute of Technology and Management. <https://b2n.ir/ke2773>
- [8] Benchaji, I., Douzi, S., & El Ouahidi, B. (2018). Using genetic algorithm to improve classification of imbalanced datasets for credit card fraud detection. In *International conference on advanced information technology, services and systems* (pp. 220-229). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-11914-0_24
- [9] Saheed, Y. K., Hambali, M. A., Arowolo, M. O., & Olasupo, Y. A. (2020). Application of GA feature selection on Naive Bayes, random forest and SVM for credit card fraud detection. In *2020 international conference on decision aid sciences and application (DASA)* (pp. 1091-1097). IEEE. <https://doi.org/10.1109/DASA51403.2020.9317228>
- [10] Rani, V. U. ., Saravanan, V. ., & Tamilselvi, J. J. . (2023). A hybrid grey wolf-meta heuristic optimization and random forest classifier for handling imbalanced credit card fraud data. *International journal of intelligent systems and applications in engineering*, 11(9s), 718–734. <https://ijisae.org/index.php/IJISAE/article/view/3220>
- [11] Furlanetto, G. C., Gomes, V. Z., & Breve, F. A. (2023). Artificial bee colony algorithm for feature selection in fraud detection process. In *international conference on computational science and its applications* (pp. 535-549). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-36805-9_35
- [12] Sikkandar, H., Saroja, S., Suseandhiran, N., & Manikandan, B. (2023). An intelligent approach for anomaly detection in credit card data using bat optimization algorithm. *Inteligencia artificial*, 26(72), 202–222. <https://doi.org/10.4114/intartif.vol26iss72pp202-222%0A>
- [13] Credit Card Fraud Detection Dataset, Kaggle. (2022). *Credit card fraud detection dataset*, Kaggle. <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>
- [14] Mirjalili, S. (2016). Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural computing and applications*, 27(4), 1053–1073. <https://doi.org/10.1007/s00521-015-1920-1>
- [15] Meraihi, Y., Ramdane-Cherif, A., Acheli, D., & Mahseur, M. (2020). Dragonfly algorithm: A comprehensive review and applications. *Neural computing and applications*, 32(21), 16625–16646. <https://doi.org/10.1007/s00521-020-04866-y>

- [16] Abdel-Basset, M., Luo, Q., Miao, F., & Zhou, Y. (2017). Solving 0–1 knapsack problems by binary dragonfly algorithm. In *International conference on intelligent computing* (pp. 491-502). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-63315-2_43
- [17] Zhang, Z. (2016). Introduction to machine learning: K-nearest neighbors. *Annals of translational medicine*, 4(11), 218. <https://doi.org/10.21037/atm.2016.03.37>
- [18] Roy, R. K. (2010). *A primer on the Taguchi method*. Society of manufacturing engineers. https://books.google.com/books/about/A_Primer_on_the_Taguchi_Method_Second_Ed.html?id=k5VBsRZfzQsC