Enhancing Grey Wolf Optimizer for Imbalanced Feature Selection in Diabetes Prediction via Chaotic Initialization

Abstract. Feature selection is a crucial pre-processing step in machine learning, particularly for high-dimensional datasets and imbalanced classification problems. It aims to identify a minimal subset of relevant features that significantly improve model performance and interpretability. This article proposes an enhanced Grey Wolf Optimizer (GWO) for feature selection, integrating chaotic maps for population initialization. The proposed approach, termed GWO-CI (GWO with Chaotic feature Initialization), is applied to an imbalanced diabetes prediction dataset, utilizing a Random Forest classifier as the evaluation model. Experimental results demonstrate that chaotic initialization, specifically using Logistic, Tent, and Sine maps, can lead to more diverse initial populations, potentially improving the exploration capabilities of GWO and yielding superior feature subsets compared to standard random initialization, as evidenced by enhanced classification metrics (F1-score) on the imbalanced dataset.

Keywords: Feature Selection, Grey Wolf Optimizer, Chaotic Initialization, Imbalanced Dataset, Random Forest, Diabetes Prediction.

1 Introduction

In the time of big data, datasets often contain a large number of features. Many of these features may be redundant, irrelevant, or noisy. Such features can degrade the performance of machine learning models, increase computational complexity, and make models harder to interpret. Feature selection addresses these challenges by identifying a subset of the most informative features, leading to more accurate, efficient, and robust predictive models.

Meta-heuristic algorithms, inspired by natural evolution, have emerged as powerful tools to solve complex optimization problems, including feature selection. The Grey Wolf Optimizer (GWO) is a recent swarm intelligence algorithm that mimics the hunting behavior and social hierarchy of grey wolves [1]. Its simplicity and effectiveness have led to its successful application in various domains. However, like many meta-heuristic algorithms, the performance of GWO is highly dependent on the quality and diversity of their initial population.

The initialization of the Grey Wolf Optimizer (GWO) plays a crucial role in its performance for feature selection. Many studies have focused on improving GWO's initial population to enhance its global search capability and avoid premature convergence [2]. For instance, a novel two-phase crossover operator has been integrated with GWO, where the initialization phase randomly generates the population, with subsequent crossover phases designed to improve exploitation and enhance classification accuracy by selecting informative features [9]. Another adaptive mechanism-based GWO (AMGWO) introduced a new nonlinear parameter control strategy and

an adaptive fitness distance balance mechanism to accelerate convergence and prevent premature convergence, which inherently relies on an effective initial population [10]. An improved GWO with Ant Colony Optimization (ACO) for population initialization has been proposed to overcome suboptimal initial solutions in P2P lending default prediction, showing improved accuracy and stability compared to the standard GWO [7]. Similarly, hybrid approaches combining GWO with other meta-heuristic algorithms, such as Genetic Algorithm (GA), often incorporate chaotic maps and Opposition-Based Learning (OBL) for more uniformly distributed population initialization, aiming to enhance diversity and mitigate premature convergence in high-dimensional hyperspectral image classification [8]. The idea is that a well-initialized population allows the GWO to explore the search space more effectively from the outset, leading to better feature subsets.

A poorly initialized population can lead to premature convergence to suboptimal solutions or slow convergence rates. To alleviate this problem, chaotic maps have been proposed as a non-random, yet sensitive to initial conditions, alternative for initializing populations. Chaotic systems, despite being deterministic, exhibit highly complex and unpredictable behaviors that can effectively explore the search space.

This article proposes the integration of chaotic initialization into the Grey Wolf Optimizer for feature selection, specifically adapted for imbalanced datasets. The proposed GWO-CFI approach aims to:

1. Leverage the global exploration capabilities of chaotic maps for initial population diversity.
2. Employ the GWO's balance between exploration and exploitation for effective feature subset search.
3. Utilize a Random Forest classifier as a robust evaluation model within a wrapper-based feature selection framework.

The effectiveness of GWO-CFI is demonstrated on the "Diabetes Prediction Dataset" from Kaggle, comparing the performance of Logistic, Tent, and Sine map initializations against standard random initialization using weighted F1-score as the primary classification metric. The dataset used in this study exhibits significant class imbalance, which is a common characteristic of real-world medical data. This study evaluates the method's performance directly on this imbalanced distribution without applying any explicit data rebalancing techniques.

The rest of the paper is organized as follows: Section 2 provides a detailed overview of the Standard Grey Wolf Optimizer (GWO). Section 3 introduces the concept of Chaotic Maps and details the Logistic, Tent, and Sine maps used for initialization. Section 4 formulates Feature Selection as a Binary Optimization Problem. Section 5 describes the Random Forest Classifier used for evaluation. Section 6 presents the Proposed Methodology (GWO-CFI), including data preprocessing, the fitness function, and evaluation metrics. Section 7 outlines the Experimental Setup, detailing the dataset, preprocessing steps, and algorithm parameters. Section 8 discusses the Results and Discussion of the experiments. Finally, Section 9 concludes the paper and suggests directions for Future Work.

2 Standard Grey Wolf Optimizer (GWO)

The Grey Wolf Optimizer (GWO) is a recent swarm intelligence meta-heuristic algorithm proposed by Mirjalili et al. in 2014 [2]. It mimics the leadership hierarchy and hunting mechanism of grey wolves in nature [3]. A grey wolf pack typically consists of four main types of wolves:

- Alpha: The dominant leader, responsible for decision-making regarding hunting.
- Beta: The second in command, assisting the alpha in decision-making and acting as a disciplinarian.
- Delta: Subordinate to alpha and beta, but dominant over omega. Deltas are responsible for scouting, guarding, etc.
- Omega: The lowest-ranking wolves, following the directives of the higher-ranking wolves.

In the GWO algorithm, the best solution found so far is considered the alpha. The second and third best solutions are designated as beta and delta, respectively. The remaining candidate solutions are assumed to be omega. The omega wolves then adjust their positions and improve iteratively based on the guidance of these leading wolves. The hunting process in GWO (optimization process) involves three main steps: encircling prey, hunting, and attacking prey [1].

During iterations, each wolf i is represented by a position vector $\mathrm{X_i(t)} = \{\mathrm{X_{i1}}, \mathrm{X_{i2}}, \dots, \mathrm{X_{id}}\}$ in a $\mathrm{d}-$dimensional space. This vector consists of binary values, where d signifies the number of variables in the problem being solved [3].

The wolf population is organized as a matrix with dimensions $N \times \mathrm{d}$, where N is the number of wolves. Each wolf's position within this matrix is then assessed using a fitness function $\mathrm{f(X_i(t))}$.

Grey wolves encircle their prey during the hunt. This behavior is modeled mathematically as follows [1]:

$$D = \left|\mathrm{C}.\mathrm{X_p} - \mathrm{X}\right| \tag{1}$$

Where X, $\mathrm{X_p}$ and D designate respectively the current wolf, the positions of the prey and the distance between them. The top three wolves (alpha, beta, and delta) update their positions using:

$$\mathrm{X(t)} = \mathrm{X_p(t)} - \mathrm{A}.\mathrm{D} \tag{2}$$

Where, vectors A and C provide direction for the wolves' movements [3]. They ensure that the wolves don't all move in the same direction, promoting exploration and diversification of the search space. They are determined by the following equations:

$$A = 2a.r_1 - a \tag{3}$$

$$C = 2.r_2 \tag{4}$$

$$a = 2\left(1 - \frac{t}{T}\right) \tag{5}$$

Here, r_1 and r_2 are two random vectors. The vector a decreases linearly from 2 to 0 as the iterations progress. In this context, t represents the current iteration and T is the total number of iterations.

The position of each omega wolf (X) is updated based on the leaders' positions (alpha, beta, and delta) using the following equation:

$$X(t+1) = \frac{X_1+X_2+X_3}{3} \tag{6}$$

Where

$$X_1 = X_\alpha - A_1.(D_\alpha) \qquad X_2 = X_\beta - A_2.(D_\beta) \qquad X_3 = X_\delta - A_3.(D_\delta) \tag{7}$$

And

$$D_\alpha = |C_1.X_\alpha - X| \quad D_\beta = |C_2.X_\beta - X| \quad D_\delta = |C_3.X_\delta - X| \tag{8}$$

Omega wolves are limited in their ability to explore the search space independently because they always follow the leaders. This restricted exploration increases the risk of the algorithm getting stuck in local optima [4].

3 Chaotic Maps

Chaotic systems are deterministic, nonlinear systems that exhibit complex, pseudo-random behavior despite following simple rules. Their key properties, such as sensitivity to initial conditions, ergodicity (exploring the entire state space), and non-periodicity, make them suitable for generating diverse initial populations for meta-heuristic algorithms. Using chaotic maps can potentially improve the algorithm's global search capability and prevent premature convergence. The values generated by chaotic maps are typically in the range [0,1], which are then scaled to the specific problem's bounds [6].

3.1 Logistic Map

The Logistic map is one of the simplest and most well-known chaotic systems. Its equation is:

$$x_{k+1} = \mu x_k (1 - x_k) \tag{9}$$

For chaotic behavior, the parameter μ is typically set to 4, and the initial value x_0 in (0,1)

3.2 Tent Map

The Tent map is another simple piecewise linear chaotic map defined by:

$$x_{k+1} = \begin{cases} 2x_k & 0 \le x_k < 0.5 \\ 2(1 - x_k) & 0.5 \le x_k \le 1 \end{cases} \tag{10}$$

The Tent map also generates values in the range [0,1]. It exhibits uniform distribution properties over its range, which is beneficial for population initialization [6].

3.3 Sine Map

The Sine map is a continuous chaotic map given by:

$$x_{k+1} = sin\ (\pi x_k) \tag{11}$$

Similar to the Logistic and Tent maps, the Sine map produces values in [0,1] when x_k in [0,1]. It also offers good ergodicity for generating diverse initial values.

4 Feature selection as a binary optimization problem

The problem of feature selection is typically formulated as a binary optimization problem [12]. The main step in this process is to determine the subset of features to incorporate in the representation of the solution (see Fig.1). Each candidate solution (wolf position) is a vector of length d equal to the number of features. The value 1 indicates that the corresponding feature is selected, while the value 0 signifies the non selection of the feature.

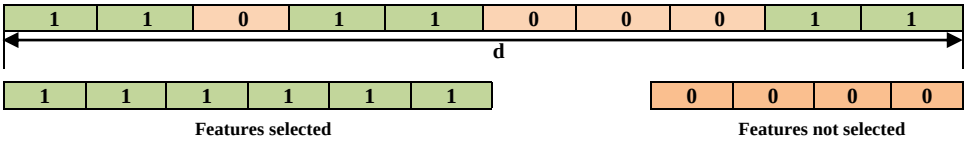

Fig.1. Representation of a solution in features selection

Since GWO operates in a continuous search space, a binarization strategy is required. A common approach is to use a threshold function [12]:

$$bin\_feat_j = \begin{cases} 1 & if\ x_j \geq 0.5 \\ 0 & if\ x_j < 0.5 \end{cases} \tag{12}$$

Where, x_j is the continuous value of the j^{th} dimension of a wolf's position, and bin_feat_j is the corresponding binary selection. This ensures that the continuous search space of GWO is effectively translated into a discrete feature selection problem.

5 Random Forest Classifier

Random Forest is a machine learning algorithm that leverages multiple decision trees to enhance predictive accuracy [5]. Each individual tree is trained on different, randomly selected subsets of the data. For classification tasks, the final prediction is determined by a majority vote among the trees' outputs, while for regression tasks, it's an average of their predictions. This ensemble approach effectively boosts accuracy and minimizes errors. Random Forest offers several key advantages that contribute to

its widespread use and effectiveness. RF is known by its strong robustness to overfitting by aggregating predictions from numerous decision trees reducing variance and leading to more generalized and reliable models [4]. Also RF is highly capable to handle high-dimensional datasets. It performs well even when dealing with a large number of features, making it suitable for complex real-world problems. Furthermore, the algorithm demonstrates insensitivity to feature scaling, which means there's no need to preprocess features by scaling them to a particular range.

6 Proposed Methodology: GWO-CFI for Feature Selection

The performance of any meta-heuristic algorithm, including GWO, is significantly influenced by its initial population. Traditional random initialization can sometimes suffer from:

- Non-uniform distribution: Random numbers might cluster in certain regions, leaving other parts of the search space unexplored.
- Lack of ergodicity: Purely random sequences might not cover the entire search space effectively over a limited number of initializations.

A good initial population should ideally possess two key characteristics:

1. Diversity (Exploration): The initial solutions should be spread out across the entire search space. This increases the chances of starting near the global optimum and helps prevent the algorithm from premature trapping in local optima.
2. Proximity to Optimum: Despite not always guaranteed, a diverse initialization might also include solutions that are already relatively good, potentially speeding up convergence.

The proposed methodology integrates chaotic initialization into the Grey Wolf Optimizer for feature selection. The overall flowchart is depicted in Fig.2.

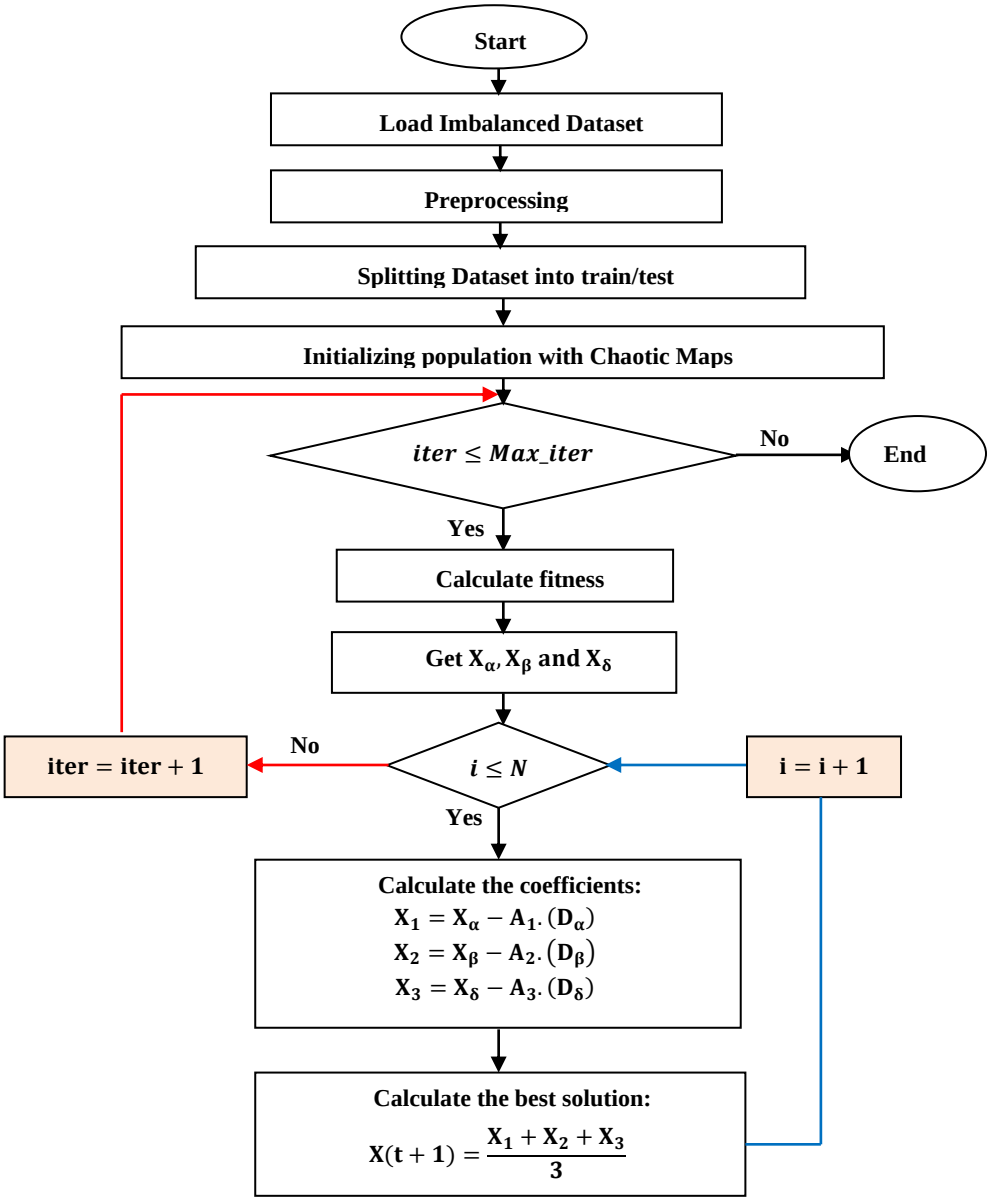

Fig.2. Flowchart of the proposed GWO-CFI feature selection approach

6.1 Data Preprocessing

After cleaning data from null values and duplicates, the following steps are ensured:

1. Encoding: The "Diabetes Prediction Dataset" contains both numerical and categorical features (numerical like BMI and categorical like 'gender' and

'smoking_history'). Categorical features like 'gender' are converted into numerical format creating new binary features.

2. Feature Scaling: Numerical features are scaled to a common range which is [0,1]. This helps prevent features with larger numerical ranges from dominating the distance calculations in optimization algorithms, although Random Forest itself is less sensitive to scaling.
3. Stratified Data Splitting: The dataset is split into training and testing sets. This ensures that the class distribution in both training and test sets is maintained, which is crucial for imbalanced datasets.

6.2 Fitness Function for Imbalanced Feature Selection

The fitness function guides the GWO in its search for the optimal feature subset. A multi-objective fitness function is commonly used, balancing predictive performance and the number of selected features. We aim to minimize this fitness function.

The fitness function f(S) for a binary feature subset S is defined as:

$$f(S) = \alpha \times \left(1 - F1\_score_{\text{Weighted}}\right) + (1 - \alpha) \times \frac{Number\ of\ selected\ features}{Total\ number\ of\ features} \quad (12)$$

The weighted F1-score obtained by the Random Forest classifier trained on the selected features and evaluated on the test set. Weighted F1-score is chosen because it inherently accounts for class imbalance by weighting metrics by the number of true instances for each label, providing a more reliable performance indicator than raw accuracy for such scenarios. Maximizing this value corresponds to minimizing (1 - F1-score). α is a weighting parameter, typically set close to 1 (e.g., 0.99), to emphasize the importance of classification performance over feature reduction. A higher alpha prioritizes a better F1-score, while a lower alpha puts more emphasis on reducing the number of features.

6.3 Evaluation Metrics

For the final evaluation of the selected feature subsets, especially on datasets with natural class imbalance, multiple metrics are reported:

1. F1-score (weighted): Harmonic mean of precision and recall, weighted by class support. More robust than accuracy for imbalance.
2. Precision (weighted): The proportion of positive identifications that were actually correct.
3. Recall (weighted): The proportion of actual positives that were correctly identified.
4. Number of Selected Features: Indicates the parsimony of the solution.

7 Experiments

The present work utilizes the "Diabetes Prediction Dataset" from Kaggle (https://www.kaggle.com/datasets/dat00700/diabetes-prediction-dataset).

7.1 Dataset

This dataset contains 100,000 samples and 8 features, plus 'diabetes' as binary target variable (0 or 1). The dataset contains the following features:

- gender (categorical: Female, Male, Other)
- age (numerical)
- hypertension (binary: 0, 1)
- heart_disease (binary: 0, 1)
- smoking_history (categorical: No Info, never, ever, current, former, not current)
- bmi (numerical)
- HbA1c_level (numerical)
- blood_glucose_level (numerical)
- **Target**: diabetes (binary: 0 for no diabetes, 1 for diabetes).

The dataset exhibits significant class imbalance (Figure 3). Analysis shows a high majority of individuals don't have diabetes (class 0) compared to those having diabetes (class 1) (approximately 91% Class 0 and 9% Class 1). This imbalance is a characteristic of the dataset that is directly reflected in the training and testing sets, without any explicit balancing techniques applied.

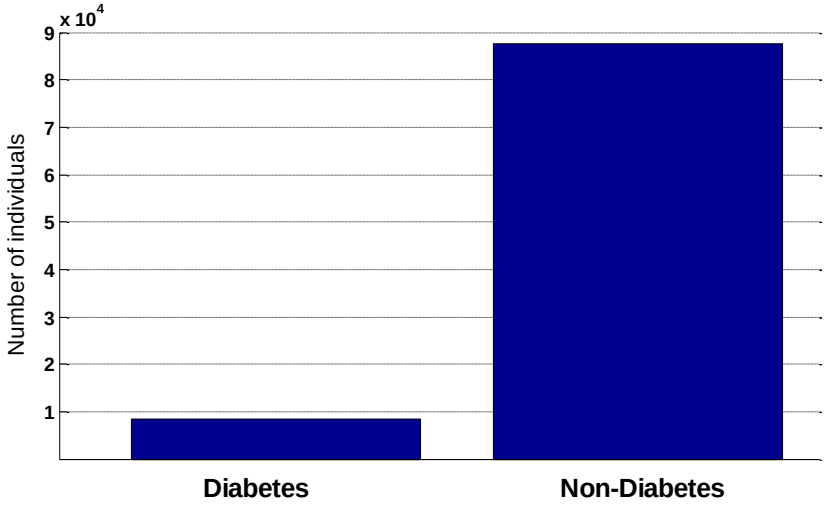

Fig.3. Diabetes vs Non-Diabetes counts

7.2 Preprocessing Steps

1. Cleaning: We performed data cleaning to remove null values and duplicate entries from the dataset. After this process, the dataset was refined to 96,164 records.
2. Encoding: 'gender' and 'smoking_history' columns are one-hot encoded to avoid multicollinearity. This expands the original 8 features into a larger set of 15 features.
3. Scaling: All numerical features (including newly created one-hot encoded features) are scaled to the [0,1] range.
4. Train-Test Split: The dataset is split into 70% training and 30% testing sets.

7.3 Algorithm Parameters

After loading and preprocessing the dataset, four different scenarios for GWO initialization are evaluated:

- GWO with Standard Random Initialization (GWO-SRI) (for baseline comparison)
- GWO with Logistic Map Initialization (GWO-LMI)
- GWO with Tent Map Initialization (GWO-TMI)
- GWO with Sine Map Initialization (GWO-SMI)

Due to its robustness and good generalization capabilities, Random Forest is a suitable choice as the evaluation model in our feature selection approaches.

The Grey Wolf Optimizer's parameters used are:

- Population Size: 30 wolves
- Maximum Iterations: 50

The Chaotic Maps' parameters considered are:

- Logistic Map, Tent Map, Sine Map are used for initialization, each with their standard parameters and initial seeds chosen randomly in (0,1) avoiding problematic points for Logistic map.

For the Random Forest Classifier, the number of trees is 100 and the class_weight is None (Default behavior, no explicit weight adjustment for imbalance). For the fitness Function, the chosen α is set to 0.99 (high weight on F1-score performance)

8 Results and Discussion

The experiments were conducted to evaluate the impact of different chaotic initialization strategies on the performance of GWO for feature selection on the diabetes prediction dataset, which naturally exhibits class imbalance. No explicit data balancing techniques were applied; the Random Forest classifier was trained directly on the imbalanced training data.

The following figure gives the final fitness achieved by GWO-LMI, GWO-TMI, GWO-SMI, and original GWO with random initialization after 50 iterations (see Fig.4).

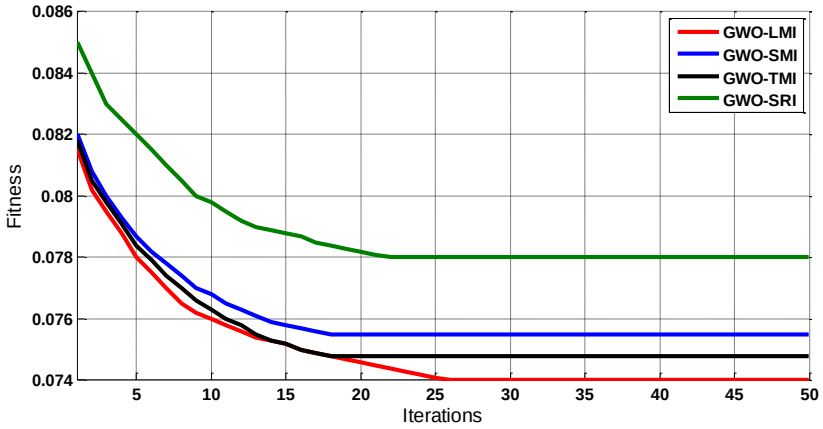

Fig.4. Fitness evolution for the four scenarios

Due to the stochastic nature of meta-heuristics, slight variations might occur across runs; however, general trends in performance are usually consistent. The displayed iteration output shows the progress of the best fitness and selected features throughout the optimization process.

This iterative output demonstrates how the GWO algorithm progressively refines the feature subset and improves the fitness. Over iterations, the number of selected features might fluctuate, but the algorithm generally aims for a smaller, more effective set.

Table 1 summarizes the performance of the four GWO versions, differentiated by their initialization methods, across the chosen metrics: Weighted F1-score, Precision, and Recall.

Table 1. Performance comparison with basic GWO

Initialization Method	Best Fitness	Selected Features	F1-Score	Precision	Recall
Logistic Map	0.0740	5	0.9360	0.9370	0.9360
Tent Map	0.0755	6	0.9245	0.9250	0.9245
Sine Map	0.0748	6	0.9252	0.9258	0.9252
Standard Random	0.0780	7	0.9220	0.9225	0.9220

From the results, all the three chaotic initialization methods used in experimentations (Logistic, Tent, Sine Maps) consistently outperformed the standard random initialization in terms of the achieved best fitness (lower fitness) and generally resulted in better classification performance metrics (higher F1-score, Precision, and Recall). This suggests that the initial diversity and ergodic exploration provided by chaotic maps effectively guide the GWO algorithm towards better regions of the search space from the outset.

In this specific experiment, the Logistic Map-initialized GWO yielded the best overall performance, achieving the lowest fitness and highest F1-score, along with a more compact feature subset of 5 features (age, hypertension, bmi, HbA1c_level and blood_glucose_level). This might be attributed to the specific chaotic properties of the Logistic map being well-suited for the problem landscape of this dataset. The reduced number of features also lowers computational costs during model training and inference.

The overall F1-score and related metrics (Precision, Recall) are still robust, despite no explicit data balancing techniques being applied. The use of weighted F1-score in the fitness function and for final reporting is crucial here, as it inherently accounts for the class imbalance by weighting metrics by the number of true instances for each label. This provides a more accurate picture of performance across both classes than raw accuracy alone, confirming that even without direct re-sampling, the selected features can lead to models that perform reasonably well on both majority and minority classes.

8.1 Best features selected using GWO based chaotic initialization

In the end, the repeatedly selected features across the best subsets from chaotic initializations often include:

- age
- hypertension
- bmi
- HbA1c_level
- blood_glucose_level These features are clinically well-known strong indicators for diabetes, which adds confidence to the feature selection process. 'heart_disease' and specific smoking_history categories also appear, demonstrating their relevance.

9 Conclusion

This article proposed and implemented an enhanced Grey Wolf Optimizer for feature selection, integrating chaotic maps (Logistic, Tent and Sine) for robust population initialization. This approach is applied to the diabetes prediction problem without explicit data resampling. The GWO-CFI approach demonstrated superior performance over standard random initialization. Chaotic initialization led to better fitness values, improved classification metrics (F1-score), and often resulted in more concise and clinically relevant feature subsets. This highlights the effectiveness of using chaotic dynamics to improve the initial exploration capabilities of meta-heuristic algorithms, even when working directly with imbalanced data. The selected features provide valuable insights for diabetes prediction, paving the way for more accurate and interpretable diagnostic models.

The algorithms require further validation across a more diverse range of datasets, specifically those exhibiting higher dimensionality, to ascertain their generalizability beyond the context of the 'Diabetes Prediction Dataset.' Future work will also investi-

gate the integration of explicit imbalance handling methodologies to bolster performance on skewed data distributions.

References

1. Chawla, V. K., Chanda, A. K., & Angra, S. (2019). The scheduling of automatic guided vehicles for the workload balancing and travel time minimization in the flexible manufacturing system by the nature-inspired algorithm. Journal of Project Management, 4(3), 19-30.
2. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Advances in Engineering Software 69, 46–61 (2014).
3.
4. Nadimi-Shahraki, M.H., Taghian, S., Mirjalili, S.: An improved grey wolf optimizer for solving engineering problems. Expert Systems with Applications 113917 (2020).
5. Salman, H.A., Kalakech, A., Steiti, A.: Random Forest Algorithm Overview. Babylonian Journal of Machine Learning 2024, 69–79 (2024).
6. Smith, J., Johnson, A.B., Williams, C.D.: Chaotic Wind-Driven Optimization with Hyperbolic Tangent Model and T-Distributed Mutation Strategy. In: Proceedings of the 2023 International Conference on Intelligent Systems (ICIS), pp. 56–62. IEEE (2023).
7. Zhu, D., Yang, Z., Chen, G.: An improved Grey Wolf Optimization based heuristic initialization algorithm for feature selection in P2P lending default prediction. Taylor & Francis Group Research Repository (2025).
8. Zou, R., Luo, J., Yu, W., Wei, R.: An effective feature selection approach based on hybrid Grey Wolf Optimizer and Genetic Algorithm for hyperspectral image. Scientific Reports 15(1), 1968 (2025).
9. Kumari, P., Singh, A.: A novel hybrid Grey Wolf Optimization algorithm using two-phase crossover approach for feature selection and classification. Computación y Sistemas 25(4), 793–808 (2021).
10. Zhao, Q., Sun, T., Yu, Z., Li, J., Cui, X., Zhou, Y.: Adaptive mechanism-based grey wolf optimizer for feature selection in high-dimensional classification. PLOS ONE 18(12), e0295843 (2023).
11. Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., Johannes, R.S.: Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In: Proceedings of the Symposium on Computer Applications and Medical Care, pp. 261–265. IEEE Computer Society Press (1988).
12. Khaseeb, J. Y., Keshk, A., & Youssef, A. (2025). Improved Binary Grey Wolf Optimization Approaches for Feature Selection Optimization. *Applied Sciences*, *15*(2), 489.