DOI: 10.14421/ijid.2026.4888

IJID (International Journal on Informatics for Development), *e-ISSN: 2549-7448*
Vol. 15, No. 1, June 2026, Pp.672-682

Comparative Analysis of The Combination of Metaheuristic and Machine Learning Algorithms

Sirmayanti Sirmayanti*
Department of Electrical Engineering
Politeknik Negeri Ujung Pandang
Makassar, Indonesia
*sirmayanti.sirmayanti@poliupg.ac.id

Farhan Rahman
Department of Electrical Engineering
Politeknik Negeri Ujung Pandang
Makassar, Indonesia
farhanrahman@poliupg.ac.id

Pulung Hendro Prastyo
Department of Informatics and Computer Engineering
Politeknik Negeri Ujung Pandang
Makassar, Indonesia
pulung.hendro@poliupg.ac.id

Mahyati
Department of Chemical Engineering
Politeknik Negeri Ujung Pandang
Makassar, Indonesia
mahyatikimia@poliupg.ac.id

Article History
Received November 28th, 2025
Revised December 30th, 2025
Accepted July17th, 2025
Published January, 2026

Abstract— **Diabetes affects about 1.9% of the global population, mainly through Type 2 diabetes. Machine learning (ML) serves a pivotal role in enhancing diabetes prediction by analyzing complex datasets. Feature selection, a crucial ML pre-processing step, improved prediction accuracy by identifying relevant data and discarding irrelevant features. This study investigates the combination of metaheuristic algorithms and ML techniques to enhance diabetes prediction accuracy and computational efficiency. Utilizing the PIMA, Early Stage, and Vanderbilt datasets, experiments evaluated ten algorithm-model combinations based on metrics like accuracy, precision, the Wilcoxon test, and convergence curves. Key findings included that Firefly Algorithm-Logistic Regression, Bat Algorithm-Logistic Regression, and Cuckoo Search-Logistic Regression achieved 74.72% accuracy on PIMA; Firefly Algorithm-Support Vector Machine and Cuckoo Search-Naïve Bayes achieved 83.39% accuracy and 96.15% precision on Early Stage; and Firefly Algorithm-Naïve Bayes achieved 92.88% accuracy and precision on Vanderbilt. These results highlighted the potential of integrating metaheuristics with ML methods to improve clinical diagnostics. Future research is recommended to validate algorithm robustness across diverse datasets to further optimize diabetes prediction strategies.**

***Keywords*—complex_dataset; diabetes_prediction; disease_detection; feature_selection; prediction_accuracy**

1 INTRODUCTION

Diabetes is a condition where blood sugar levels are not controlled in the body, leading to increased blood glucose levels, often referred to as hyperglycemia [1]. Diabetes is a severe health issue that affects many people globally, including in Indonesia. The International Diabetes Federation (IDF) reports that diabetes prevalence in the world is 1.9%, with Type 2 diabetes being the most prevalent, accounting for 95% of the global population. In Indonesia, Type 1 diabetes cases reached 41,8 million in 2022, making it the country with the highest prevalence of Type 1 diabetes in ASEAN and the third-highest prevalence among 204 countries globally [2]. In an era of rapid technological development, by 2023, various human jobs will be more accessible. According to Michael Chui in [3], there are many emerging technology trends, such as next-generation software engineering and the application of Artificial Intelligence in various industries, including the medical sector. Health services are an essential pillar of a healthy society, so applying AI and computational methods in healthcare systems is necessary to create healthier societies and reduce the risk of disease in future generations. It will improve the quality of life and introduce the concept of "telemedicine" [4].

Currently, healthcare professionals conduct medical examinations to predict the risk of developing diabetes. The data collected from the evolving diagnostic technology is beneficial for the diagnosis and treatment of diseases. However, doctors often find it difficult to quickly organize and analyze this data. Therefore, ML is increasingly used in the medical field to help doctors predict diseases and determine the outcome of their treatment [5]. ML is a technique that imitates human behaviour by learning from data and is highly effective in completing certain tasks [6]. Classification and prediction are two of the many tasks that ML can do in problem-solving. Classification sets the same pattern for a given class or target. ML classification is also used to predict disease through data that serves as a predictor and target to determine whether a person suffers from diabetes [7].

Feature selection is one of the classification pre-processing techniques commonly used in ML and Statistics to improve learning performance and solve problems with high-dimensional data. In high-dimensional data, feature selection is employed to select a relevant subset of features and remove redundant, excessive, and undesirable features prior to developing a classification model. [8]. Redundant features or attributes are removed because they do not contribute well as predictors to the learning model. After all, the information they provide has been presented or represented by other features. [9]. In addition, irrelevant features negatively affect the accuracy of the classification results and add to the difficulty of finding useful information in the data. [10].

Feature selection is divided into three methods: Filter, Wrapper, and Embedded. Filter methods are simple techniques that do not rely on ML algorithms and always focus on data characteristics. [11]. In contrast to the Filter method, the Wrapper method combines metaheuristic algorithms with ML algorithms to obtain the best features and gives better results than the Filter method. [11]. This approach uses a modeling algorithm that generates and evaluates each subset. The generated subset in Wrapper techniques is derived from various search algorithms. [12]. Meanwhile, the Embedded method efficiently selects features and performs well in the training process [5]. The hierarchy of feature selection algorithms can be seen in Fig. 1.

Research into the prediction of diabetes using a combination of metaheuristic algorithms is widely used and continuously developed to contribute to methods to detect diabetes quickly and achieve better performance efficiency. Herlambang et. al. in [13] obtained an accuracy performance of 74.67% using the pure XGBoost model ML. Subsequently, Astuti et al. [14] used the BWOA algorithm in combination with K-Nearest Neighbor, Naïve Bayes, Random Forest, Logistic Regression, Decision Tree, and Neural Network and achieved the best accuracy on BWOA-NB of 76%. Furthermore, Shankar et al. in [15] conducted experiments with the Ant Colony Optimization algorithm, obtaining an accuracy of 71%, and compared with the Grey Wolf Optimization algorithm with an accuracy of 81.15%.

Various studies have utilized several algorithmic approaches in combining metaheuristic and machine learning algorithms. Although these studies highlight the potential of such combinations in improving classification performance, they often focus on a single pairing of metaheuristic and machine learning algorithms without conducting a comparative analysis across multiple combinations. Moreover, limited research has been conducted in the context of diabetes prediction using a comprehensive evaluation of different metaheuristic-based feature selection methods integrated with various classifiers. Therefore, this study aims to fill this gap by performing a comparative analysis of different metaheuristic and machine learning algorithm combinations to identify the most effective pairing for diabetes disease classification.

2 METHOD

The flowchart delineates a feature selection process employing metaheuristic algorithms, shown in Fig. 2. The general phase of the flowchart is dataset acquisition (PIMA, Early Stage, and Vanderbilt), followed by dataset preprocessing, and then proceeds to feature selection with ten metaheuristic algorithms. The next phase after the feature selection is ML modeling which uses five ML algorithms. The last is model evaluation.

2.1 Dataset

This study uses three diabetes datasets with binary labels (diabetes or non-diabetes) for classification. Based on Table 1, the first dataset is the Pima Indians Diabetes Dataset (PIMA or PIDD), which is publicly available through the UCI Machine Learning Repository. It comprises 768 records and 8 numerical attributes related to diagnostic measurements of Pima Indian women aged 21 and older. The second dataset is the Early-stage Diabetes Risk Prediction Dataset, obtained from Kaggle, which contains 520 samples and 16 attributes capturing various symptoms and risk factors such as polyuria, polydipsia, and sudden weight loss. The third dataset is initially developed by the Biostatistics Program at Vanderbilt

University. The data was gathered through a survey involving several hundred rural African-American patients, focusing on various diagnostic parameters.

The three datasets are open and public datasets and are often used in research experiments. The inclusion of these three datasets ensures that the proposed approach is tested on data with varying characteristics, enhancing the generalizability and robustness of the study findings. Each dataset was split into training, validation, and testing sets in the following proportions: 80% training, and 20% testing.

2.2 *Dataset Preprocessing*

Data pre-processing is a crucial step to optimize model performance. Fig. 3 shows the process of pre-processing data in this research.

Data pre-processing begins with data cleaning for the PIMA, Early Stage, and Vanderbilt datasets. At this stage, outlier handling is carried out to ensure the absence of outlier data and that the data is normally distributed using the Interquartile Range (IQR). Furthermore, in the dataset, missing value handling is also carried out by filling the value using the Median if the data is not evenly distributed (there is skewness) and using the Mean when the data is normally distributed. The last cleaning stage is to perform data encoding and transformation of string data to numeric. After performing the data cleaning process, the resulting data features are then moved on to the feature selection stage using the wrapper method. Ten metaheuristic algorithms are used for feature selection, and the KNN algorithm aims to evaluate the performance of the selected feature subset in terms of predictive ability, which is then used to inform the objective function that aims to select the optimal features. After selecting the best features, they are then substituted into the ML algorithm as a combination of metaheuristics, and finally, the algorithm is evaluated for performance.

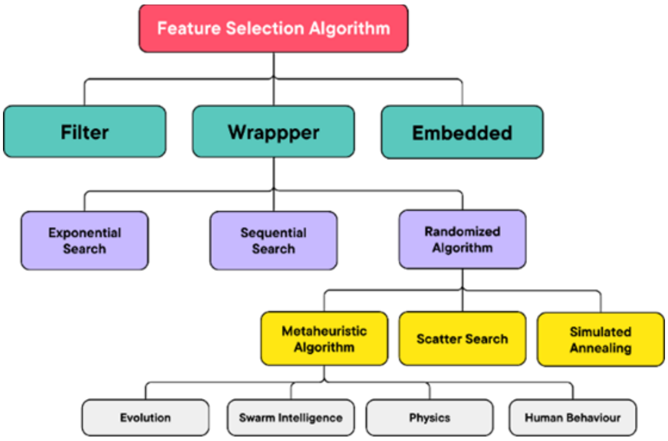


Figure 1. The hierarchy of feature selection algorithms [11]

Table 1. Dataset Description

Dataset	Feature	Instance	Source
PIMA	9	768	[16]
Early Stage	16	520	[17]
Vanderbilt	16	390	[18]

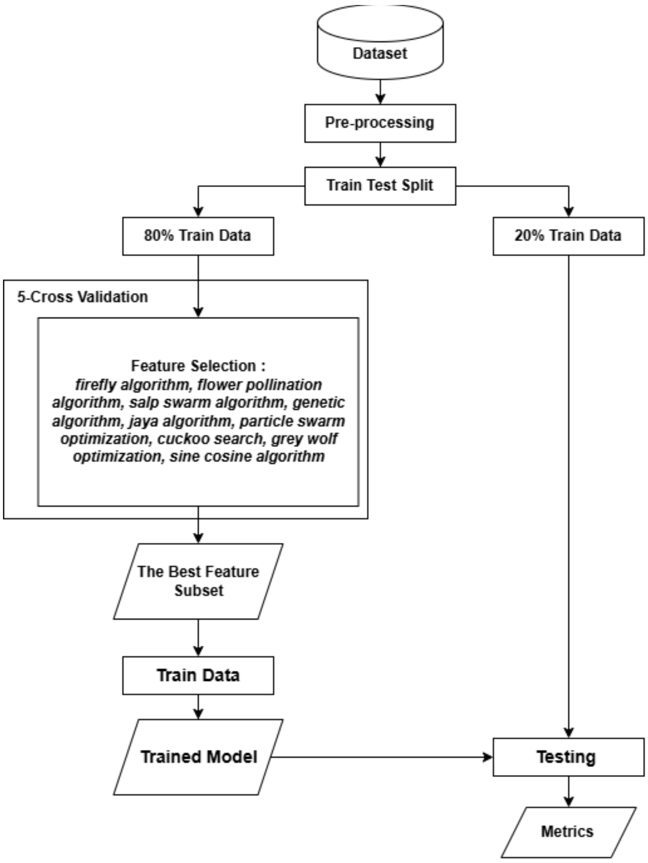

Dataset
Pre-processing
Train Test Split
80% Train Data
20% Train Data
5-Cross Validation
Feature Selection :
firefly algorithm, flower pollination algorithm, salp swarm algorithm, genetic algorithm, jaya algorithm, particle swarm optimization, cuckoo search, grey wolf optimization, sine cosine algorithm
The Best Feature Subset
Train Data
Trained Model
Testing
Metrics

Figure 2. Research flowchart

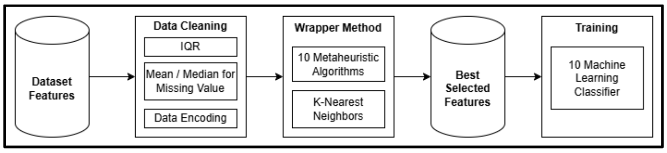


Figure 3. Data pre-processing phase

2.3 *Feature Selection Using Metaheuristic Algorithm*

Feature selection is a technique that identifies a subset of input features to improve or maintain classification accuracy. It can be categorized into filter and wrapper approaches. Filter methods rely on statistical, information theory, distance measurements, and intrinsic data characteristics to evaluate features. In contrast, wrapper methods evaluate the best combination of features by optimizing classification performance using a specific learning algorithm [19]. While filter methods are more general and do not involve a specific learning algorithm, wrapper methods can always achieve better classification results, making them a primary focus of research in feature selection. Metaheuristic algorithms are a type of wrapper method used for optimization and feature selection problems. They are derivative-free techniques that start by generating random solutions and do not require calculating the derivative of the search space, unlike gradient search techniques. These algorithms are characterized by their simplicity, flexibility, and ability to avoid local optima. Thus, this study focuses on wrapper feature selection.

This study employs ten different metaheuristic algorithms. These algorithms were selected based on their popularity, diversity, and proven effectiveness in solving high-dimensional optimization problems. Each algorithm offers a unique strategy inspired by biological, physical, or social processes, providing a broad spectrum of exploration and exploitation behaviours in the search space. Using a wide variety of algorithms enables a comprehensive evaluation of how different feature selection strategies affect classification performance in diabetes datasets, where the presence of irrelevant or redundant features can reduce model accuracy and interpretability.

After selecting the most relevant features using each metaheuristic algorithm, five classification algorithms, such as Support Vector Machine (SVM), k-Nearest Neighbors (k-NN), Random Forest (RF), Naïve Bayes (NB), and Decision Tree (DT), are used individually. Each classifier is applied to the feature set chosen by one metaheuristic algorithm. This approach ensures a clear and systematic assessment of how well each combination of feature selection and classification performs. The decision to evaluate classifiers separately rather than combining multiple feature selection techniques or classifiers was made to isolate the impact of each method. Using them together could obscure the contribution of individual algorithms and introduce complexity without clear interpretive benefits. This separation provides more precise insights into which metaheuristic–classifier pairings are most effective for diabetes classification across different datasets.

2.3.1 *Firefly Algorithm (FA): The standard FA algorithm* generates new solutions based on attractiveness, where fireflies move towards more attractive, brighter fireflies. This movement is determined by a randomization parameter α, which controls the randomness of the attraction. The movement is oriented toward the optimal solution, and the distance traveled is affected by the intensity of the flash-lighting [20].

2.3.2 *Flower Pollination Algorithm (FPA): The FPA is a* nature-inspired algorithm that simulates the fundamental pollination behaviour of flowers. Four rules are conceptualized in an idealized manner. The first rule of global pollination involves the interaction of living organisms and the transfer of pollen through cross-pollination, facilitated by a pollinating agent that follows a Lévy flight pattern. Rule 2 mandates the occurrence of abiotic and self-pollination to facilitate local pollination. Rule 3 states that the floral constant represents the likelihood of reproduction, which is directly related to the similarity between two flowers. Rule 4 pertains to the exchange probability, denoted as p, which ranges from 0 to 1. This probability can be influenced by external factors, such as wind, that affect the transfer of pollen between local and global populations. Indigenous pollination contributed significantly to the overall pollination activity [21].

2.3.3 *Salp Swarm Algorithm (SSA):* SSA is a randomized population-based algorithm that replicates the swarming behaviour of salps during their search for food in the water. Salps typically form a collective group called a salp chain in turbulent seas. In the SSA algorithm, the leader is the salp located at the front of the chain, while the remaining salps are referred to as followers. Similar to previous swarm-based methods, the location of salps is determined within an s-dimensional search space, where s represents the number of variables in a given problem [22].

2.3.4 *Jaya Algorithm (JA):* The Jaya optimization algorithm is utilized to address issues related to properly tuning algorithm-specific parameters, which are crucial for optimizing solutions and avoiding local optima. It is particularly valuable in designing an optimal subset of features to enhance classification performance [23].

2.3.5 *Genetic Algorithm (GA):* The GA derives inspiration from the process of biological evolution. Mutation and crossover are widely employed operators in genetic algorithms. Mutation and crossover are widely employed genetic algorithm operators. Mutation operates on an individual solution and often modifies a characteristic randomly or based on a predetermined criterion. Crossover, however, employs two parent solutions to generate two offspring, leading to the creation of novel and enhanced solutions. Typically, the mathematical model relies on an initial population of n individuals represented by chromosomes. Each iteration from a maximum number of t epochs consists of three operations: reproduction, mutation, and selection. The fittest individuals, as determined by the fitness function, are considered the solution to the given problem after the algorithm [24].

2.3.6 *Particle Swarm Optimization (PSO):* The PSO was initially introduced by Kennedy and Eberhart [25] as a method for addressing binary optimization challenges. In the PSO algorithm, the collective group of individuals is referred to as a swarm. This swarm consists of N particles that navigate across a search space with many dimensions. The particle symbolizes the prospective solution and traverses the search space to locate the optimal answer. Each particle autonomously seeks the global maximum or minimum based on its own accumulated experience and knowledge [26].

2.3.7 *Bat Algorithm (BA):* The BA was formulated by relying on inspiration from the echolocation behavior exhibited by bats. Within the context of BA, an artificial bat possesses vectors representing its position, velocity, and frequency. These vectors are continually updated over the iterations. The artificial bats navigate the search space by utilizing location and velocity vectors within the continuous real domain [27].

2.3.8 *Cuckoo Search (CS):* The CS algorithm is inspired by the obligate brood parasitism of the cuckoo, where it lays its eggs in the nests of smaller birds, such as starlings. Upon the hatching of the egg, the starling assumes the role of caretaker for the cuckoo chick, treating it as if it were its biological offspring. As the cuckoo chick grows larger than the other chicks in the nest, it dominates and displaces them, ultimately leading to their complete expulsion from the nest. Each individual in this population is represented by a "nest," and each "egg" in the nest symbolizes a potential solution. The term "cuckoo eggs" is used to denote new solutions that are introduced into the population. As suggested by Levy, the replacement of a better solution with a bad solution is proposed [28].

2.3.9 *Grey Wolf Optimization (GWO):* The GWO is influenced by the guidance and hunting behavior exhibited by packs of grey wolves. In every population of grey wolves, there exists a reciprocal hierarchy that determines the dominance and authority. The alpha wolf, who leads the entire pack, is the most influential in hunting, feeding, and migrating. The beta wolf, being the second most powerful, assumes leadership in the event of the alpha's death or illness. Alpha and beta have greater influence than omega and delta. The GWO algorithm draws its main inspiration from this particular form of social intelligence [29].

2.3.10 *Sine Cosine Algorithm (SCA):* During the initial phase of optimization utilizing the SCA in feature selection, the SCA will randomly choose various sets of features from the original feature set to produce population group feature subsets. Next, the evaluation function is used to score each feature subset, and the feature subset with the highest score is identified as the optimal feature subset. Subsequently, the feature subsets that have been initialized are disrupted to generate new feature subsets with a specified number of points. Furthermore, an assessment function is employed to evaluate the score of each newly generated subset of features. The subset with the highest score is then compared to the best subset of features from the previous round to determine the current optimal subset of features. This process is repeated to acquire the optimal subset of features following the maximum number of repetitions. An essential aspect of the SCA feature selection process is that the optimization approach employed by the SCA has an impact on the variety of feature subsets [29].

2.4 *Machine Learning Model*

This research focuses on five ML classifiers: KNN, SVM, NB, DT, and LR. These models are simple and non-parametric, making them adaptable to various data distributions. SVM offers high accuracy and robustness in high-dimensional spaces. NB is computationally efficient and provides a probabilistic framework, while DT is easy to interpret and captures non-linear relationships. LR, known for its simplicity and clear probabilistic outputs, is a strong baseline model. These algorithms balance complexity and usability, making them suitable for robust diabetes prediction and insights into risk factors [30] [31].

2.4.1 *K-Nearest Neighbors (KNN):* The KNN technique is a classification approach that assigns data objects to classes based on their shortest distance. The selection of the optimal *K* value for this algorithm relies on the analysis of the available data. A higher value of *K* can mitigate the impact of noise on classification, but it can also result in more indistinct boundaries between different classifications. This approach employs an appropriate distance metric to categorize novel data. The value of the nearest neighbor distance *K* is computed, and the anticipated class label of the nearest neighbor is assigned as the class label of the new instance.

2.4.2 *Support Vector Machine (SVM):* The origins of SVM can be traced back to statistical learning theory. As a classification task, it searches for the most effective decision boundary (hyperplane) that separates the instances of one class from another. The SVM is a fundamental type of supervised classifier that aims to maximize the margin to achieve optimal generalization. It effectively addresses the issues of overfitting and underfitting by utilizing different kernel functions that facilitate nonlinear separation [32].

2.4.3 *Naïve Bayes (NB):* The NB is a highly successful and efficient algorithm for inductive learning in ML and data mining. Despite relying on attribute independence, NB exhibits competitive solid performance in the classification process. The assumption of attribute independence in accurate data is uncommon. However, even if this assumption is violated, empirical investigations have shown that the performance of NB classification remains relatively high [33].

2.4.4 *Decision Tree (DT):* DT is a predictive model that utilizes a tree or hierarchical structure. The purpose of decision trees is to convert data into decision trees and decision rules. The primary advantage is the ability to decompose intricate decision-making processes into more manageable ones, hence facilitating the identification of solutions to current issues for decision-makers [34].

2.4.5 *Logistic Regression (LR):* LR is a type of supervised ML classifier that calculates real-valued features from the input data. It then multiplies each feature by a weight, adds them together, and applies a sigmoid function to the result to get a probability. A threshold is employed to determine a course of action [35].

2.5 Model Evaluation

The model evaluation is conducted after the integration of ML algorithms. The system utilizing the classification procedure is anticipated to classify all of the data accurately. The measures employed for model validation in this study are accuracy and precision. Accuracy refers to correctly identified instances of the total number of cases, whereas precision refers to the proportion of cases with positive outcomes. In this study, the dataset was split into 80% training and 20% testing using stratified sampling to maintain class distribution. Additionally, we employed k-fold cross-validation (k=5) to validate model performance across different splits, avoid overfitting and to ensure robust evaluation across cross-validation, computing accuracy, F1-score, and precision for each model.

In this analysis, the ML models use the following hyperparameter which were determined based on prior research and grid search tuning.

- KNN: n_neighbors = 12
- SVM: kernel=’rbf’
- NB: var_smoothing = 1e-09
- DT: criterion=’entropy’, max_depth=3, max_leaf_nodes=5, min_samples_split =2
- LR: C=0.0001, penalty=’L2;

Table 2. Metaheuristic Algorithm’s Parameter

Algorithms	Hyperparameter Values

FA	ub = 1
	lb = 0
	thres = 0.5
	alpha = 1
	beta0 = 1
	gamma = 1
	theta = 0.97

FPA	ub = 1
	lb = 0
	thres = 0.5
	beta = 1.5
	P = 0.8

SSA	ub = 1
	lb = 0
	thres = 0.5

JA	ub = 1
	lb = 0
	thres = 0.5

GA	ub = 1
	lb = 0
	thres = 0.5
	CR = 0.8 # crossover rate
	MR = 0.01 # mutation rate

PSO	ub = 1
	lb = 0
	thres = 0.5
	w = 0.9 # inertia weight
	c1 = 2 # acceleration factor
	c2 = 2 # acceleration factor

BA	ub = 1
	lb = 0
	thres = 0.5
	fmax = 2 # maximum frequency
	fmin = 0 # minimum frequency
	alpha = 0.9 # constant
	gamma = 0.9 # constant
	A_max = 2 # maximum loudness
	r0_max = 1 # maximum pulse rate

CS	ub = 1
	lb = 0
	thres = 0.5
	Pa = 0.25 # discovery rate
	alpha = 1 # constant
	beta = 1.5 # levy component

GWO	ub = 1
	lb = 0
	thres = 0.5

SCA	ub = 1
	lb = 0
	thres = 0.5
	alpha = 2 # constant

Feature selection is a crucial step in improving model performance by identifying the most relevant features while reducing dimensionality. In this study, we employed various metaheuristic algorithms to optimize the selection process. The effectiveness of these algorithms depends on properly tuning their hyperparameters. Table 2 presents the hyperparameters used for various metaheuristic algorithms applied in feature selection experiments. These algorithms are designed to optimize the selection of relevant features by exploring the search space efficiently. Each algorithm operates within predefined upper (ub = 1) and lower (lb = 0) bounds, ensuring that the selected features are appropriately constrained. Additionally, most algorithms utilize a threshold (thres = 0.5) to determine feature inclusion. To evaluate the significance, this study utilized the Wilcoxon signed-rank test data validation method and the Metaheuristic Algorithm learning curve.

In this work, the fitness value is utilized as the validation parameter for the metaheuristic algorithm. An optimal fitness value is characterized by its capacity to optimize algorithm performance while minimizing the number of features rather than solely relying on accuracy as a performance metric. The metaheuristic algorithm will undergo thirty iterations to evaluate its performance during the feature selection stage. Subsequently, the algorithm will be evaluated based on the average fitness value, convergence curve, and Wilcoxon score. A lower fitness value indicates superior algorithm performance. An exemplary method is characterized by rapid convergence and the ability to avoid becoming trapped in the optimal position. The convergence curve illustrates this. Furthermore, the Wilcoxon value can be employed with a significance level (α) of 0.05 to determine the statistical significance of performance comparisons among various metaheuristic algorithms. A Wilcoxon *p*-value greater than

0.05 suggests that there is no statistically significant difference between a specific algorithm and the other combination of algorithms.

3 RESULT AND DISCUSSION

3.1 Accuracy and Precision

PIMA, Early Stage, and Vanderbilt were the three main datasets used to train a combination model using Metaheuristic and ML techniques. Table 3 shows the average of 30 accuracy model combinations, and Table 4 shows the average of 30 precision model combinations. The best performance of average accuracy and precision is written in bold. While using the PIMA dataset, the FA-LR, BA-LR, and CS-LR combination gets the highest average accuracy of 74.71% compared to FPA, which only gets 73.34%, and SSA-LR with an accuracy of 71.42. Then, the average accuracy value of the JA algorithm in combination with LR is 74.39%, which is the highest value among the other five ML algorithm combinations. Furthermore, GA-NB data gets an average accuracy of 74.06%, the best accuracy of the Genetic Algorithm. Next, PSO gets the best average accuracy of 74.69% with a combination of ML Naïve Bayes Classifier algorithms. The BA implementation gets the average result of the BA-LR algorithm with an accuracy of 74.71%, the same as CS-LR. In the implementation of the GWO algorithm combination, the highest average accuracy obtained is GWO-LR with an average of 74.01%. Finally, SCA-NB became the algorithm combination model with the best average accuracy of 74.38%.

From Fig. 4, the SSA-SVM combination has the lowest value among the other ten algorithm combinations, with a value of 66.87%. In addition, the precision value by CS-SVM and SCA-SVM is the same precision value, which is 70.94%. Then, the top three algorithm combinations are BA-LR with an average precision value of 74.71%, followed by FPA-SVM with a value of 73.34%, and FA-DT with a value of 72.57%. Since the BA-LR combination has the highest average precision value on the PIMA dataset, it is the recommended algorithm.

The comparison of the best-performing algorithm using the Early-Stage dataset is illustrated in Fig. 5. The graph shows that the FA-SVM combination achieves an average accuracy of 83.39%, outperforming other algorithm combinations within the FA category. The highest average precision is also attained by the FA-SVM algorithm, achieving 96.15%, which is superior to the FA-NB category. Next, the FPA achieves the best average accuracy with the FPA-LR combination, at 74.91%, whereas the highest average precision is observed in the FPA-DT combination, at 83.29%. In the SSA category, the SSA-SVM combination demonstrates the best average accuracy of 74.48%, while the SSA-NB combination achieves an average precision of 85.01%. In the JA, both the average accuracy and precision peaks are found in the JA-SVM combination, with values of 82.90% and 95.68%, respectively. The GA performs best, achieving an average accuracy of 81.71% in the GA-SVM combination, with the highest average precision of 95.68%. Within the Particle Swarm Optimization framework, PSO-NB achieves the highest average accuracy at 80.07%.

Simultaneously, it also secures the best average precision of 92.37%. For the Bat Algorithm, the combination BA-LR achieves an accuracy of 78.07% and a precision of 92.54%. Cuckoo Search achieves an accuracy of 83.39% in the CS-NB combination and a precision of 96.15%. Grey Wolf Optimization attains its best average accuracy and precision in the GWO-SVM combination, recording values of 82.62% and 92.87%, respectively. Lastly, the Sine Cosine Algorithm achieves the highest average accuracy of 81.92% in the SCA-NB combination and an average precision of 95.62% in the SCA-SVM combination.

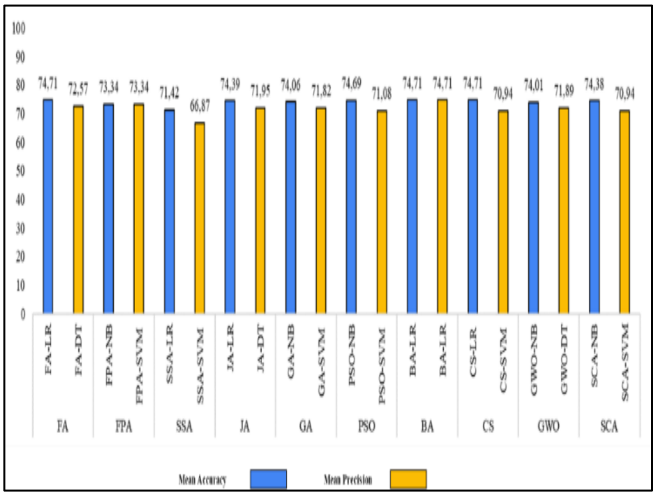


Figure 4. Best combination performance based on average accuracy and Average precision on the PIMA dataset

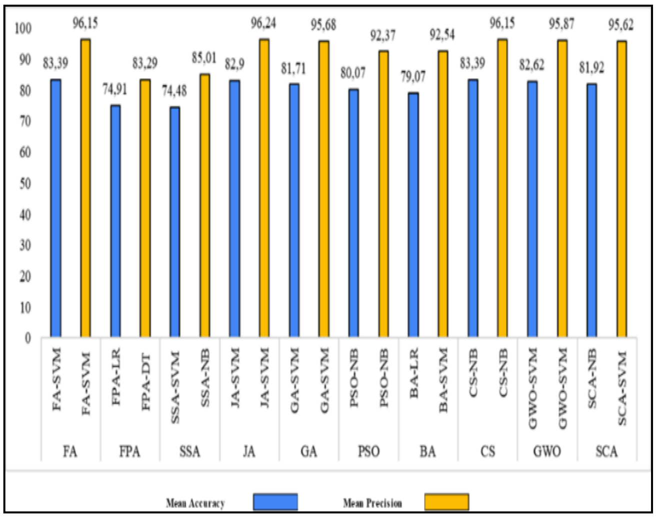


Figure 5. Best combination performance based on average accuracy and average precision on the Early-Stage dataset

Fig. 6 also presents the best of the average accuracy and average precision results using the Vanderbilt dataset. The FA-KNN combination achieves the highest average accuracy at 92.64%, while FA-NB attains an average precision of 94.12%. In comparison, for the Flower Pollination Algorithm, the FPA-SVM combination secures the best average accuracy of 87.37% and an average precision of 87.79%. Similarly, the SSA obtains its highest accuracy in the SSA-LR combination at 87.79% and achieves an average precision of 88.59% in the SSA-NB combination. In the context of the JA, the best average accuracy and precision are found in the JA-LR combination, yielding 92.14% accuracy and 93.10% precision.

Furthermore, the GA delivers the best average accuracy of 90.94% in the GA-NB combination, with its corresponding best average precision of 91.89% to PSO, the preferred algorithm is PSO-SVM with an average accuracy of 90% and an average precision of 90.87%. Conversely, within the BA framework, BA-DT achieves the highest accuracy result of 89.94%, while its average precision on the algorithm is 91.46%. Contrasting with Cuckoo Search, the CS-NB combination performs best, with an average accuracy of 92.12% and a precision of 93.22%. Moreover, Grey Wolf Optimization demonstrates an algorithm achieving the highest accuracy of 90.96% in the GWO-LR combination, followed by the best precision of 91.86% in the GWO-KNN combination. Lastly, the Sine Cosine Algorithm performs best, with an accuracy of 90.50% in the SCA-LR combination and an average precision of 91.89% in the SCA-NB combination.

Table 5 presents a comparative analysis of various machine learning models applied to the PIMA, Early Stage, and Vanderbilt datasets, demonstrating that our proposed models consistently outperform conventional methods in both accuracy and precision. For the PIMA dataset, prior studies using K-Means, Decision Tree (DT), PCA + LR, and KNN achieved accuracy rates between 67.00% and 73.00%, while our proposed FA-LR, BA-LR, and CS-LR models surpass them with an accuracy of 74.71% and a competitive precision of 70.75%. In the Early-Stage dataset, traditional models such as Multilayer Perceptron (MLP), Deep Neural Networks (DNN), Logistic Regression (LR), and KNN attained accuracy values ranging from 64.40% to 80.00%, whereas our proposed FA-SVM model significantly outperforms them with an outstanding accuracy of 83.39% and an exceptional precision of 96.15%.

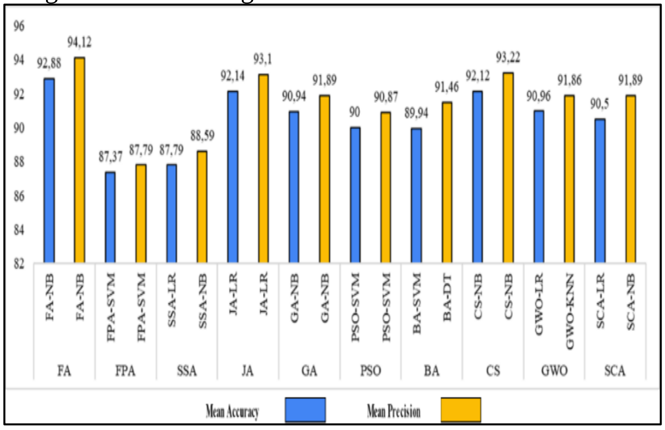


Figure 6. Best combination performance based on average accuracy and average precision on the Vanderbilt dataset

Similarly, for the Vanderbilt dataset, existing approaches, including Random Forest, LR, Stochastic Gradient Descent (SGD), and Gradient Boosting, achieved accuracy rates between 71.00% and 91.50%, while our proposed FA-NB model demonstrates the highest performance with an impressive accuracy of 92.88% and a remarkable precision of 94.14%. These results highlight the clear advantage of our metaheuristic-based feature selection techniques, proving their superior effectiveness in enhancing predictive performance across diverse datasets and complex classification tasks.

Table 6 shows a comparison of different studies that used various methods and validation techniques to predict diabetes using the PIMA, Early-Stage, and Vanderbilt datasets. It includes the method used in each study, the validation approach, and the accuracy achieved. For the PIMA dataset, past studies applied algorithms like Firefly-Bat Neural Networks, Ant Colony Optimization, and Bees Algorithm, with accuracy results ranging from 70% to 73%. In this study, the combination of FA–LR and Cuckoo Search and Logistic Regression (CS–LR) achieved a higher accuracy of 74.71% using 10-fold cross-validation. Next, for the Early-Stage dataset, previous research tested SVM, Multilayer Perceptron, and Naïve Bayes, with accuracies between 64.40% and 76.60%. The method proposed in this study, Firefly Algorithm with SVM (FA–SVM) and Cuckoo Search with Naïve Bayes (CS–NB), improved the accuracy to 83.92%, also using 10-fold cross-validation. On the Vanderbilt dataset, other studies reached up to 92.5% accuracy using traditional methods like Logistic Regression and Decision Trees. However, the current study’s approach using the Firefly Algorithm with Naïve Bayes (FA–SVM) achieved the highest accuracy of 92.98% with 10-fold cross-validation.

Table 3 Classification Accuracy (%) of Ten Metaheuristic Algorithms of Five Classifiers for Three Datasets

Dataset	ML Algorithm	Metaheuristic Algorithm									
		FA	FPA	SSA	JA	GA	PSO	BA	CS	GWO	SCA
PIMA	KNN	72.826	71.181	70.186	71.636	71.980	72.339	72.339	72.320	71.967	72.320
	SVM	74.392	73.349	71.567	73.372	73.902	73.588	74.160	73.489	73.097	73.489
	NB	74.696	73.173	70.108	73.893	74.062	74.696	74.696	74.019	74.019	74.380
	DT	72.904	71.432	70.277	72.727	72.140	72.123	72.904	72.121	72.675	72.121
	LR	**74.718**	72.562	71.424	74.393	73.997	73.688	**74.718**	**74.718**	73.616	73.917
Early-Stage	KNN	82.464	71.303	72.875	82.140	79.666	75.732	76.408	81.594	81.630	81.578
	SVM	**83.392**	73.774	74.483	82.905	81.712	77.816	78.120	82.297	82.624	81.686
	NB	83.274	71.058	72.486	81.633	78.705	80.075	78.800	**83.392**	79.382	81.924
	DT	82.496	73.163	72.401	82.676	79.601	75.287	77.535	82.058	81.950	80.431
	LR	82.535	74.911	72.284	81.526	80.415	73.601	79.078	81.964	80.748	81.359
Vanderbilt	KNN	89.636	85.286	84.123	89.166	88.717	88.149	88.004	90.038	89.029	87.867
	SVM	92.649	87.376	86.064	90.102	90.465	90	89.940	91.910	90.551	90.153
	NB	**92.888**	86.876	87.470	91.675	90.948	89.764	89.376	92.128	90.149	90.940
	DT	87.025	82.363	78.465	86.235	86.290	84.294	89.406	86.085	85.807	85.893
	LR	92.696	86.897	87.799	92.149	90.282	89.388	89.406	92.085	90.965	90.500

Table 4. Classification Precision (%) of Ten Metaheuristic Algorithms by Five Classifiers for Three Datasets

Dataset	ML Algorithm	Metaheuristic Algorithm									
		FA	FPA	SSA	JA	GA	PSO	BA	CS	GWO	SCA
PIMA	KNN	68.377	65.117	62.371	66.364	67.173	68.019	68.019	67.964	67.121	67.964
	SVM	72.399	70.580	66.870	70.545	71.827	71.083	72.399	70.948	68.335	70.948
	NB	70.338	67.787	61.128	68.834	69.104	70.338	70.338	68.834	68.834	69.677
	DT	**72.577**	68.133	62.052	71.959	69.979	70.093	**72.577**	69.200	71.892	69.2
	LR	70.759	63.634	50.192	69.941	68.870	64.258	70.759	70.759	67.886	68.556
Early-Stage	KNN	95.116	81.455	82.222	94.980	92.517	88.359	88.600	94.478	94.791	94.579
	SVM	96.155	82.980	84.392	**96.242**	95.685	90.504	92.543	95.921	95,879	95,629
	NB	96.153	83.202	85.013	93.876	92.357	92.379	92.208	96.155	92,802	95,084
	DT	95.867	83.290	79.513	95.936	92.415	87.326	89.726	95.923	95.817	93.966
	LR	95.016	82.650	77.729	94.598	93.519	83.669	91.526	95.436	94.212	93.875
Vanderbilt	KNN	92.491	87.833	86.839	92.081	91.383	90.859	90.640	93.041	91.869	90.625
	SVM	93.943	87.796	86.308	90.959	91.449	90.879	90.779	93,070	91.472	91.104
	NB	**94.124**	87.598	88.592	92.850	91.898	90.707	90.228	93.221	90.955	91.893
	DT	92.292	87.525	86.858	91.423	91.164	89.505	91.465	91.318	90.737	90.355
	LR	93.745	87.388	88.392	93.105	91.055	90.127	90.022	93.063	91.745	91.275

Table 5. Comparative Similar Study Showing Accuracy and Precision

Dataset	Study	Method	Accuracy	Precision
PIMA	[36]	K-Means	67.00%	N/A
	[37]	DT	71.35%	N/A
	[38]	PCA + LR	72.70%	64.30%
	[39]	KNN	73.00%	87.00%

	This Study	**Proposed FA-LR, BA-LR, and CS-LR**	**74.71%**	**70.75%**
Early Stage	[40]	Multilayer Perceptron	64.40%	64.40%
	[41]	Deep Neural Network	78.00%	N/A
	[42]	LR	78.00%	N/A
	[43]	KNN	80.00%	81.20%
	This Study	**Proposed FA-SVM**	**83.39%**	**96.15%**
Vanderbilt	[44]	Random Forest	71.00%	78.00%
	[45]	LR	89.00%	N/A
	[46]	Stochastic Gradient Descent	90.68%	N/A
	[47]	Gradient Boosting	91.50%	N/A
	This Study	**Proposed FA-NB**	**92.88%**	**94.14%**

Table 6. Comparative Study of the Method Used and the Validation Method

Dataset	Study	Method	Validation Method	Accuracy
PIMA	[48]	Hybrid Firefly-Bat Optimized Fuzzy Artificial Neural Network	10-Fold Cross Validation	70%
	[15]	Ant Colony Optimization (ACO)	K-Fold Cross Validation	71%

	[49]	Baseline Algorithm DT, SVM dan AdaBoost	Train_test_split 70%: 30%	DT = 70.80%
				SVM = 66.50%
				AB = 71.00%

	[50]	Multi-Objective Bees Algorithm	10-Fold Cross Validation	72%
	[51]	Binary Wheal Optimization Algorithm (BWOA)-KNearest Neighbors	K-Fold Cross Validation	73%
	This Study	**Firefly Algorithm-Logistic Regression (FA-LR) and Cuckoo Search-Logistic Regression (CS-LR)**	**10-Fold Cross Validation**	**74.7186%**
Early-Stage	[52]	SVM	K-Fold Cross Validation	66.56%
	[40]	Multilayer Perceptron	10-Fold Cross Validation	64.40%
	[53]	Naïve Bayes	Cross-Validation	76.60%
	[54]	SVM	Train_test_split 60: 40%	62%
	This Study	**Firefly Algorithm-Support Vector Machine Classifier (FA-SVM)** **Cuckoo Search-Naïve Bayes (CS-NB)**	**10-Fold Cross Validation**	**83.392%**
Vanderbilt	[55]	Univariate Feature Selection LR	10-Fold Cross Validation	88.89%
	[56]	Adaptive Synthetic Sampling-KNN (ADASYN-AIIKNN)	10-Fold Cross Validation	89.80%
	[57]	Genetic Algorithm- Decision Tree	Train Test Split 80:20	90.06%
	[58]	Quadratic Discriminant Analysis	10-Fold Cross Validation	85.91%
	[59]	Logistic Regression	Train Test Split 70%: 30%	92.5%
	This Study	**Firefly Algorithm-Naïve Bayes (FA-NB)**	10-Fold Cross Validation	**92.88%**

3.2 *Best Fitness and Wilcoxon Test*

The data in Table 7 are performance results measured by best fitness and Wilcoxon Signed-Rank test values to test statistical significance (α). This test uses the p-value, which is above 0.05, meaning that the algorithm combination has no significance, and the model performance is not much different from other models. This study proposes the FA-LR algorithm on the PIMA CS-NB dataset on the Early-Stage dataset, and FA-NB on the Vanderbilt dataset, which is marked as **"-"** to calculate statistical significance because it has the highest accuracy.

In using the PIMA dataset, the best fitness obtained from the experimental results is 0.12641, which is the smallest value in fitness value acquisition. The model evaluation metric is to use the Wilcoxon Signed-Rank test to test statistical significance (α). This test uses the p-value, which is above 0.05, meaning that the algorithm combination has no significance, and the model performance is not much different from other models. The algorithm combinations that are the reducing factor in the Wilcoxon test are the FA-LR, BA-LR, and CS-LR algorithms. Algorithms that do not have statistical significance are shown in FA-NB, FPA-LR, JA-NB, JA-LR, GA-LR, PSO-NB, PSO-LR, BA-NB, CS-NB, GWO-NB, GWO-LR, SCA-NB, and SCA-LR.

In the Early-Stage dataset, the optimal fitness value of 0.06344 is achieved by utilizing different algorithm combinations. Specifically, the FA yields the best results with the FA-SVM, FA-NB, and FA-KNN combinations. The FPA performs well with the FPA-SVM, FPA-DT, and FPA-LR combinations. The SSA produces favorable outcomes with the SSA-DT combination. The JA demonstrates good performance with the JA-NB combination. Lastly, the PSO algorithm achieves satisfactory results with the PSO-NB combination. The combinations of BA-KNN, BA-SVM, and BA-DT, along with the BA, yielded the highest performance.

Additionally, the combination GWO-SVM, using the GWO algorithm, and the combination SCA-NB, using the SCA, also achieved excellent results. Then, in the Wilcoxon Test results, the FA-NB, FA-DT, and FA-LR combinations have no significance against FA-SVM or CS-NB in the FA category. In the FPA category, only the FPA-LR combination is of statistical importance with a value of 0.0002624.

Furthermore, in SSA, SSA-KNN and SSA-VM are algorithms with statistical significance. In contrast to JA, only the JA-NB and JA-DT combinations do not have any significance. Similarly, other algorithm combinations that do not have significance against FA-SVM and CS-NB algorithms include PSO-KNN and PSO-SVM. While utilizing the Vanderbilt dataset, the minimum iteration in the combination of the FA, JA, and SCA offers the optimal fitness point of 0.0672. On the other hand, the SSA-DT combination obtains the highest fitness value, which equals 0.1335. According to the Wilcoxon test, the Firefly Algorithm demonstrates statistical significance in the FA-SVM and FA-LR combinations. The FPA-NB and FPA-LR combinations have insignificant values of 0.297 and 0.3660, respectively. The SSA demonstrates relevance solely in the SSA-KNN combination, with a precise value of 0.00665. The Jaya Algorithm yielded noteworthy combinations of JA-SVM and JA-NB compared to JA-LR.

The combination of GA-KNN and GA-NB demonstrates a notable advantage over GA-SVM in the context of GA. The PSO-KNN combination presents the lowest level of importance compared to PSO-SVM in PSO. The BA had a notable impact on the combination of BA-DT. The CS demonstrates notable improvements in CS-KNN and CS-LR compared to CS-NB. In the Wilcoxon Test value, FA only has a significant value in the combination of FA-SVM and FA-LR. Next, only FPA-SVM is the algorithm combination with a significance value of 0.000305. SSA has no algorithm that has significance.

Furthermore, JA obtained JA-NB, which became a significant algorithm against FA-NB. Next, GA obtained the GA- GA-NB combination. Unlike PSO, only the PSO-NB combination is significant. Similarly, BA with BA-NB is a significant algorithm. Then, CS obtained CS-SVM as a combination, which was significant against FA-NB. Then, in GWO, only the GWO-NB combination has a significant value. Finally, SCA only obtains SCA-NB with a combination of algorithms with significance. From the results of the analysis, it can be seen that the combination of metaheuristic algorithms using NB and SVM is statistically significant against FA-NB.

3.3 *Convergence Curve*

Table 8 demonstrates that each of the five combinations of the FA consistently attained the global optimum on the PIMA dataset, with a fitness value of 0.1264 being the most optimal. FA-LR, FA-SVM, and FA-NB rapidly reached convergence, while FA-NB had an initial value of 0.2512. Next, the JA and PSO exhibited parallel patterns, with JA-DT commencing at 0.2511, while JA-KNN, JA-SVM, JA-NB, and JA-LR attained the optimal fitness value of 0.0634. PSO-DT and PSO-NB commenced at 0.2500 and 0.2506, respectively, but PSO-KNN, PSO-SVM, and PSO-LR initiated at 0.1264. The GWO algorithm demonstrated that GWO-DT and GW-NB initially had values that were not optimal, whereas GWO-KNN, GWO-SVM, and GWO-LR quickly achieved the optimal values.

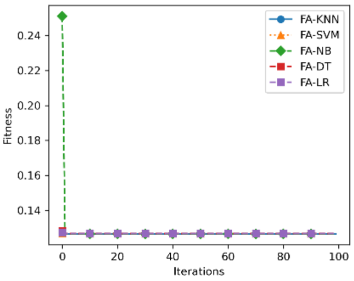
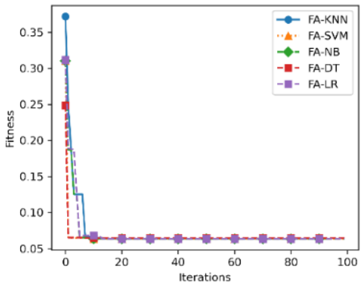
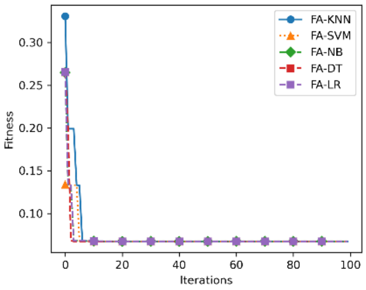
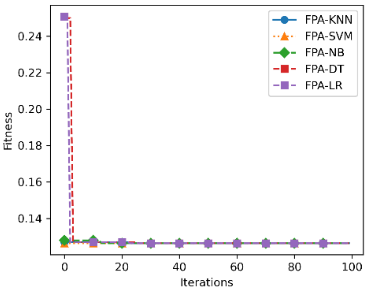
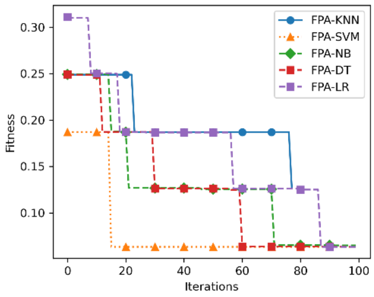
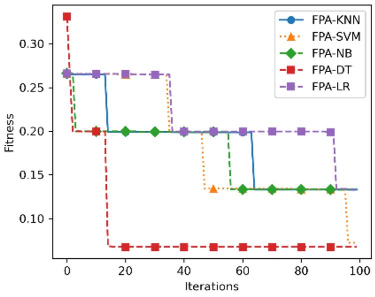
Besides, the SCA shows excellent performance, especially when used with SCA-SVM, SCA-LR, and SCA-DT. The combinations of BA-DT and BA-LR quickly and effectively reached the global optimum. The global optimum was swiftly achieved only by using the CS-LR. The SSA and GA showed initial differences, with SSA-KNN starting at 0.3735 and GA-LR at 0.3734. However, all combinations of the ten algorithms finally reached the global optimum of 0.1264 without premature convergence

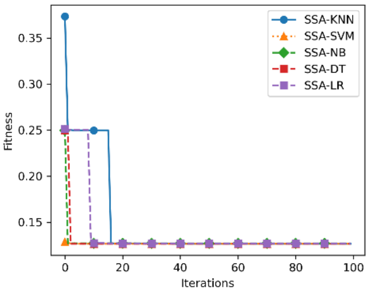
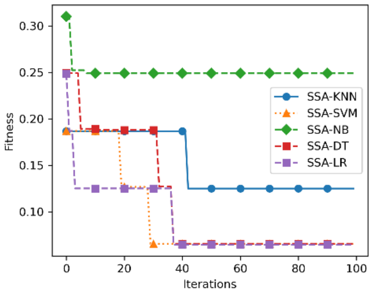
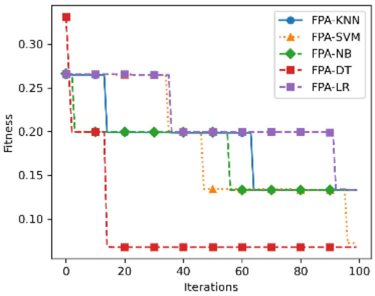
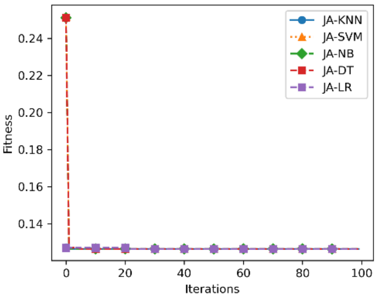
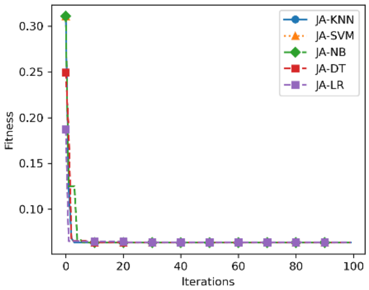
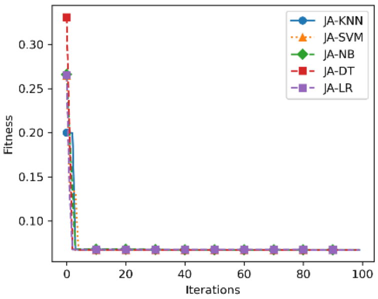
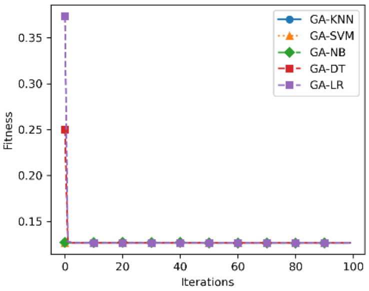
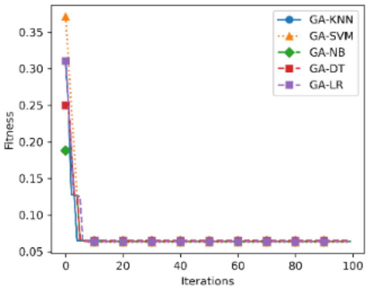
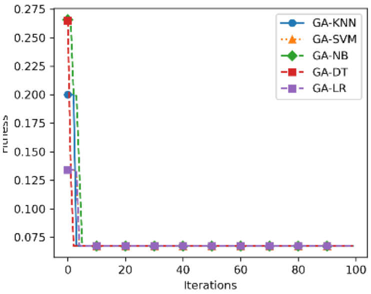
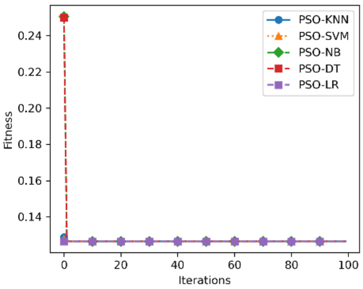
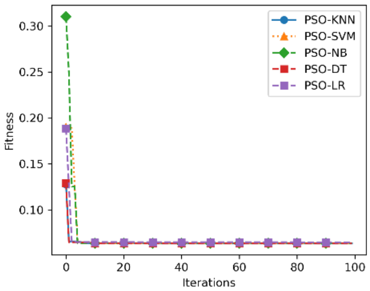
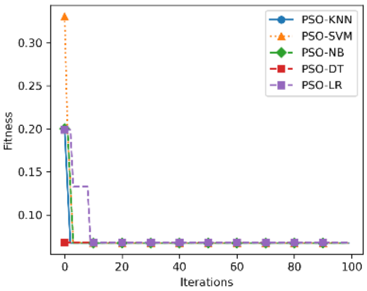
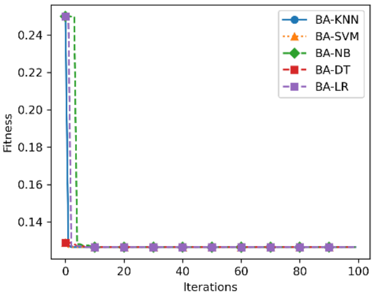
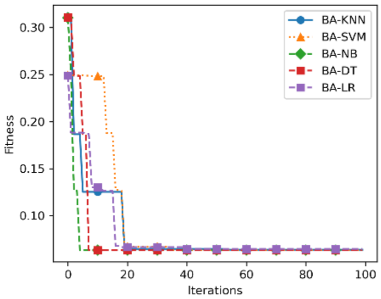
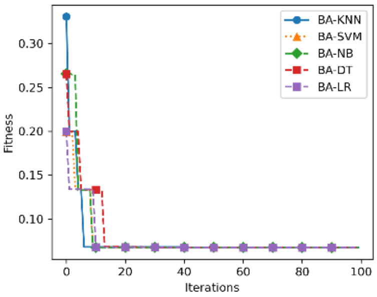
Table 7. Best Fitness and the Wilcoxon Test for Each Combined Algorithm

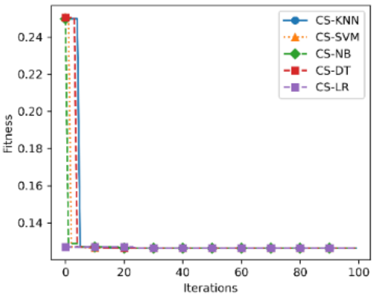
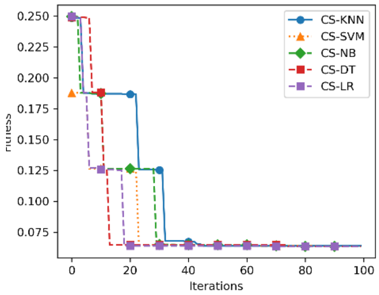
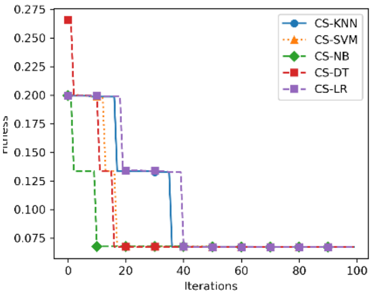
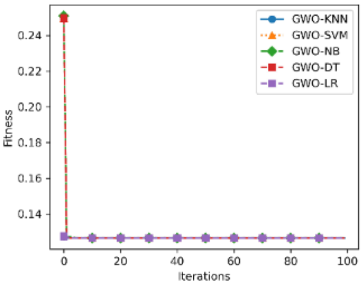
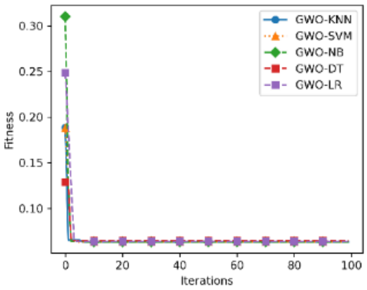
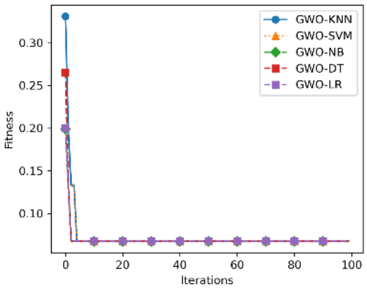
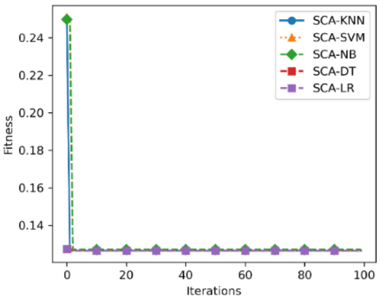
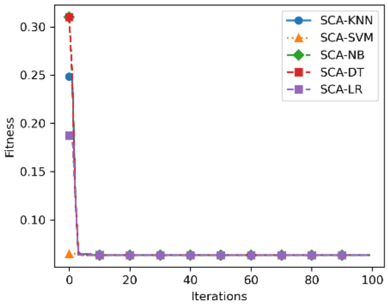
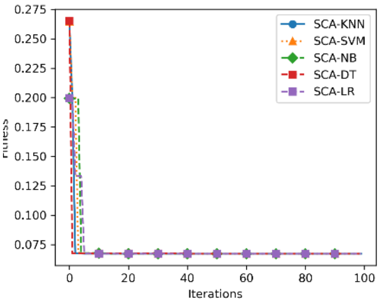
Combined Algorithm	Best Fitness			Wilcoxon Test		
	PIMA	Early Stage	Vanderbilt	PIMA	Early Stage	Vanderbilt
FA-KNN	**0.126412338**	**0.0634436**	**0.0672821**	1.86265E-09	0.003314453	1.86E+06
FA-SVM	0.126607143	**0.0634436**	**0.0672821**	3.85624E-06	-	0.000305511
FA-NB	**0.126412338**	**0.0634436**	**0.0672821**	0.309023385	0.317310508	-
FA-DT	0.126607143	0.0647181	**0.0672821**	1.86265E-09	0.067889155	1.86E+06
FA-LR	0.126607143	**0.0634436**	**0.0672821**	-	0.179712495	0.000112725
FPA-KNN	**0.126412338**	0,0665809	0.1330256	1.86E+06	2.91E+11	0.034536734
FPA-SVM	**0.126412338**	**0.0634436**	0.0721538	0.000152871	3.54E+08	-
FPA-NB	**0.126412338**	0.0650123	0.1328974	0.013208347	3.23E+10	0.297734257
FPA-DT	**0.126412338**	**0.0634436**	0.0675385	1.86E+06	1.23E+11	0.000441624
FPA-LR	**0.126412338**	**0.0634436**	0.1327692	0.670180589	0.00026245	0.366010269
SSA-KNN	**0.126412338**	0.1251225	0.1337949	1.86E+06	0.000345249	0.006650218
SSA-SVM	**0.126412338**	0.0656985	0.0679231	3.70E+08	0.000257131	0.492496412
SSA-NB	0.126996753	0.0638358	**0.0672821**	0.000141032	1.90E+11	-
SSA-DT	**0.126412338**	**0.0634436**	0.1335385	1.86E+06	2.62E+11	1.30E+07
SSA-LR	0.126996753	0.0647181	0.1330256	0.005033508	5.50E+10	0.715132967
JA-KNN	**0.126412338**	**0.0634436**	**0.0672821**	1.86E+06	0.00146885	1.42E+10
JA-SVM	**0.126412338**	**0.0634436**	**0.0672821**	3.87E+09	0.043114447	0.022600679
JA-NB	**0.126412338**	**0.0634436**	**0.0672821**	0.18204979	0.067889155	0.049282444

Combined Algorithm	Best Fitness			Wilcoxon Test		
	PIMA	Early Stage	Vanderbilt	PIMA	Early Stage	Vanderbilt
JA-DT	**0.126412338**	**0.0634436**	**0.0672821**	1.86E+06	0.10880943	1.86E+06
JA-LR	**0.126412338**	0.0638358	**0.0672821**	0.317310508	0.027707849	-
GA-KNN	**0.126412338**	0.0638358	**0.0672821**	1.86E+06	0.000293053	0.044907209
GA-SVM	**0.126412338**	**0.0634436**	**0.0672821**	1.30E+06	0.007685794	-
GA-NB	**0.126412338**	0.0638358	**0.0672821**	0.058997545	0,002217721	0.037783417
GA-DT	**0.126412338**	0.0649142	0.0675385	1.86E+06	0.005062032	1.60E+10
GA-LR	**0.126412338**	0.0638358	0.0675385	0.179712495	0.007685794	0.837257554
PSO-KNN	**0.126412338**	**0.0634436**	**0.0672821**	1.86E+06	2.70E+11	0.026630168
PSO-SVM	0.126996753	0.0638358	**0.0672821**	3.13E+10	7.99E+09	-
PSO-NB	**0.126412338**	**0.0634436**	**0.0672821**	0.309023385	0.007685794	0.852549787
PSO-DT	**0.126412338**	**0.0634436**	0.0680513	1.86E+06	0.000654958	1.60E+10
PSO-LR	**0.126412338**	0.0647181	0.0679231	0.10880943	0.000293053	0.779721214
BA-KNN	**0.126412338**	0.0638358	0.0675385	1.86E+06	0.00013142	0.253436105
BA-SVM	**0.126412338**	**0.0634436**	0.0675385	3.86E+09	0.000651666	0.87871595
BA-NB	**0.126412338**	**0.0634436**	**0.0672821**	0.309023385	0.003502272	-
BA-DT	**0.126412338**	**0.0634436**	**0.0672821**	1.86E+06	0.000978707	0.00171859
BA-LR	**0.126412338**	0.0647181	**0.0672821**	-	0.00146885	0.526770984
CS-KNN	0.126542208	0.0638358	**0.0672821**	1.86E+06	0.00220902	0.001232104
CS-SVM	0.126542208	0.0634436	**0.0672821**	3.86E+09	0.067889155	0.031864783
CS-NB	0.126542208	**0.0628365**	**0.0672821**	0.147378523	-	-
CS-DT	0.126542208	0.0634436	**0.0672821**	1.86E+06	0.011718686	3.86E+08
CS-LR	**0.126412338**	0.0634436	**0.0672821**	-	0.017960478	0.01718927
GWO-KNN	0.126542208	**0.0634436**	**0.0672821**	1.86E+06	0.000437777	0.109432058
GWO-SVM	0.126542208	**0.0634436**	**0.0672821**	3.86E+09	0.067889155	0.416649397
GWO-NB	0.126542208	**0.0634436**	**0.0672821**	0.147378523	0.002183045	0.892825524
GWO-DT	0.126542208	**0.0634436**	0.0675385	1.86E+06	0.011718686	3.86E+08
GWO-LR	0.126542208	0.0638358	0.0675385	0.10880943	0.002217721	-
SCA-KNN	0.126542208	**0.0634436**	**0.0672821**	1.86E+06	0.00146393	0.040489722
SCA-SVM	0.126542208	**0.0634436**	**0.0672821**	3.86E+09	0.007685794	-
SCA-NB	0.127191558	**0.0634436**	**0.0672821**	0.263139822	0.027707849	0.123902137
SCA-DT	0.126542208	0.0647181	**0.0672821**	1.86E+06	0.007685794	4.42E+09
SCA-LR	0.126542208	**0.0634436**	**0.0672821**	0.179712495	0.017960478	0.646895924

Table 8. Convergence Curves for Each Algorithm

Alg.	**Datasets**		
	PIMA	**Early Stage**	**Vanderbilt**
FA			
FPA			

Alg.	Datasets		
	PIMA	Early Stage	Vanderbilt
SSA			
JA			
GA			
PSO			
BA			

Alg.	**Datasets**		
	PIMA	**Early Stage**	**Vanderbilt**
CS			
GWO			
SCA			

In comparison with the Early-Stage dataset, which uses several exploration methods to locate the global best/optimum solution. The FA-KNN combination begins with the highest fitness value of 0.372, followed by FA-SVM, FA-NB, and FA-LR at 0.3105. FA-DT achieves the fastest search initiation with a time of 0.2486. All combinations effectively reach the global optimum at 0.0634 without being confined to local optima. Then, the FPA-LR combination initiates at a distance of 0.31113971 and takes three exploration steps: [0.25044, 0.1872, 0.1262] toward the minimum fitness value. Next, the FPA-NB, FPA-DT, and FPA-KNN models achieve a score of 0.250, while the FPA-SVM model starts exploring the fastest with a score of 0.1872.

The FPA-KNN algorithm is the only one that becomes stuck in a local minimum, prematurely converging at a value of 0.1262, and so fails to find the optimal solution. The convergence of the Salp Swarm Algorithm (SSA). The SSA-NB algorithm commences with a value of 0.3101 and becomes trapped in a local minimum at 0.252, which is also the starting point for the SSA-DT and SSA-LR algorithms. SSA-DT reaches a value of 0.189 at iteration 6 and a value of 0.1273 at iteration 3. On the other hand, SSA-LR reaches a value of 0.1273 at iteration 4. The SSA-KNN algorithm commences with an initial value of 0.187 and concludes with a final value of 0.1253. However, it fails to get the ideal solution due to premature convergence. SSA-SVM exhibits superior computational efficiency in identifying the optimal fitness, commencing from the same initial point as SSA-KNN. Only three permutations of the SSA algorithm attain the global optimum.

The JA commences the search initially with JA-NB at a value of 0.311, subsequently followed by JA-DT and JA-LR. The GA ranks GA-SVM as the highest with a value of 0.371, followed by GA-LR and GA-KNN. Among the PSO algorithms, PSO-DT demonstrates the highest speed in converging towards the global optimum, while PSO-NB exhibits the slowest convergence. The BA demonstrates that BA-LR has the highest speed, starting at 0.24887255, while BA-KNN has the slowest performance. The CS algorithm demonstrates that CS-NB, CS-LR, and CS-SVM are the most efficient, while CS-DT is the slowest, starting with a value of

0.249. On the other hand, GWO exhibits the GWO-DT variant with the highest initial speed of 0.1290, whereas the SCA identifies SCA-SVM as the fastest.

Lastly, the convergence curves of various metaheuristic algorithm combinations highlight their differing efficiencies in reaching global optima by using the Vanderbilt dataset. The FA combined with KNN shows the slowest start with a fitness value of 0.3308, while FA-SVM achieves the fastest global optimum search with a fitness value of 0.13366. Next, FPA-DT demonstrates the farthest search point but excels in deeper exploration, achieving a global best at 0.06753846. Then, the SSA shows the longest initial search distance with SSA-KNN and SSA-NB at 0.3308, with SSA-SVM being the fastest to reach the global best. Furthermore, the JA exhibits the farthest convergence process with JA-DT at 0.3310, followed by JA-SVM, JA-NB, JA-LR, and JA-KNN, all achieving global optimum. The PSO shows only PSO-NB with a fitness value of 0.3307 as the farthest point, while others like PSO-KNN, PSO-NB, and PSO-NB start at 0.2003, with PSO-DT achieving the fastest global best. Next, BA reveals BA-KNN as having the longest exploration process compared to BA-SVM and BA-LR, starting at 0.2001, but all combinations eventually achieve the global best. CS algorithm shows CS-KNN, CS-SVM, CS-NB, and CS-LR reaching global best fitness at 0.1992, the fastest among them, while CS-DT starts at 0.2656, with all combinations achieving global best. In the GWO algorithm, GWO-KNN initiates the farthest search at 0.331, yet successfully attains the global best.

Table 9. Comparison of Convergence Curves of the Top 5 Best Performing Algorithm

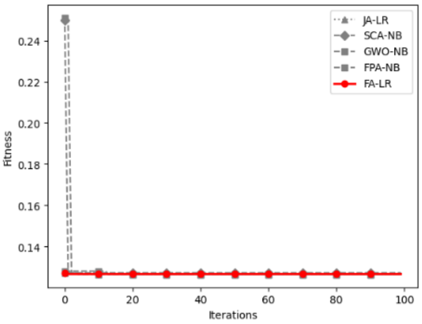
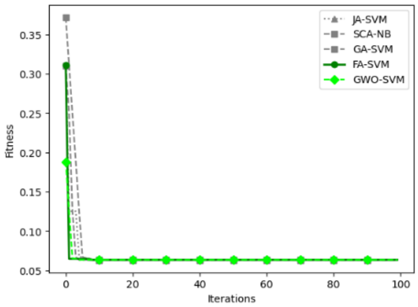
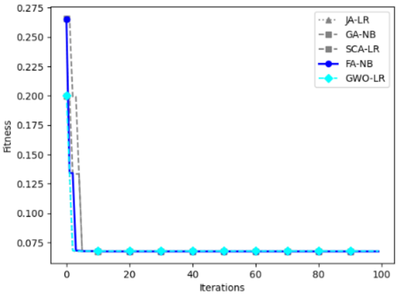
Dataset	Best-Performing Convergence Curve
PIMA	
Early Stage	
Vanderbilt	

Table 10. Important Features Selected for Each Algorithm

Alg.	**Dataset**		
	PIMA	**Early Stage**	**Vanderbilt**
FA	Glucose, Insulin, BMI	Polyuria, Polyphagia, Delayed Healing	Glucose, HDL Chol, BMI, Waist
FPA	Glucose, Insulin	Gender, Polyuria, Polyphagia Genital Thrush, Visual Blurring, Delayed Healing, Partial Paresis, Muscle Stiffness	Glucose, HDL Chol, Gender, Height, Weight, BMI, Systolic BP, Diastolic BP, Waist
SSA	Glucose, Blood Pressure, Insulin, Diabetes Pedigree Function, Age	Gender, Polyuria, Polydipsia, Sudden Weight Loss, Visual Blurring, Delayed Healing, Partial Paresis Muscle, Stiffness, Alopecia	Glucose, Chol / HDL ratio Age, Height, BMI, Gender Hip, Diastolic BP, Waist/ hip ratio, Systolic BP
JA	Glucose, Insulin	Polyuria, Polydipsia, Sudden Weight Loss	Glucose, Height
GA	Glucose, Insulin	Age, Polyuria, Polydipsia	Glucose, Height, Diastolic BP
PSO	Glucose, Blood Pressure, Age, Insulin	Polyuria, Sudden Weight Loss, Visual Blurring, Irritability, Muscle Stiffness	Glucose, BMI, Diastolic BP
BA	Glucose, Insulin	Polyuria, Sudden Weight Loss, Polyphagia	Glucose, HDL Chol, Height, Diastolic BP
CS	Glucose, Age	Age, Polyuria, Polyphagia	Glucose, Height, Diastolic BP
GWO	Glucose, Blood Pressure	Polyuria, Polydipsia	Glucose, HDL Chol, Waist
SCA	Glucose	Polyuria, Polydipsia	Glucose, Height, HDL, Chol Diastolic BP

In Table 9, which contains a comparison of the top 5 best-performing algorithms in the convergence curve, it can be seen that the proposed FA LR model (the same convergence is also in BA-LR and CS-LR) on the PIMA dataset can achieve the best fitness quickly. In contrast to the Early-Stage dataset, in the global solution search process, the proposed FA-SVM (which has the same performance as CS-NB) succeeds in reaching the best solution even though it starts from a more distant initial point compared to GWO-SVM. Finally, in the same case as Early Stage, using the Vanderbilt dataset, the proposed FA-NB succeeded in searching for the global best despite initializing farther from GWO-LR.

3.4 *Selected Features*

The feature selection techniques utilized by the metaheuristic algorithms consist of ten different methods, each possessing distinct characteristics in conducting search and feature selection. Table 10 presents the selected features identified by each metaheuristic algorithm for each data. The selected features, such as glucose levels, insulin levels, BMI, and blood pressure, were chosen based on their strong correlation with diabetes diagnosis and progression. For instance, glucose and insulin are fundamental indicators of diabetes, directly influencing how models differentiate between diabetic and non-diabetic patients. Similarly, symptoms like polyuria (excessive urination) and polydipsia (excessive thirst) are well-known early signs of diabetes, contributing to better sensitivity in ML models. Other physiological markers, including blood pressure and BMI, serve as indicators of metabolic disorders, helping the model detect patterns associated with diabetes-related complications. Beyond feature selection, it is important to empirically justify their impact on model performance.

4 CONCLUSION

This study has proposed a comprehensive comparison of combined Metaheuristic and ML for detecting diabetes disease. In conclusion, the BA-LR and CS-LR PIMA achieved the highest average accuracy of 74.71% in the PIMA dataset. The FA-SVM and CS-NB combination achieved the highest average accuracy of 83.39% on the Early-Stage dataset, with CS-NB having the highest precision of 96.15%. The SSA implementation suffered from premature convergence, causing SSA-KNN and SSA-NB to become trapped in local optima. The algorithm with the highest average accuracy and precision on the Vanderbilt dataset was 94.12% and 92.64%, respectively. The combination of FPA-KNN, SSA-KNN, and SSA-NB became stuck in local optima. Future research should explore the performance of the algorithms on different datasets to determine their generalizability and robustness. Additionally, combining metaheuristics and ML algorithms showed promising results, and further research could investigate more combinations and variations to find the optimal combination for different data.

CREDIT AUTHOR STATEMENT

Sirmayanti played a key role as the lead author responsible for the research, paper writing, overseeing the research process, ensuring its smooth execution and strict adherence to the established methodology. Farhan Rahman was responsible for developing the code and executing the experiments. Pulung Hendro Prastyo contributed to the creation of the literature review, helping to identify the novelty of the research. Finally, Mahyati assisted with the revision and improvement of the manuscript writing.

COMPETING INTERESTS

In accordance with the publication ethics of this journal, Sirmayanti, Farhan Rahman, Pulung Hendro Prastyo, and Mahyati, the authors of this article, hereby declare that there are no conflicts of interest (COI) or competing interests (CI) associated with this work.

DECLARATION OF GENERATIVE AI AND AI-ASSISTED TECHNOLOGIES IN THE WRITING PROCESS

Grammarly was utilized during the drafting process to refine grammar and style. Nonetheless, the authors affirm that the manuscript's central arguments, conclusions, and literature review were developed independently, without reliance on generative AI or automated content tools. All content has been thoroughly verified by the authors, who assume full responsibility for the final work.

REFERENCES

[1] L. Ryden, G. Ferrannini, and E. Standl, "Risk prediction in patients with diabetes: is SCORE 2D the perfect solution?," Jul. 21, 2023, *Oxford University Press*. doi: 10.1093/eurheartj/ehad263.

[2] A. Ahdiat, "Jumlah Penderita Diabetes Tipe 1 di ASEAN Berdasarkan Kelompok Usia (2022)," databoks. Accessed: Jul. 05, 2024. [Online]. Available: https://databoks.katadata.co.id/datapublish/2023/02/10/indonesia-punya-penderita-diabetes-tipe-1-terbanyak-di-asean

[3] McKinsey Digital, "Technology Trends Outlook 2023," 2023.

[4] P. Solanki, D. Baldaniya, D. Jogani, B. Chaudhary, M. Shah, and A. Kshirsagar, "Artificial intelligence: New age of transformation in petroleum upstream," Feb. 01, 2022, *KeAi Publishing Communications Ltd.* doi: 10.1016/j.ptlrs.2021.07.002.

[5] R. Liu, Y. Rong, and Z. Peng, "A review of medical artificial intelligence," Jun. 01, 2020, *KeAi Communications Co.* doi: 10.1016/j.glohj.2020.04.002.

[6] R. B. Lukmanto, Suharjito, A. Nugroho, and H. Akbar, "Early detection of diabetes mellitus using feature selection and fuzzy support vector machine," in *Procedia Computer Science*, Elsevier B.V., 2019, pp. 46–54. doi: 10.1016/j.procs.2019.08.140.

[7] A. A. S. P. R. A. S. Roihan, "Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper," 2019.

[8] O. O. P. A. O. H. F. S. Shilan Hamed, "Filter-Wrapper Combination and Embedded Feature Selection for Gene Expression Data," *Int. J. Advance Soft Compu*, 2021.

[9] F. Navazi, Y. Yuan, and N. Archer, "An examination of the hybrid meta-heuristic machine learning algorithms for early diagnosis of type II diabetes using big data feature selection," *Healthcare Analytics*, vol. 4, Dec. 2023, doi: 10.1016/j.health.2023.100227.

[10] C. Bielza and P. Larrañaga, *Data-Driven Computational Neuroscience*. Cambridge University Press, 2020. doi: 10.1017/9781108642989.

[11] P. Agrawal, H. F. Abutarboush, T. Ganesh, and A. W. Mohamed, "Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019)," *IEEE Access*, vol. 9, pp. 26766–26791, 2021, doi: 10.1109/ACCESS.2021.3056407.

[12] T. M. Le, T. M. Vo, T. N. Pham, and S. V. T. Dao, "A Novel Wrapper-Based Feature Selection for Early Diabetes Prediction Enhanced with a Metaheuristic," *IEEE Access*, vol. 9, pp. 7869–7884, 2021, doi: 10.1109/ACCESS.2020.3047942.

[13] Herlambang Dwi Prasetyo, Pandu Ananto Hogantara, and Ika Nurlaili Isnainiyah, "A Web-Based Diabetes Prediction Application Using XGBoost Algorithm," *Data Science: Journal of Computing and Applied Informatics*, vol. 5, no. 2, pp. 49–59, Jul. 2021, doi: 10.32734/jocai.v5.i2-6290.

[14] L. W. Astuti, I. Saluza, E. Yulianti, and D. Dhamayanti, "Feature Selection Menggunakan Binary Wheal Optimizaton Algorithm (BWOA) pada Klasifikasi Penyakit Diabetes," *Jurnal Ilmiah Informatika Global*, vol. 13, no. 1, Mar. 2022, doi: 10.36982/jiig.v13i1.2057.

[15] S. Shankar and Manikandan, "Diagnosis of diabetes diseases using optimized fuzzy rule set by grey wolf optimization," *Pattern Recognit. Lett.*, vol. 125, pp. 432–438, Jul. 2019, doi: 10.1016/j.patrec.2019.06.005.

[16] UC Irvine Machine Learning Repository, "Pima Indians Diabetes Database," Kaggle. Accessed: Apr. 25, 2024. [Online]. Available: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

[17] UC Irvine Machine Learning Repository, "Early Stage Diabetes Risk Prediction," archive.ics.uci.edu. Accessed: Apr. 25, 2024. [Online]. Available: https://archive.ics.uci.edu/dataset/529/early+stage+diabetes+risk+prediction+dataset

[18] R. Hoyt, "Type 2 Diabetes." [Online]. Available: https://figshare.com/articles/dataset/Type_2_Diabetes/8011535

[19] O. Tarkhaneh, T. T. Nguyen, and S. Mazaheri, "A novel wrapper-based feature subset selection method using modified binary differential evolution algorithm," *Inf. Sci. (N Y).*, vol. 565, pp. 278–305, Jul. 2021, doi: 10.1016/j.ins.2021.02.061.

[20] N. Bacanin, K. Venkatachalam, T. Bezdan, M. Zivkovic, and M. Abouhawwash, "A novel firefly algorithm approach for efficient feature selection with COVID-19 dataset," *Microprocess. Microsyst.*, vol. 98, Apr. 2023, doi: 10.1016/j.micpro.2023.104778.

[21] X.-S. Yang, "Flower Pollination Algorithm for Global Optimization," in *Unconventional Computation and Natural Computation*, N. Durand-Lose Jérôme and Jonoska, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 240–249.

[22] F. Jia, S. Luo, G. Yin, and Y. Ye, "A Novel Variant of the Salp Swarm Algorithm for Engineering Optimization," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 13, no. 3, pp. 131–149, Jun. 2023, doi: 10.2478/jaiscr-2023-0011.

[23] H. Das, B. Naik, and H. S. Behera, "A Jaya algorithm based wrapper method for optimal feature selection in supervised classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 3851–3863, Jun. 2022, doi: 10.1016/j.jksuci.2020.05.002.

[24] Y. Ramadhani, A. Mubarok, S. Hidayatullah, and W. Wiguna, "Attribute Optimization: Genetic Algorithms and Neural Network for Voice Analysis Classification of Parkinson's Disease," Scitepress, Aug. 2020, pp. 3074–3079. doi: 10.5220/0009947030743079.

[25] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1995, pp. 1942–1948 vol.4. doi: 10.1109/ICNN.1995.488968.

[26] J. Too, A. R. Abdullah, and N. M. Saad, "Hybrid binary particle swarm optimization differential evolution-based feature selection for EMG signals classification," *Axioms*, vol. 8, no. 3, 2019, doi: 10.3390/axioms8030079.

[27] R. B. Mohamed, M. M. Yusof, N. Wahid, N. Murli, and M. Othman, "Bat algorithm and k-means techniques for classification performance improvement," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 15, no. 3, pp. 1411–1418, Sep. 2019, doi: 10.11591/ijeecs.v15.i3.pp1411-1418.

[28] M. Alzaqebah *et al.*, "Memory based cuckoo search algorithm for feature selection of gene expression dataset," *Inform. Med. Unlocked*, vol. 24, Jan. 2021, doi: 10.1016/j.imu.2021.100572.

[29] Q. Al-Tashi, H. Md Rais, S. J. Abdulkadir, S. Mirjalili, and H. Alhussian, "A Review of Grey Wolf Optimizer-Based Feature Selection Methods for Classification," in *Evolutionary Machine Learning Techniques*, 2020, pp. 273–286. doi: 10.1007/978-981-32-9990-0_13.

[30] M. Bansal, A. Goyal, and A. Choudhary, "A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning," *Decision Analytics Journal*, vol. 3, p. 100071, Jun. 2022, doi: 10.1016/j.dajour.2022.100071.

[31] Y. Ramdhani, D. F. Apra, and D. P. Alamsyah, “Feature selection optimization based on genetic algorithm for support vector classification varieties of raisin,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 30, no. 1, p. 192, Apr. 2023, doi: 10.11591/ijeecs.v30.i1.pp192-199.

[32] R. Patil, S. Tamane, S. A. Rawandale, and K. Patil, “A modified mayfly-SVM approach for early detection of type 2 diabetes mellitus,” *International Journal of Electrical and Computer Engineering*, vol. 12, no. 1, pp. 524–533, Feb. 2022, doi: 10.11591/ijece.v12i1.pp524-533.

[33] M. Vishwakarma and N. Kesswani, “A new two-phase intrusion detection system with Naïve Bayes machine learning for data classification and elliptic envelop method for anomaly detection,” *Decision Analytics Journal*, vol. 7, Jun. 2023, doi: 10.1016/j.dajour.2023.100233.

[34] A. Wahab, S. Samarinda, I. Lishania, R. Goejantoro, and Y. N. Nasution, “Comparison of the Classification for Naive Bayes Method and the Decision Tree Algorithm (J48) for Stroke Patients in Abdul Wahab Sjahranie Samarinda Hospital,” *Jurnal EKSPONENSIAL*, vol. 10, no. 2, 2019.

[35] J. Daniel and J. H. Martin, “Logistic Regression,” in *Speech and Language Processing*, vol. 1, California: Stanford University, 2024, ch. 5, pp. 1–25. Accessed: Jul. 13, 2024. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/5.pdf

[36] D. Westari, “Performa Comparison of the K-Means Method for Classification in Diabetes Patients Using Two Normalization Methods,” *INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH AND ANALYSIS*, vol. 04, no. 01, Jan. 2021, doi: 10.47191/ijmra/v4-i1-03.

[37] I. M. Karo and Hendriyana, “Klasifikasi Penderita Diabetes menggunakan Algoritma Machine Learning dan Z-Score,” *Jurnal Teknologi Terpadu*, vol. 8, pp. 1–6, 2022.

[38] M. R. Belgaum *et al.*, “Enhancing the Efficiency of Diabetes Prediction through Training and Classification using PCA and LR Model,” *Annals of Emerging Technologies in Computing*, vol. 7, no. 3, pp. 78–91, 2023, doi: 10.33166/AETiC.2023.03.004.

[39] V. Diranisha, A. Triayudi, and R. T. Komalasari, “Implementation of K-Nearest Neighbour (KNN) Algorithm and Random Forest Algorithm in Identifying Diabetes,” *SAGA: Journal of Technology and Information System*, vol. 2, May 2024, doi: 10.58905/SAGA.vol2i2.253.

[40] N. Nipa, M. H. Riyad, S. Satu, Walliullah, K. C. Howlader, and M. A. Moni, “Clinically adaptable machine learning model to identify early appreciable features of diabetes,” *Intelligent Medicine*, vol. 4, no. 1, pp. 22–32, Feb. 2024, doi: 10.1016/j.imed.2023.01.003.

[41] S. Wei, X. Zhao, and C. Miao, “A comprehensive exploration to the machine learning techniques for diabetes identification,” in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, 2018, pp. 291–295. doi: 10.1109/WF-IoT.2018.8355130.

[42] R. D. Joshi and C. K. Dhakal, “Predicting type 2 diabetes using logistic regression and machine learning approaches,” *Int. J. Environ. Res. Public Health*, vol. 18, no. 14, Jul. 2021, doi: 10.3390/ijerph18147346.

[43] D. Sanghavi, D. Sanghavi, and N. Patil, “Early-Stage Diabetes Mellitus Risk Prediction and Symptom Association: A Comparative Analysis Using Feature Importance,” *Educational Administration Theory and Practices*, Jan. 2024, doi: 10.53555/kuey.v30i1.6939.

[44] P. Zhang, C. Fonnesbeck, D. C. Schmidt, J. White, and S. A. Mulvaney, “Understanding Barriers to Diabetes Self-Management Using Momentary Assessment and Machine Learning.”

[45] P. Rajendra and S. Latifi, “Prediction of diabetes using logistic regression and ensemble techniques,” *Computer Methods and Programs in Biomedicine Update*, vol. 1, Jan. 2021, doi: 10.1016/j.cmpbup.2021.100032.

[46] E. Preprint, S. Gill, and P. Pathwar, “Prediction of Diabetes Using Various Feature Selection and Machine Learning Paradigms,” 2021.

[47] K. Verma and P. Harshavardhanan, “Type-2 Diabetes Prediction using Machine Learning Algorithms and Ensembles with Hyperparameters,” 2024. [Online]. Available: http://creativecommons.org/licenses/by/3.0/,whichpermitsunrestricteduse,providedtheoriginalauthorandsourcearecredited.

[48] G. T. Reddy and N. Khare, “Hybrid Firefly-Bat optimized fuzzy artificial neural network based classifier for diabetes diagnosis,” *International Journal of Intelligent Engineering and Systems*, vol. 10, no. 4, pp. 18–27, Aug. 2017, doi: 10.22266/ijies2017.0831.03.

[49] S. Kumar Bhoi *et al.*, “Prediction of Diabetes in Females of Pima Indian Heritage: A Complete Supervised Learning Approach,” 2021.

[50] N. Hartono, “Multi-Objective Bees Algorithm for Feature Selection,” SciTePress, Dec. 2022, pp. 358–369. doi: 10.5220/0010754200003113.

[51] L. W. Astuti, I. Saluza, E. Yulianti, and D. Dhamayanti, “Feature Selection Menggunakan Binary Wheal Optimizaton Algorithm (BWOA) pada Klasifikasi Penyakit Diabetes,” *Jurnal Ilmiah Informatika Global*, vol. 13, no. 1, Mar. 2022, doi: 10.36982/jiig.v13i1.2057.

[52] S. Hassan, T. Akter, F. Tasnim, and M. K. Newaz, “Machine Learning Models to Identify Discriminatory Factors of Diabetes Subtypes,” in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, Springer Science and Business Media Deutschland GmbH, 2023, pp. 55–67. doi: 10.1007/978-3-031-34622-4_5.

[53] B. P. Kumar, “Diabetes Prediction and Comparative Analysis Using Machine Learning Algorithm,” *International Research Journal of Modernization in Engineering Technology and Science*, vol. 4, no. 5, pp. 4688–4696, 2022, [Online]. Available: www.irjmets.com

[54] V. Vakil, S. Pachchigar, C. Chavda, and S. Soni, “Explainable predictions of different machine learning algorithms used to predict Early Stage diabetes,” 2021.

[55] P. Rajendra and S. Latifi, “Prediction of diabetes using logistic regression and ensemble techniques,” *Computer Methods and Programs in Biomedicine Update*, vol. 1, Jan. 2021, doi: 10.1016/j.cmpbup.2021.100032.

[56] S. Balasubramanian, R. Kashyap, S. T. CVN, and M. Anuradha, “Hybrid Prediction Model for Type-2 Diabetes with Class Imbalance,” in *Proceedings of the 2020 IEEE International Conference on Machine Learning and Applied Network Technologies, ICMLANT 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020. doi: 10.1109/ICMLANT50963.2020.9355975.

[57] S. Gill and P. Pathwar, “Prediction of Diabetes Using Various Feature Selection and Machine Learning Paradigms,” 2021.

[58] Edgar Ceh-Varela, L. Maes, and Sarbagya Ratna Shakya, “Machine Learning Analysis of Factors Contributing to Diabetes Development,” *Cloud Computing and Data Science*, pp. 157–182, Jan. 2024, doi: 10.37256/ccds.5120243751.

[59] M. M. Mijwil and M. Aljanabi, “A Comparative Analysis of Machine Learning Algorithms for Classification of Diabetes Utilizing Confusion Matrix Analysis,” *Baghdad Science Journal*, vol. 21, no. 5, pp. 1712–1728, 2024, doi: 10.21123/BSJ.2023.9010.