

Keywords: diabetes detection, QIFPNN, accuracy, F1-score

Yulianto Triwahyuadi POLLY <sup>1\*</sup>, Adriana FANGGIDAE <sup>1</sup>,  
 Juan Rizky Manuel LEDOH <sup>1</sup>, Clarissa Elfira AMOS PAH <sup>1</sup>, Bertha S. DJAHI <sup>1</sup>,  
 Kisan Emiliano Rape TUPEN <sup>1</sup>

<sup>1</sup> Universitas Nusa Cendana, Indonesia, yuliantopolly@staf.undana.ac.id, adrianafanggidae@staf.undana.ac.id, juanledoh@staf.undana.ac.id, clarissaelfira@staf.undana.ac.id, bertha.djahi@staf.undana.ac.id, kisanemiliano1821@gmail.com

\* Corresponding author: yuliantopolly@staf.undana.ac.id

## A new approach for diabetes risk detection using quadratic interpolation flower pollination neural network

### Abstract

*This study aims to evaluate and compare five algorithms in diabetes detection, namely Flower Pollination Neural Network (FPNN), Particle Swarm Optimization Neural Network (PSOENN), Bat Artificial Neural Network (BANN), Stochastic Gradient Descent (SGD), and Quadratic Interpolation Flower Pollination Neural Network (QIFPNN). These algorithms were tested on a diabetes risk dataset divided into training, validation, and testing subsets. The evaluation was based on three main aspects: accuracy, F1 score, and training time. Experimental results showed that QIFPNN outperformed others with an average accuracy of 97.90% and an F1 score of 98.30%, although it required the longest training time (4107.89 seconds). FPNN and BANN achieved competitive accuracy (97.34% and 97.43%) and F1 scores (97.84% and 97.91%), while SGD offered a favorable trade-off with accuracy of 96.87%, F1 score of 97.42%, and the shortest training time (584.50 seconds). PSOENN performed less well with an average accuracy of 89.26% and an F1 score of 91.45%. These results indicate that QIFPNN can be relied upon as an effective diabetes risk detection model with superior predictive performance. Although the training time of QIFPNN is longer due to its sophisticated optimization process, this is only a concern during model development, as the final trained model can be efficiently used for real-time prediction in practical applications.*

### 1. INTRODUCTION

Diabetes mellitus is a chronic disease resulting from impaired insulin production or utilization, leading to high blood glucose levels. If left untreated, it can lead to serious complications such as cardiovascular disease, kidney failure and vision loss. The global burden of diabetes is growing rapidly. In 2021, 537 million people worldwide will be living with diabetes, rising to 783 million by 2045 (International Diabetes Federation, 2024). Indonesia ranks fifth in the world with 19.5 million cases in 2021, expected to rise to 28.6 million by 2045 (Mediakom, 2024).

This trend has prompted researchers to develop accurate machine learning (ML) prediction methods. Abed Mohammed et al. (2024) proposed a hybrid K-means and PCA method with Random Forest (RF), which achieved 95.2% accuracy. Amma (2024) introduced En-RfRsK, which integrates RF, k-NN and R-SVM and achieved 88.89% accuracy. Nissar et al. (2024) applied several ML algorithms, with RF achieving 96.15%. Bhat et al. (2022) compared six models and found RF to be the most effective (98%). Chaves & Marques (2021) found neural networks to be the best with 98.1% accuracy. M. S. Islam et al. (2023), Oladimeji et al. (2024) and Emon et al. (2021) also confirmed the superiority of RF with accuracies up to 99%. M. M. F. Islam et al. (2020) used cross validation and reported 97.4% with RF. Shaik & Siddique (2024) investigated diabetes classification using ECG and PPG signals. While many studies report high accuracy, most use only training-test splitting and oversampling, which may not ensure robust generalization. This limitation motivates the development of a more robust model with rigorous validation.

Artificial intelligence has become an important pillar in the development of intelligent diagnostic tools, especially in the early detection of chronic and degenerative diseases. Recent studies have demonstrated the effectiveness of artificial neural networks (ANN) in medical diagnostics, including the identification of osteoarthritis using vibroarthrography signals (Machrowska et al., 2024b; Machrowska et al., 2024a). For example, (Machrowska et al., 2024b) reported an MLP-based classification accuracy of 91.07% for the detection of knee osteoarthritis based on recurrence quantification analysis. In a follow-up study, Machrowska et al. (2024a) applied EEMD-DFA algorithms combined with ANN classifiers and achieved an accuracy of 93%, with both sensitivity and specificity reaching 0.93. Similarly, Karpiński, (2022) investigated vibroacoustic signal analysis for osteoarthritis diagnosis using machine learning techniques and showed promising results with reduced diagnostic complexity. The application of ANN is not limited to musculoskeletal disorders. Kulisz et al. (2021) successfully modeled the groundwater quality index using ANN with high predictive accuracy ( $R^2 = 0.9984$ ), and Kujawska et al. (2022) demonstrated that ANN and LSTM models outperformed traditional regression methods in predicting PM10 air pollution levels. These studies confirm that ANN-based approaches have great potential for modelling complex, non-linear biomedical data, especially when coupled with hybrid optimization mechanisms. The proposed QIFPNN model, which combines neural networks with Quadratic Interpolation Flower Pollination optimization, is well aligned with this trend and thus represents a promising direction for improving diabetes detection systems. Although these studies did not focus specifically on diabetes, they reflect a broader trend in medical AI research that emphasizes data-driven decision support and rigorous model evaluation. However, in the context of diabetes prediction, different studies often use simpler data partitioning strategies.

All previous studies on diabetes identified their best models by dividing the data into only two groups: training data and test data. This approach potentially limits the accuracy of diabetes identification. In the studies by Bhat et al. (2022); M. S. Islam et al. (2023); and Oladimeji et al. (2024), the datasets were conditioned using oversampling techniques to create balanced datasets with equal class distributions. Here, we propose a novel model using the Quadratic Interpolation Flower Pollination Neural Network (QIFPNN). QIFPNN is designed to identify the best model by utilizing three different sets of data: training data, validation data, and test data. These groups are managed based on stratified k-fold cross-validation (Prusty et al., 2022; Raschka, 2018; Mahesh et al., 2023) applied to unbalanced datasets. This stratification ensures that each fold maintains the same class proportions as the original dataset. This approach provides excellent model stability in recognizing training, validation, and test data, including unseen data excluded from model building. QIFPNN has been shown to perform effectively on balanced and unbalanced, binary and non-binary data sets (Polly et al., 2021; 2023).

Given the high prevalence and significant impact of diabetes, the development of more accurate and stable prediction methods is critical. An ideal prediction method is expected to predict the risk of diabetes with high accuracy, especially in the early stages. QIFPNN is a modification of the Flower Pollination Algorithm (FPA) designed to optimize the training process of artificial neural networks, specifically the Multi-Layer Perceptron (MLP). QIFPNN introduces a novel mechanism, Quadratic Interpolation Flower Pollination (QIFP), to improve the efficiency and accuracy of finding optimal solutions during neural network training. Using quadratic interpolation (QI) in global pollination improves the efficiency of search space exploration, as QI is more effective in finding local minima than relying solely on Levy flight or random search. Thus, QIFPNN can prevent neural networks from overfitting or getting trapped in suboptimal solutions, which is often the case with conventional optimization methods (Polly et al., 2021; 2023). With the development of a better prediction method, it is hoped that early detection of diabetes can be improved, disease management can be enhanced, and healthcare costs can be reduced. Therefore, the development of more accurate and stable diabetes prediction models is a critical step in the prevention and control of diabetes mellitus.

In general, this study aims to develop a new model for predicting diabetes risk that is more accurate and stable. Specifically, the objectives of this research are: (1) to develop a diabetes detection model using the QIFPNN approach and (2) to compare the performance of the developed model with other metaheuristic-based algorithms such as Flower Pollination Neural Network (FPNN), Bat Algorithm Neural Network (BANN), and Particle Swarm Optimization Neural Network (PSO) (Aalimahmoody et al., 2021; Al Bataineh et al., 2022; Chiroma et al., 2016; Yang; 2012; 2014; 2020) and Stochastic Gradient Descent (SGD) to evaluate the effectiveness of the developed model in diabetes detection. This research is expected to significantly contribute to the development of new methods for more accurate and stable diabetes detection. The resulting detection model can be used for diabetes screening, enabling early detection and more appropriate treatment. In addition,

this study is expected to provide valuable information for policy makers in designing programs to prevent and control diabetes.

## 2. METHOD

### 2.1. Data

This study utilizes the Early Stage Diabetes Risk Prediction dataset from the UCI Machine Learning Repository. UCI (2020) which contains information on the risk of early-stage diabetes based on several factors measured by patients.

The dataset consists of categorical and numerical variables. Most of the features in this dataset are binary (in "yes" or "no" format) and represent the presence of certain symptoms or conditions, except for the feature "age" which is a numeric (integer) value indicating the patient's age. The dataset contains 16 attributes, 15 of which are input features used as predictors, and 1 of which is the output label indicating diabetes risk, which is the target of the classification. The details of the attributes are shown in Table 1.

**Tab. 1. Details of the attributes from the "Early Stage Diabetes Risk Prediction" dataset**

Attributes	Data Type	Value	Description
Age	Numeric	16 to 90 years	Patient age range
Gender	Categorical	Male/Female	Patient gender
Polyuria	Categorical	Yes/No	Frequent urination
Polydipsia	Categorical	Yes/No	Frequent thirst
Sudden weight loss	Categorical	Yes/No	Sudden weight loss
Weakness	Categorical	Yes/No	Feeling weak
Polyphagia	Categorical	Yes/No	Excessive hunger
Genital thrush	Categorical	Yes/No	Fungal infections in the genital area
Visual blurring	Categorical	Yes/No	Blurred vision
Itching	Categorical	Yes/No	Itching
Irritability	Categorical	Yes/No	Irritability
Delayed healing	Categorical	Yes/No	Slow wound healing
Partial paresis	Categorical	Yes/No	Weakness in parts of the body
Muscle stiffness	Categorical	Yes/No	Muscle stiffness
Alopecia	Categorical	Yes/No	Baldness
Class	Categorical	Positive/Negative	Diabetes risk classification results: Positive (diagnosed with diabetes) or Negative (not diagnosed with diabetes)

This dataset consists of 520 patient samples, divided into 320 samples (approximately 61.5% of the total dataset) classified as positive (diagnosed with diabetes) and 200 samples (approximately 38.5% of the total dataset) classified as negative (not diagnosed with diabetes), making this dataset class unbalanced, with more samples classified as positive than negative. Most of the trait attributes in this dataset are binary, with values limited to "Yes" or "No". It indicates the presence of symptoms or conditions associated with diabetes risk. The Age characteristic ranges from 16 to 90 years and represents the age variation of the patients in this dataset. There are no missing values in this dataset.

Data preprocessing is critical to ensuring the quality of the dataset used in machine learning. The preprocessing steps applied to this dataset include techniques such as categorical data encoding and normalization. The categorical data encoding technique converts categorical data such as "Yes"/"No" for binary attributes, "Male"/"Female" for gender attributes, and "Positive/Negative" for class attributes into numerical format, i.e., Yes = 1, No = 0 for binary attributes, Male = 1, Female = 0 for gender attributes, and Positive = 1, Negative = 0 for the class attribute. The normalization technique is applied to the numeric Age attribute, which has a wide range of values (from 16 to 90). The normalization technique scales the Age values to a range of 0 to 1 using equation (1).

$$y = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

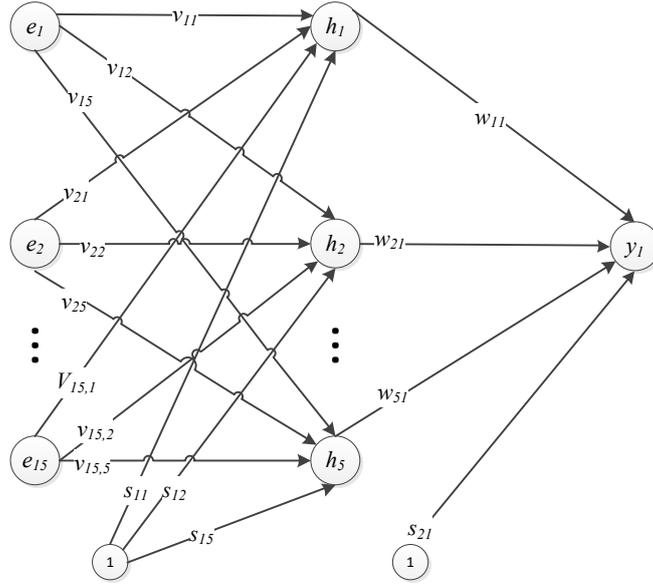
where:  $x$  – the Age feature,  
 $y$  – the normalised data for the Age feature,  
 $x_{max}$  – the maximum values of the Age feature,  
 $x_{min}$  – the minimum values of the Age feature.

## 2.2. Description of QIFPA

The architecture of QIFPNN can be seen in Figure 1, where it adopts the MLP architecture. In this study, the QIFPNN architecture satisfies several aspects, including the number of neurons in the input layer corresponding to the number of features in the dataset. For the Early Stage Diabetes Risk Prediction dataset, there are 15 input features, so the input layer has 15 neurons ( $e_1$  to  $e_{15}$ ). The number of hidden layers is one, and the number of neurons in it follows equation (2) (Lin et al., 2015; Polly et al., 2021) ( $h_1$  to  $h_5$ ). In the output layer, since this is a binary classification problem, the output layer will have one neuron ( $y_1$ ) represents the predicted class (positive / negative).

$$b = \sqrt{(a + c)} + Z \quad (2)$$

The activation function used in the hidden layer  $f(h_{out})$  is given by equation (3), and the activation function in the output layer  $f(y_{out})$  is shown in equation (4), both of which are binary sigmoid functions. The binary sigmoid function converts the output to a probability ranging from 0 to 1 (Polly, 2022). The binary sigmoid result at the output layer is then used to determine whether the result belongs to the positive (1) or negative (0) class, using a threshold of 0.5.



**Fig. 1. MLP architecture for diabetes with the number of neurons in the input layer as  $a = 15$ , the number of neurons in the hidden layer as  $b = 5$ , and the number of neurons in the output layer as  $c = 1$**

where:  $b$  – the number of neurons within the hidden layer,  
 $a$  – the number of neurons within the input layer,  
 $c$  – the number of neurons within the output layer,  
 $Z$  – it is set to one. The setting of  $Z = 1$  aims to simplify the architecture size of QIFPNN for faster learning time.

$$f(h_{out}) = \frac{1}{1 + \exp(-h_{out})} \in (0,1) \quad (3)$$

$$f(y_{out}) = \frac{1}{1 + \exp(-y_{out})} \in (0,1) \quad (4)$$

QIFPNN integrates QI and FPA into a neural network and represents an innovative approach to improve the optimization and training process of neural networks. FPA is a metaheuristic optimization algorithm based on the flower pollination process. This algorithm has two main stages: Global Pollination and Local Pollination. In Global Pollination, the agent (solution) performs exploration in the global search space, mimicking the cross-pollination process where pollen (solution information) is carried by wind or distant insects. In local pollination, the agent performs exploration around the current solution at a closer distance, mimicking the self-pollination process, which is more local. Key elements in FPA include Levy Flight and Switch Probability( $p$ ). FPA uses Levy Flight to support global exploration, a random search technique that jumps far from one solution to another in the search space. The switch probability( $p$ ) in FPA determines whether the agent will perform global or local pollination. QI is a numerical optimization method used to estimate the minimum point of a quadratic function. QI involves fitting a quadratic function (a 2nd degree polynomial) based on three points on the curve, and from this fit, determines a new minimum point. This technique is used to refine the search for a more precise solution, especially in local search environments.

In QIFPNN, the global pollination in FPA is modified using QI. Thus, it is known as the Quadratic Interpolation Flower Pollination Algorithm (QIFPA). The main changes are as follows:

Global pollination with QI: In the global pollination process, QIFPA performs random searches using QI to refine the global search process. QI determines a better local solution by estimating the optimal position based on quadratic function fitting. The global pollination process with QI is given by equation (5).

$$\bar{x}_i^{t+1} = \bar{x}_i^t + \bar{Q}(\bar{g}_* - \bar{x}_i^t) \quad (5)$$

where:  $\bar{x}_i^t$  – the pollen vector position or the solution vector for  $i$  ( $i = 1, 2, \dots, n$ ) at iteration  $t$ ,  
 $n$  – the population size or the number of solution vectors,  
 $\bar{g}_*$  – the best solution vector currently found among all solutions at this iteration,  
 $\bar{Q}$  – the step vector.

The population is referred to in Equations (6) and (7).

$$X^t = \{\bar{x}_1^t, \bar{x}_2^t, \dots, \bar{x}_n^t\} \quad (6)$$

$$\bar{x}_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{id}^t) \quad (7)$$

Population-based metaheuristic algorithms such as QIFPNN imply that each individual/agent/pollen (solution) in the population acts as a candidate solution. At each generation (iteration), the algorithm improves the quality of the population. This population represents a set of solution vectors in the search space.  $X^t$  is the set of solution vectors at iteration  $t$  while  $d$  is the dimension or the number of solution variables. Thus, each element of the solution vector can be written as  $x_{iz}^t$  ( $z = 1, 2, \dots, d$ ).

The derivative of the quadratic polynomial function is used to find the value of  $r^*$  which results in the minimum or maximum fitness. Figure 2 shows this process. Then the step vector  $\bar{Q}$  is constructed by the following steps:

1. Using a normal distribution  $\bar{Q} \sim N(\mu, \sigma)$  to generate  $\bar{Q}$ , where  $\mu = r^*$  and  $\sigma = f(g^*)$ ,
2. Set zero, 20 per cent of the elements of the vector  $\bar{Q}$  using Equation (8),

$$Q_c = \begin{cases} 0, & c=l \\ Q_c, & \text{otherwise} \end{cases} \quad (8)$$

where:  $Q_c$  – an element of the step vector  $\bar{Q}$  at index  $c$ ,  
 $l$  – a random integer that satisfies  $l \in [1, d]$ , generated for 20 per cent of  $d$ .

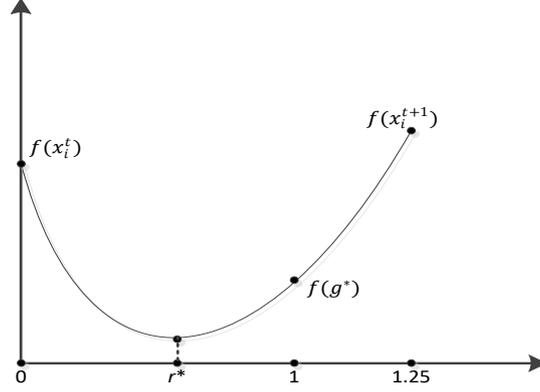


Fig. 2. The illustration of the minimum point  $r^*$  of a quadratic function based on three points on the curve

Next, QIFPA's search space starts from a narrower region and gradually expands until it reaches the search space  $[LbReal, UbReal]^d$ . Here,  $LbReal$  and  $UbReal$  are the lower and upper bounds of the search space, respectively, with values of  $LbReal = -80$  and  $UbReal = 80$ . This expansion process is performed in the following steps:

1. Identification of the initial search space:

$$[Lb, Ub]^d = [-1, 1]^d \quad (9)$$

2. For every 50 iterations, expand the search space by:

$$[Lb, Ub]^d = \begin{cases} [Lb - 1, Ub + 1]^d & t \text{ MOD } 50 = 0 \\ [Lb, Ub]^d & \text{otherwise} \end{cases} \quad (10)$$

3. Repeat step 2 until  $Lb$  equals  $LbReal$  and  $Ub$  equals  $UbReal$ . Where the values of  $LbReal$  and  $UbReal$  are of integer type, and zero is the result of the sum of  $LbReal$  and  $UbReal$ .

Local pollination follows the original FPA mechanism, where agents perform local searches based on the difference between the current solution and the best solution nearby, referencing equation (11).

$$\bar{x}_i^{t+1} = \bar{x}_i^t + \varepsilon(\bar{x}_j^t - \bar{x}_k^t) \quad (11)$$

Random walk  $\varepsilon$  is a real random number uniformly distributed under the condition that  $\varepsilon$  is in the range  $[0, 1]$  while  $\bar{x}_j^t$  and  $\bar{x}_k^t$  represent the positions of the pollen  $j$  and  $k$  respectively.

### 2.3. Implementation in MLP

QIFPNN uses QIFPA to train the parameters (weights and biases) of the neural network architecture, specifically the MLP architecture. The pollen position vector or solution vector  $\bar{x}_i$  for diabetes is shown in Figure 3.

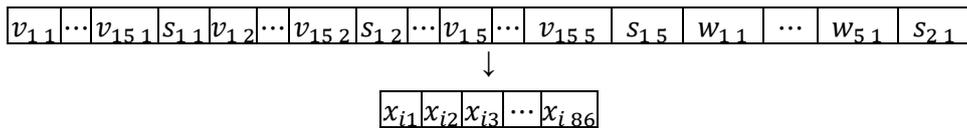


Fig. 3. Illustration of the pollen position vector or solution vector for diabetes with  $a = 15$  (number of input neurons),  $c = 1$  (number of output neurons), and  $b = 5$  (number of neurons in the hidden layer)

where the number of solution variables  $d$  is equal to the number of weights and biases, namely:

$$d = (15 \times 5) + 5 + (5 \times 1) + 1 = 86$$

## 2.4. Optimisation process

The training process of the neural network model in QIFPNN is a combination of FPA with QI, which serves as the optimization mechanism for training the weights and biases of the neural network. The training process is shown in Figure 4. The following is a step-by-step explanation of the model training process.

The process begins by defining the architecture and parameters of QIFPNN, including the number of neurons in the input, hidden, and output layers.  $(a, b, c)$  the population size of the pollen grains ( $n$ ); the lower and upper bounds  $[LbReal, UbReal]$  the error target ( $targetError$ ) the convergence iteration threshold ( $iterThOfCA$ ) the maximum number of iterations ( $maxIteration$ ) and the switching probability ( $p$ ). QIFPNN performs optimization using QIFPA, which replaces gradient-based optimization methods (such as Stochastic Gradient Descent) with a metaheuristics-based approach. The optimization steps are as follows:

– Population Initialisation:

1. At the beginning of the training process, a group of agents/pollen grains/solutions ( $x_i$ ) is initialized. Each agent represents a set of weights and biases of the neural network.
2. The initial weights and biases are randomly initialized within the range of  $(x_i = Lb + Border[0,1](Ub - Lb))$  according to a uniform distribution.

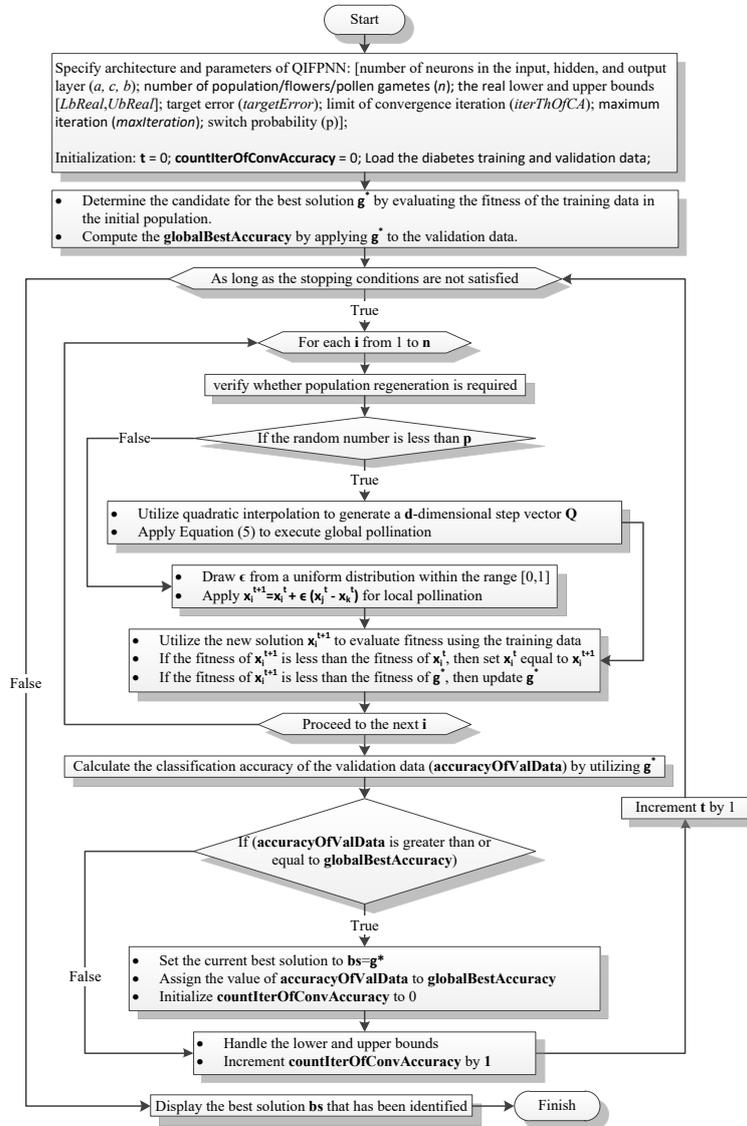


Fig. 4. Flowchart of the QIFPNN training process

– Fitness Evaluation (Loss Function Evaluation):

Each agent in the population is evaluated to determine how effectively it (the set of weights and biases) minimizes the loss function. This is achieved by passing the input data through the neural network and calculating the error or loss at the output using the Mean Square Error (MSE) loss function.

– Global Pollination with QI:

At the global pollination stage, QIFPA uses QI to update weights and biases. QI aims to find a globally superior solution by estimating the optimal position of weights and biases through quadratic function fitting. The three best current solutions perform quadratic interpolation, which predicts a new optimal point in parameter space. Weights and biases are updated using this optimal point, accelerating convergence to a superior solution. This allows the algorithm to perform a more intelligent global search than random search methods.

– Local Pollination:

1. In the local pollination phase, agents exploit solutions in their environment. Like local search techniques, agents attempt to improve the weights and biases based on the best available solutions.
2. Local pollination uses an exploitation strategy where the weights and biases are updated based on the best solution within the population.
3. The switching probability determines whether an agent will perform global pollination (using QI) or local pollination. This probability is set to 0.8 to ensure that global pollination is performed more often than local pollination.

– Iteration until convergence:

The global and local pollination processes are repeated for several iterations until the stopping criteria are met, which are

1. The maximum number of iterations has been reached ( $maxIteration = 4000$ ) or
2. A slight change in the loss function from one iteration to the next (convergence), i.e,  $targetError < 10^{-3}$  or
3. the limit of the convergence iteration ( $iterThOfCA = 700$ ) is reached.

– Model evaluation:

The best-trained model is validated using stratified K-fold cross-validation, which ensures consistency of the class distribution across folds, critical for unbalanced data such as diabetes risk datasets. Final performance is evaluated on the test set using accuracy and F1 score metrics.

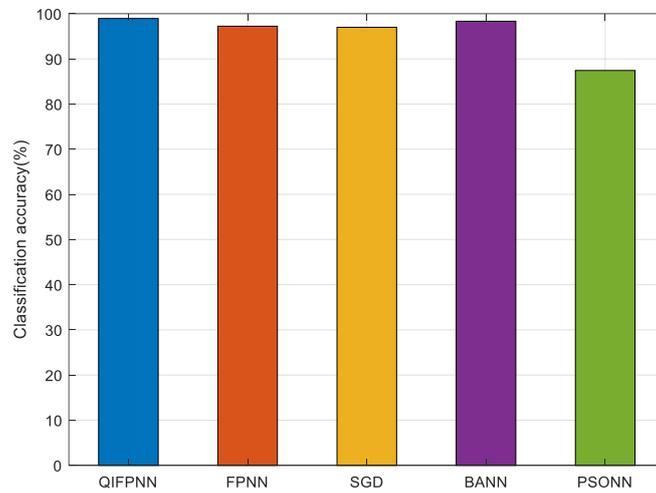
Stratified K-fold cross-validation ensures that the model does not overfit and generalizes well to new data. Stratified K-fold cross-validation is a variation of k-fold cross-validation in which the data are split so that each fold retains the same class proportion as in the original dataset. (Prusty et al., 2022; Raschka, 2018; Mahesh et al., 2023).. This is critical for unbalanced datasets such as the Diabetes Risk Prediction dataset.

### 3. RESULTS AND DISCUSSION

The evaluation results of the models on the training subset, which illustrate the performance of each algorithm, are presented in Table 2 and Figure 5. These results indicate that the QIFPNN algorithm achieved the highest accuracy among the five algorithms tested.

**Tab. 2. Average 20 runs of classification accuracy per fold in the training subset of diabetes**

Algorithm		Fold										Mean
		1	2	3	4	5	6	7	8	9	10	
QIFPNN (%)	Min	88.39	99.29	95.25	98.10	95.25	94.77	98.10	95.72	97.39	98.34	96.06
	Max	100.00	100.00	100.00	100.00	100.00	99.76	100.00	100.00	100.00	100.00	99.98
	Ave	97.89	99.64	98.78	99.63	98.19	98.92	99.25	98.44	99.37	99.35	98.95
FPNN (%)	Min	95.50	87.89	93.82	90.50	90.74	83.37	91.92	92.87	95.49	93.13	91.52
	Max	99.05	99.05	99.29	99.05	99.05	99.29	98.57	99.29	99.29	98.82	99.07
	Ave	97.59	97.35	97.79	97.15	97.05	96.41	96.65	97.03	97.60	97.68	97.23
BANN (%)	Min	95.73	84.56	82.42	74.58	95.72	87.65	89.55	89.31	97.39	93.36	89.03
	Max	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	Ave	98.93	98.80	97.83	98.29	97.96	98.36	98.28	97.39	99.37	98.12	98.33
PSOINN (%)	Min	79.15	78.15	75.77	71.50	76.72	81.00	77.67	75.06	70.31	82.23	76.76
	Max	93.36	97.15	94.30	94.30	93.59	96.44	94.06	93.11	95.25	94.31	94.59
	Ave	88.36	88.00	85.49	87.36	86.86	88.79	87.02	86.59	87.96	87.87	87.43
SGD (%)	Min	95.97	96.67	95.72	88.36	95.01	93.11	92.40	92.87	96.20	93.36	93.97
	Max	99.05	99.05	99.29	99.05	99.05	98.81	99.29	98.57	99.29	98.82	99.03
	Ave	97.81	98.49	96.94	97.07	97.21	96.96	95.93	95.44	97.78	96.18	96.98



**Fig. 5. Average 10-fold classification accuracy of the training subset of diabetes**

FPNN and BANN also showed strong performance, exceeding 97%, while PSOINN showed significantly lower accuracy than the others. Interestingly, SGD showed competitive performance with an accuracy of 96.98%, which was slightly lower than QIFPNN, FPNN and BANN, but significantly better than PSOINN.

The evaluation of the training subset showed that QIFPNN excelled in the detection of diabetes with an accuracy of 98.95%. This superiority can be attributed to QIFPNN's ability to optimize the training process through a more efficient quadratic interpolation mechanism than other algorithms. Although FPNN and BANN achieved respectable accuracies (97.23% and 98.33%, respectively), they could not outperform QIFPNN. SGD, on the other hand, also showed strong performance with an accuracy of 96.98%, making it a viable option due to its simplicity and speed. However, its slightly lower accuracy indicates that its gradient-based approach may not capture complex data patterns as effectively as metaheuristic-driven models such as QIFPNN.

In contrast, PSOINN had a much lower accuracy (87.43%), likely due to inefficiencies in parameter fitting. This limitation can be attributed to the suboptimal population search strategy, which hindered its ability to find the best solutions. These results demonstrate that advanced methods, such as QIFPNN, can significantly improve recognition accuracy compared to traditional approaches. In addition, the performance of SGD is promising, especially in scenarios that require efficient model performance under limited computational resources.

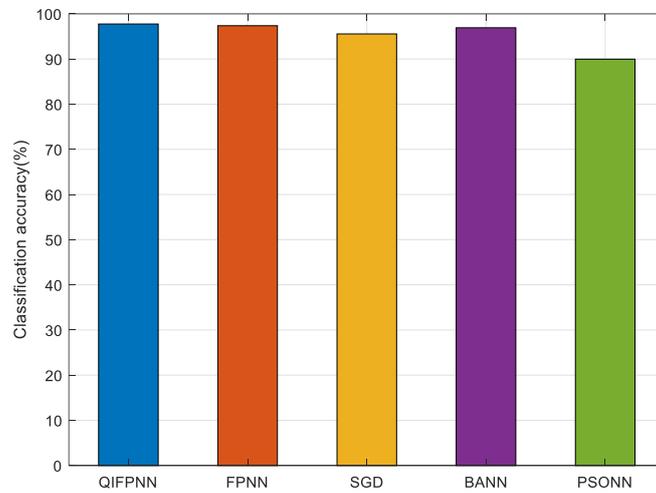
The evaluation results of the models on the validation subset, highlighting the performance of each algorithm, are presented in Table 3 and Figure 6. These results show that the QIFPNN algorithm maintained its superiority and achieved the highest accuracy on both subsets. Although there was a slight decrease in

accuracy on the validation subset compared to the training subset, QIFPNN consistently showed superior performance.

**Tab. 3. Average 20 runs of classification accuracy per fold in the validation subset of diabetes**

Algorithm		Fold										Mean
		1	2	3	4	5	6	7	8	9	10	
QIFPNN (%)	Min	97.83	95.74	100.00	95.74	95.74	93.62	97.87	91.49	95.74	95.65	95.94
	Max	100.00	100.00	100.00	100.00	97.87	100.00	100.00	97.87	100.00	100.00	99.57
	Ave	98.26	97.45	100.00	97.77	97.66	96.28	99.79	94.68	97.34	98.37	97.76
FPNN (%)	Min	97.83	91.49	100.00	95.74	95.74	93.62	97.87	91.49	91.49	95.65	95.09
	Max	100.00	100.00	100.00	100.00	100.00	97.87	100.00	97.87	97.87	100.00	99.36
	Ave	98.48	95.43	100.00	99.36	97.98	95.74	99.47	94.89	94.26	98.15	97.38
BANN (%)	Min	91.30	91.49	89.36	87.23	93.62	82.98	93.62	82.98	91.49	95.65	89.97
	Max	100.00	100.00	100.00	100.00	97.87	100.00	100.00	97.87	100.00	100.00	99.57
	Ave	96.96	96.70	99.36	95.96	97.45	95.21	98.51	94.15	96.81	98.15	96.93
PSOINN (%)	Min	84.78	80.85	89.36	87.23	72.34	85.11	82.98	80.85	74.47	80.43	81.84
	Max	100.00	93.62	100.00	97.87	97.87	95.74	97.87	93.62	91.49	97.83	96.59
	Ave	90.76	86.91	95.53	92.34	90.74	90.43	91.70	87.02	83.62	90.65	89.97
SGD (%)	Min	93.48	91.49	100.00	93.62	97.87	91.49	95.74	91.49	89.36	95.65	94.02
	Max	97.83	95.74	100.00	97.87	97.87	95.74	100.00	93.62	97.87	97.83	97.44
	Ave	95.22	93.62	100.00	95.21	97.87	92.77	97.55	93.30	92.34	97.72	95.56

The accuracy results on the validation subset further confirm previous findings that QIFPNN is a practical algorithm for diabetes detection. With an accuracy of 97.76%, QIFPNN demonstrated strong generalization capabilities, despite a slight drop from its accuracy on the training subset. FPNN and BANN also showed competitive performance, with accuracies of 97.38% and 96.93%, respectively. The slight differences among these results suggest that all three algorithms (QIFPNN, FPNN, and BANN) can effectively detect diabetes; however, QIFPNN has a slight advantage in terms of stability and generalization ability.



**Fig. 6. Average 10-fold classification accuracy of the validation subset of diabetes**

SGD, a conventional gradient-based optimization method, achieved a validation accuracy of 95.56%. Although its performance was lower than that of QIFPNN, FPNN, and BANN, it still demonstrated reasonable generalization. However, the performance gap suggests that SGD may be less able to capture complex patterns in the diabetes dataset compared to population-based algorithms. This difference may be due to the limited exploration capacity of SGD, which relies heavily on gradient direction and may converge prematurely on suboptimal solutions.

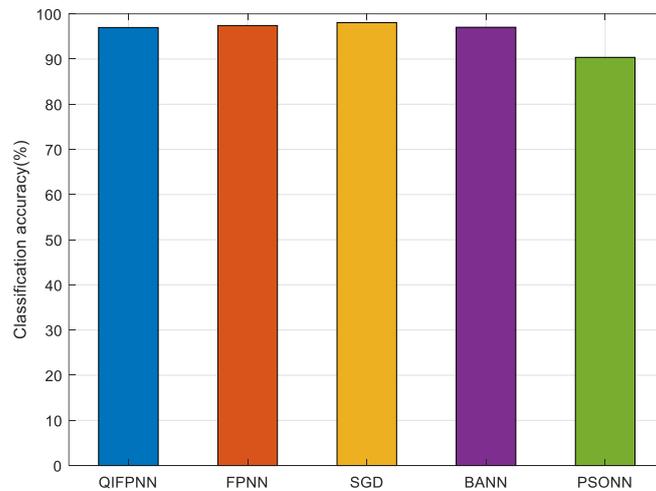
Although PSOINN showed an improvement on the validation subset with an accuracy of 89.97%, it still fell significantly behind the other four algorithms. This indicates that PSOINN needs further optimization or parameter tuning to improve its performance in this context. Overall, the accuracy results on the validation subset indicate that QIFPNN excels during training and maintains its superior performance when tested on

previously unseen data. At the same time, SGD provides a good starting point and performs better than PSONN, although it lags behind the other metaheuristic-based methods.

The evaluation results of the models on the test subset, which illustrate the performance of each algorithm, are shown in Table 4 and Figure 7. These results show that QIFPNN consistently demonstrated strong performance, achieving the highest accuracy on the training and validation subsets, and competitive accuracy on the test subset.

**Tab. 4. Average 20 runs of classification accuracy per fold in the test subset of diabetes**

Algorithm		Fold										Mean
		1	2	3	4	5	6	7	8	9	10	
QIFPNN (%)	Min	92.31	92.31	94.23	94.23	94.23	90.38	94.23	94.23	94.23	96.15	93.65
	Max	98.08	98.08	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	99.62
	Ave	96.25	95.96	97.40	96.92	97.88	96.15	97.50	97.50	96.44	97.79	96.98
FPNN (%)	Min	96.15	90.38	92.31	94.23	92.31	90.38	94.23	92.31	96.15	94.23	93.27
	Max	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	Ave	98.17	97.21	97.88	97.31	96.73	96.92	96.92	97.60	97.50	97.98	97.42
BANN (%)	Min	94.23	88.46	90.38	75.00	94.23	75.00	92.31	90.38	92.31	94.23	88.65
	Max	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	Ave	98.37	96.44	97.31	95.58	98.37	95.19	97.31	97.50	96.54	97.79	97.04
PSONN (%)	Min	76.92	76.92	75.00	69.23	75.00	82.69	78.85	78.85	75.00	80.77	76.92
	Max	96.15	98.08	98.08	98.08	98.08	98.08	96.15	98.08	96.15	98.08	97.50
	Ave	90.58	89.81	90.29	90.38	89.71	92.12	89.81	90.58	89.81	90.67	90.38
SGD (%)	Min	98.08	96.15	96.15	96.15	96.15	94.23	96.15	98.08	96.15	96.15	96.35
	Max	100.00	100.00	100.00	100.00	100.00	98.08	100.00	100.00	100.00	100.00	99.81
	Ave	98.56	97.98	97.69	98.65	97.88	97.31	98.08	98.17	98.27	98.08	98.07



**Fig. 7. Average 10-fold classification accuracy of the test subset of diabetes**

The accuracy results on the test subset show that QIFPNN achieved an accuracy of 96.98%. Although this represents a slight decrease from its performance on the training and validation subsets, it still demonstrates a strong ability to detect diabetes on unseen data. Interestingly, FPNN recorded the highest accuracy on the test subset at 97.42%, suggesting that while QIFPNN excelled during training and validation, FPNN showed slightly better adaptability to the test data. BANN also showed stable performance with an accuracy of 97.04%, reinforcing the finding that all three algorithms (QIFPNN, FPNN, and BANN) are generally reliable for practical applications.

The SGD algorithm, which is a conventional gradient-based training method, achieved an accuracy of 98.07% on the test subset, the highest among all methods. This result indicates that SGD, despite its relatively simple optimization mechanism compared to evolutionary-based methods, can perform remarkably well in generalization when properly configured. Its performance on the test data suggests that SGD was less prone to overfitting than some population-based methods, possibly due to its more deterministic gradient descent approach and efficient convergence.

PSOINN continued to lag behind the other four algorithms with an accuracy of 90.38%. Despite an improvement from its validation results, PSOINN's performance highlights the need for further optimization of its architecture and parameter tuning.

Overall, the accuracy results on the test subset confirm that QIFPNN excels in training and validation and maintains strong performance in the final test phase. However, the superior accuracy of SGD on this subset suggests that it is a highly effective baseline algorithm that should not be overlooked. While FPNN showed an advantage over QIFPNN at this stage, BANN showed stability, including SGD, which introduces a competitive alternative with exceptional generalization ability.

The evaluation results, which present the average classification accuracy across the three subsets (training, validation, and testing) and reflect the performance of each algorithm, are shown in Table 5. These results show that QIFPNN outperformed other algorithms on the training and validation subsets and achieved the highest average accuracy on all three subsets.

**Tab. 5. Recapitulation of mean classification accuracy of the training, validation, and test subsets for the ten folds of diabetes**

Algorithm	Average accuracy of training subset (%)	Average accuracy of validation subset (%)	Average accuracy of test subset (%)	Mean accuracy of training, validation, and test subsets (%)
QIFPNN	98.95	97.76	96.98	97.90
FPNN	97.23	97.38	97.42	97.34
BANN	98.33	96.93	97.04	97.43
PSOINN	87.43	89.97	90.38	89.26
SGD	96.98	95.56	98.07	96.87

The average accuracy across the three subsets provides a clear overview of the overall performance of each algorithm. QIFPNN achieved an average accuracy of 97.90%, demonstrating consistent performance across all evaluation levels. This indicates that QIFPNN can effectively generalize its results based on new data, making it an excellent choice for diabetes detection. FPNN and BANN also showed solid performance, with average accuracies of 97.34% and 97.43%, respectively. Although slightly lower than QIFPNN, these three algorithms proved their strong ability to detect diabetes.

The SGD algorithm, a gradient-based optimizer, demonstrated competitive performance with an average accuracy of 96.87% across the three subsets (training: 96.98%, validation: 95.56%, testing: 98.07%). While its average accuracy is slightly lower than QIFPNN, FPNN, and BANN, SGD showed remarkably high test accuracy, surpassing both QIFPNN and BANN. This suggests that SGD has strong adaptability to unseen data and can perform well in various practical applications.

While PSOINN shows improvement with an average accuracy of 89.26%, it still lags significantly behind the other algorithms. This limitation highlights the need for further development and optimization of this model to compete with the performance of QIFPNN, FPNN, BANN, and even SGD. Overall, these results support the conclusion that QIFPNN, FPNN and BANN are effective methods for diabetes detection, with SGD emerging as a promising alternative due to its balance of performance and efficiency, and QIFPNN being the most consistent performer overall.

The evaluation results based on the F1 score metric across the three subsets (training, validation, and testing), which reflect the performance of each algorithm, are presented in Table 6 and Figure 8 through Figure 10. These results show that QIFPNN also achieved the highest F1 score on the training subset.

**Tab. 6. Recapitulation of mean F1-score of the training, validation, and test subsets for the ten folds of diabetes**

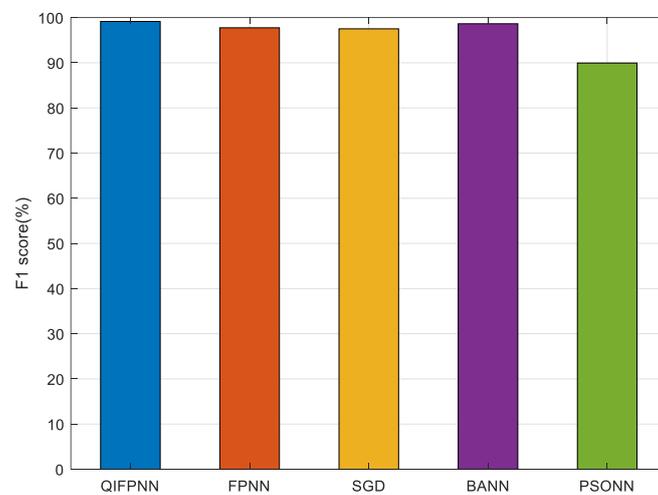
Algorithm	Average F1-score of training subset (%)	Average F1-score of validation subset (%)	Average F1-score of test subset (%)	Mean F1-score of training, validation, and test subsets (%)
QIFPNN	99.14	98.17	97.57	98.30
FPNN	97.75	97.87	97.90	97.84
BANN	98.64	97.47	97.63	97.91
PSOINN	89.95	91.95	92.44	91.45
SGD	97.52	96.33	98.40	97.42

The F1 score results on the training subset show that QIFPNN excels in accuracy and achieves a balance between precision and recall, with an F1 score of 99.14%. This figure reflects QIFPNN's exceptional ability

to detect positive diabetes cases, indicating a low error rate in both positive and negative predictions. FPNN and BANN also demonstrated strong F1 scores of 97.75% and 98.64%, respectively. These performances indicate that both algorithms effectively detect diabetes, although they do not outperform QIFPNN. PSONN, with an F1 score of 89.95%, shows significant room for improvement, particularly in minimizing false positives and negatives.

When SGD was included in the training process, the F1 score for SGD on the training subset reached 97.52%, which, although lower than QIFPNN, is still highly competitive. This indicates that SGD can effectively balance precision and recall, although not as efficiently as QIFPNN or BANN. The lower F1 score observed in SGD compared to QIFPNN suggests that the gradient-based approach may face challenges in capturing the complexity of the diabetes dataset, especially when compared to metaheuristic-based models such as QIFPNN. Nevertheless, the performance of SGD remains remarkable, as it shows good predictive results, making it a viable option for diabetes detection.

The high F1 scores achieved by QIFPNN and BANN highlight their ability to provide more accurate predictions in the context of diabetes detection. SGD, although slightly behind, offers a robust alternative with its ability to provide efficient and stable training results. It emphasizes the importance of relying on accuracy and F1 score metrics to get a complete picture of model performance.



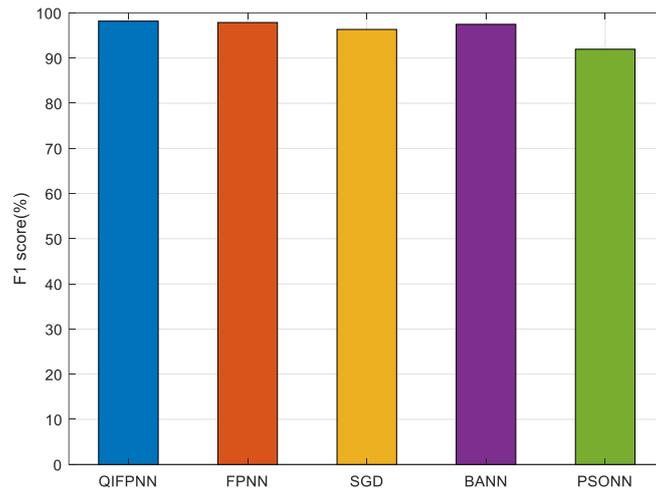
**Fig. 8. Average 10-fold F1-score of the training subset of diabetes**

The F1 score results on the validation subset show that QIFPNN maintains its strong performance with an F1 score of 98.17%. This score demonstrates QIFPNN's ability to achieve a balanced trade-off between precision and recall, highlighting its effectiveness in detecting positive diabetes cases. FPNN and BANN also achieved impressive F1 scores of 97.87% and 97.47%, respectively, suggesting that these algorithms are similarly practical in the validation context, although QIFPNN retains its superior edge.

SGD, although a more traditional optimization method, demonstrated competitive performance with an F1 score of 96.33%. Although it lags slightly behind QIFPNN, FPNN, and BANN, this result underscores SGD's strong ability to produce stable predictions. Its F1 score shows that SGD can reasonably balance precision and recall, especially in the context of unseen data. Although SGD was not as effective as QIFPNN, it performed well enough to provide a solid baseline and demonstrated greater adaptability to real-world applications.

PSONN, with an F1 score of 91.95%, shows a significant improvement over its performance on the training subset, but still has room for further optimization. This result indicates that PSONN can adapt reasonably well to unseen data, but needs improvements to match the performance of the other models.

Overall, the F1 score results on the validation subset underscore the reliability of QIFPNN, FPNN and BANN in diabetes detection, with QIFPNN remaining the most robust model. The inclusion of SGD provides a practical alternative and shows that competitive results can still be achieved using a traditional optimization technique, especially for applications requiring faster training times and lower computational costs.



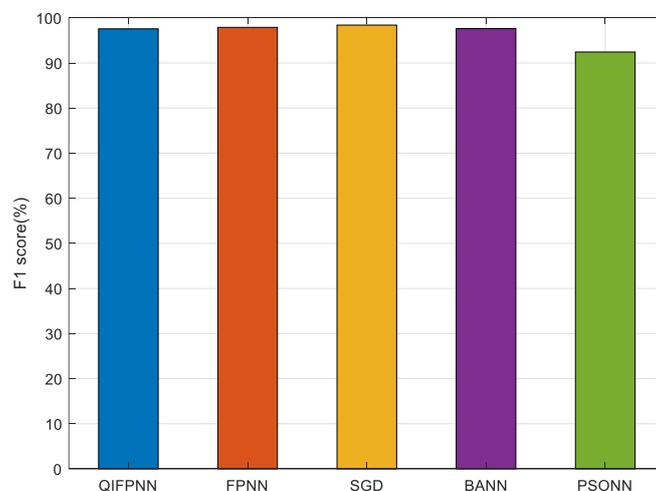
**Fig. 9. Average 10-fold F1-score of the validation subset of diabetes**

The F1 score results on the test subset show that QIFPNN has maintained its excellent performance with an F1 score of 97.57%. This indicates that QIFPNN continues to effectively balance precision and recall in the detection of diabetes on unseen data. FPNN and BANN also achieved satisfactory results, with F1 scores of 97.90% and 97.63%, respectively. These results confirm that both algorithms are practical and can perform well on unknown data.

SGD outperforms QIFPNN, FPNN, and BANN on the test subset with an F1 score of 98.40%. It demonstrates that SGD, despite its more traditional gradient-based approach, can achieve competitive performance in generalizing to unseen data. The improvement in SGD's F1 score highlights its efficiency in fine-tuning model parameters and ensuring a balance between precision and recall. Although slightly lower in training and validation, SGD's strong performance on the test subset suggests that it can effectively handle the variability inherent in real-world data, making it a highly valuable option for use in practical applications.

PSONN, with an F1 score of 92.44%, shows an improvement over the previous subsets, although it still lags behind the performance of the other models. This score suggests that while PSONN can identify the most favorable cases, there is potential for further improvement in accuracy and prediction balance.

Overall, the F1 score results across the three subsets emphasize the reliability of QIFPNN, FPNN and BANN in diabetes detection. SGD, with its impressive performance on the test subset, emerges as a strong competitor offering a balance between accuracy and adaptability. QIFPNN consistently shows the most substantial performance across all subsets, making it the leading model in this context. However, SGD's performance suggests that it could be a viable alternative, especially in practical applications where adaptability to diverse data is critical.



**Fig. 10. Average 10-fold F1-score of the test subset of diabetes**

The average F1 score results show that QIFPNN achieves the best performance with an average score of 98.30%. This suggests that QIFPNN excels in accuracy and maintains an excellent balance between precision and recall across all subsets. This ability is critical in the context of diabetes detection, where minimizing both false positives and false negatives is essential for accurate diagnosis. FPNN and BANN also show excellent average F1 scores of 97.84% and 97.91%, respectively. These two algorithms also maintain strong performance in diabetes detection, making them viable alternatives in this field. PSOINN, with an average F1 score of 91.45%, shows a significant improvement compared to its performance in other subsets, but still needs further development to reach performance levels comparable to the other models.

SGD, with an average F1 score of 98.40%, proved to be a competitive model. Despite its traditional gradient-based optimization approach, SGD maintains a high level of performance and achieves an impressive balance between precision and recall, slightly outperforming QIFPNN in the test subset. This result indicates that SGD can handle the complexity of diabetes detection while offering strong adaptability, making it a viable option for real-world applications.

Overall, the average F1 scores across the three subsets support the conclusion that QIFPNN is the most effective model for diabetes detection, followed by FPNN and BANN. However, the addition of SGD shows that traditional methods still have competitive value, offering a balance between performance and speed. PSOINN, while showing progress, still requires additional research to improve its capabilities. These results underscore the importance of selecting the appropriate algorithm for developing effective diabetes detection systems, with QIFPNN and SGD emerging as top contenders based on their performance and adaptability across all evaluation stages.

Paired samples t-tests were performed on the training, validation, and test subsets to determine whether the observed differences in F1 scores between QIFPNN and other algorithms were statistically significant. A summary of the results is shown in Table 7.

**Tab. 7. Paired T-Test results for F1-score (QIFPNN vs other algorithms)**

Subset	Comparison	Mean Diff	t(df)	p-value	Significance
<b>Training</b>	QIFPNN vs FPNN	1.395	7.060	0.000	Yes
	QIFPNN vs BANN	0.497	2.726	0.023	Yes
	QIFPNN vs PSOINN	9.194	34.514	0.000	Yes
	QIFPNN vs SGD	1.624	5.868	0.000	Yes
<b>Validation</b>	QIFPNN vs FPNN	0.302	0.917	0.383	No
	QIFPNN vs BANN	0.707	5.238	0.001	Yes
	QIFPNN vs PSOINN	6.224	8.896	0.000	Yes
	QIFPNN vs SGD	1.839	4.110	0.003	Yes
<b>Testing</b>	QIFPNN vs FPNN	-0.327	-1.457	0.179	No
	QIFPNN vs BANN	-0.054	-0.238	0.817	No
	QIFPNN vs PSOINN	5.132	18.246	0.000	Yes
	QIFPNN vs SGD	-0.827	-3.925	0.003	Yes

The t-test results on the training subset show that QIFPNN significantly outperforms the other algorithms. Comparisons between QIFPNN and FPNN, QIFPNN and BANN, QIFPNN and PSOINN, and QIFPNN and SGD all yield p-values less than 0.05, indicating significant differences in performance. Specifically, QIFPNN achieved an F1 score of 99.14%, far exceeding FPNN (97.75%), BANN (98.64%), PSOINN (89.95%), and SGD (97.52%). It shows that QIFPNN is the most effective algorithm for detecting diabetes on the training data, with the highest ability to minimize errors in both positive and negative predictions (false positives and false negatives).

On the validation subset, t-test results again show significant differences between QIFPNN, BANN, and PSOINN, with p-values less than 0.05. QIFPNN achieved an F1 score of 98.17%, slightly better than BANN (97.47%) and PSOINN (91.95%). However, when compared to FPNN, the p-value is not significant (p-value = 0.383), indicating that FPNN (97.87%) performed similarly to QIFPNN on the validation subset. However, QIFPNN retains a slight advantage by demonstrating better generalization capabilities on unseen data. This highlights QIFPNN as a more stable and effective model for diabetes detection in practical scenarios.

The test subset T-test results show no significant difference between QIFPNN and FPNN or BANN (p-values = 0.179 and 0.817, respectively), indicating comparable performance between QIFPNN and both

algorithms on the test data. However, comparisons with PSONN and SGD show significant differences ( $p$ -values = 0.000 and 0.003, respectively), with QIFPNN outperforming both. QIFPNN achieved an F1 score of 97.57%, while FPNN (97.90%) and BANN (97.63%) showed similar performance, slightly outperforming QIFPNN in the test subset. In contrast, PSONN (92.44%) showed lower performance, highlighting the need for further model optimization. Although SGD (98.40%) showed slightly better F1 score performance, QIFPNN results remain competitive and highly reliable on unseen data.

Overall, the T-test analysis shows that QIFPNN excels in diabetes detection across all subsets (training, validation, and testing). While FPNN and BANN showed strong performance, QIFPNN consistently provided the highest accuracy and F1 scores, especially in the training and validation subsets, with significant performance differences in most comparisons. PSONN showed lower performance in all subsets, indicating a need for further optimization. Although SGD showed slightly better F1 score results in the test subset, QIFPNN performed excellently in all evaluation phases. These results underscore the importance of selecting QIFPNN as the most reliable model for diabetes detection, with FPNN and BANN as competitive alternatives. PSONN and SGD need further development to reach the same level of performance.

In addition to classification performance, we also evaluated the training efficiency of each algorithm by measuring its average learning time (see Table 8). This metric provides insight into the computational cost required to train each model. The results showed that SGD achieved the shortest training time with an average of 584.50 seconds, demonstrating its computational efficiency. PSONN followed with a training time of 1812.10 seconds, then BANN with 2906.70 seconds, and FPNN with 3135.92 seconds. While QIFPNN achieved the best average accuracy (97.90%) and F1 score (98.30%), it required the longest training time of 4107.89 seconds. These results highlight a trade-off between predictive performance and computational efficiency, where QIFPNN excels in accuracy but requires more resources and time during the training phase. This consideration is important for use in time-constrained or resource-limited environments.

**Tab. 8. Training time**

Algorithm	Training time (seconds)
QIFPNN	4107.89
FPNN	3135.92
BANN	2906.70
PSONN	1812.10
SGD	584.50

The evaluation results show that the QIFPNN algorithm consistently outperforms the accuracy and F1-score of other models across all subsets. With the highest accuracy in the training subset (98.95%) and an average F1 score of 98.30%, QIFPNN demonstrates exceptional ability to detect diabetes. The ability of this model to maintain high performance across all subsets reflects its stability and excellent generalization to new data. T-test results confirm that QIFPNN significantly outperforms other models, especially when compared to PSONN and SGD, which show significant performance gaps. Despite QIFPNN's higher computational cost during training, its superior predictive accuracy makes it the leading model for diabetes detection.

FPNN and BANN also show impressive performance, especially on the validation and testing subsets, with accuracy and F1 scores approaching those of QIFPNN. FPNN achieved the highest F1 score on the test subset (97.90%), while BANN showed a competitive F1 score of 97.63%. The T-test results also show significant differences between QIFPNN and these two models, further highlighting QIFPNN's superior performance at all levels. However, despite slightly lower performance, FPNN and BANN remain solid choices for diabetes detection.

On the other hand, PSONN showed lower performance than the other models, with an average accuracy of 89.26% and an average F1 score of 91.45%. The t-test results indicate significant performance differences between PSONN, QIFPNN, and FPNN. While PSONN improved the test subset, it still lags behind in comparison. The training time analysis further highlights that PSONN is less computationally efficient than the other models, taking significantly more time than SGD but less than QIFPNN. Despite performance improvements, PSONN still needs further optimization in architecture and parameter tuning to compete with the other proven algorithms.

In addition, SGD stands out for its training efficiency, completing the training process in only 584.50 seconds, the fastest among all models. While SGD achieved a competitive accuracy of 96.87% and an F1 score of 97.42% on the test subset, it does not outperform QIFPNN, FPNN, or BANN in terms of prediction accuracy

or F1 score. Nevertheless, SGD offers a good balance between training time and model performance, making it a suitable candidate when computational efficiency is critical.

These results emphasize the importance of selecting the appropriate algorithm based on accuracy and computational efficiency. While QIFPNN shows significant superiority in predictive performance, FPNN and BANN are viable choices. For scenarios where training time is a significant constraint, SGD offers a reasonable trade-off between performance and computational resources.

#### 4. CONCLUSIONS

This study successfully implemented and compared four algorithms for diabetes detection, namely QIFPNN, FPNN, BANN, and PSONN. The evaluation results showed that QIFPNN is the superior algorithm with an average accuracy of 97.90% and an average F1 score of 98.30%. These two metrics indicate the ability of QIFPNN to accurately and efficiently detect diabetes. T-test results confirm that QIFPNN performs significantly better than the other models, especially compared to PSONN, SGD and FPNN in several subsets. In addition, QIFPNN achieved the highest accuracy and F1 score across the three subsets, demonstrating its strong ability to generalize to new data.

FPNN and BANN also performed well, with average accuracies of 97.34% and 97.43%, respectively, and average F1 scores close to QIFPNN. Although these models did not outperform QIFPNN, the t-test results showed that FPNN and BANN were competitive with each other and could be viable alternatives for diabetes detection. FPNN recorded the highest F1 score of 97.90% on the test subset, confirming its effectiveness. However, training time analysis showed that QIFPNN required significantly more time (4107.89 seconds) compared to FPNN (3135.92 seconds) and BANN (2906.70 seconds). This trade-off between computational efficiency and predictive accuracy should be considered when deploying models in resource-constrained environments.

PSONN, while showing lower results with an average accuracy of 89.26% and an average F1 score of 91.45%, still contributes to the analysis of model performance. The t-test results highlighted the significant differences between PSONN, QIFPNN and FPNN. Despite some improvements in the test subset, PSONN requires further optimization in model architecture and parameter tuning. In addition, PSONN was more computationally expensive than SGD and took longer to train (1812.10 seconds). These results emphasize the need for future research to optimize PSONN and improve its performance.

On the other hand, although SGD showed competitive performance with an accuracy of 96.87% and an F1 score of 97.42% on the test subset, it was significantly faster in training (584.50 seconds). Although it did not outperform QIFPNN or FPNN in accuracy or F1 score, SGD offers a promising trade-off between training efficiency and performance. The t-test analysis showed that SGD is computationally efficient and may be a suitable option when computational resources or time constraints are critical.

Overall, the results of this study support the idea that QIFPNN can be relied upon as an effective diabetes detection model, providing the highest performance in terms of accuracy and F1 score. However, the trade-off between performance and computational cost, especially for QIFPNN, should be carefully considered for use in time-critical applications. Although the training time of QIFPNN is higher due to its sophisticated optimization process, this is only a concern during model development, as the final trained model can be efficiently used for real-time prediction in practical applications. The performance of FPNN, BANN, and SGD also highlights the strengths of different algorithms and provides options for different levels of performance and computational efficiency in diabetes detection. These results emphasize the importance of selecting the most appropriate model based on the specific needs of the application, taking into account both performance and computational efficiency.

Based on the results obtained, several suggestions that could be considered for future research are (1) Optimization of PSONN: Although PSONN showed lower performance, efforts can be made to optimize the parameters and architecture of the model to improve its accuracy. Techniques such as hyperparameter tuning and the addition of layers or neurons could be explored; (2) Larger dataset: Using a larger and more diverse dataset can help evaluate the robustness of the model and its ability to generalize to a larger population; (3) Further studies: Conducting further studies that examine other factors that may affect outcomes, such as demographic and clinical variables, to enrich the understanding of diabetes detection using these algorithms. With these steps, it is expected that significant improvements in diabetes detection and management can be achieved, contributing to overall public health.

## Author Contributions

The contributions of each author are described as follows: "Conceptualisation and methodology, Polly, Fanggidae, Ledoh, Amos Pah, and Djahi; implementation of the methodology, Polly, Fanggidae, and Tupen; data validation, Ledoh, Amos Pah, Djahi, and Polly; formal analysis, Polly and Fanggidae; investigation, Polly, Fanggidae, Ledoh, Amos Pah, and Djahi; writing—original draft preparation, Polly; writing—review and editing, Fanggidae, Ledoh, Amos Pah, Djahi, and Tupen; supervision, Polly and Fanggidae."

## Funding

The authors would like to thank the Faculty of Science and Engineering, Universitas Nusa Cendana, for funding this project through the research program No. 13/UN15.18.PPK/SPP/FST/IV/2024.

## Conflicts of Interest

The authors declare there are no conflicts of interest in this work.

## REFERENCES

- Aalimahmoody, N., Bedon, C., Hasanzadeh-Inanlou, N., Hasanzade-Inallu, A., & Nikoo, M. (2021). Bat algorithm-based ann to predict the compressive strength of concrete—a comparative study. *Infrastructures*, 6(6), 1–17. <https://doi.org/10.3390/infrastructures6060080>
- Abed Mohammed, A., Sumari, P., & Attabi, K. (2024). Hybrid K-means and principal component analysis (PCA) for diabetes prediction. *International Journal of Computing and Digital Systems*, 15(1), 1719–1728. <https://doi.org/10.12785/ijcds/1501121>
- Al Bataineh, A., Kaur, D., & Jalali, S. M. J. (2022). Multi-layer perceptron training optimization using nature inspired computing. *IEEE Access*, 10, 36963–36977. <https://doi.org/10.1109/ACCESS.2022.3164669>
- Amma, N. G. B. (2024). En-RfRsK: An ensemble machine learning technique for prognostication of diabetes mellitus. *Egyptian Informatics Journal*, 25, 100441. <https://doi.org/10.1016/j.eij.2024.100441>
- Bhat, S. S., Selvam, V., Ansari, G. A., Ansari, M. D., & Rahman, M. H. (2022). Prevalence and early prediction of diabetes using machine learning in north Kashmir: A case study of district bandipora. *Computational Intelligence and Neuroscience*, 2022(1), 2789760. <https://doi.org/10.1155/2022/2789760>
- Chaves, L., & Marques, G. (2021). Data mining techniques for early diagnosis of diabetes: A comparative study. *Applied Sciences*, 11(5), 2218. <https://doi.org/10.3390/app11052218>
- Chiroma, H., Khan, A., Abubakar, A. I., Saadi, Y., Hamza, M. F., Shuib, L., Gital, A. Y., & Herawan, T. (2016). A new approach for forecasting OPEC petroleum consumption based on neural network train by using flower pollination algorithm. *Applied Soft Computing Journal*, 48, 50–58. <https://doi.org/10.1016/j.asoc.2016.06.038>
- Emon, M. U., Keya, M. S., Kaiser, M. S., Islam, M. A., Tanha, T., & Zulfiker, M. S. (2021). Primary stage of diabetes prediction using machine learning approaches. *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)* (pp. 364–367). IEEE. <https://doi.org/10.1109/ICAIS50930.2021.9395968>
- International Diabetes Federation. (2024). *Diabetes around the world in 2021*. <https://idf.org/about-diabetes/diabetes-facts-figures/>
- Islam, M. M. F., Ferdousi, R., Rahman, S., & Bushra, H. Y. (2020). Likelihood prediction of diabetes at early stage using data mining techniques. In M. Gupta, D. Konar, S. Bhattacharyya, & S. Biswas (Eds.), *Computer Vision and Machine Intelligence in Medical Image Analysis* (Vol. 992, pp. 113–125). Springer Singapore. [https://doi.org/10.1007/978-981-13-8798-2\\_12](https://doi.org/10.1007/978-981-13-8798-2_12)
- Islam, M. S., Minul Alam, M., Ahamed, A., & Ali Meerza, S. I. (2023). Prediction of diabetes at early stage using interpretable machine learning. *SoutheastCon 2023* (pp. 261–265). IEEE. <https://doi.org/10.1109/SoutheastCon51012.2023.10115152>
- Karpiński, R. (2022). Knee joint osteoarthritis diagnosis based on selected acoustic signal discriminants using machine learning. *Applied Computer Science*, 18(2), 71–85. <https://doi.org/10.35784/acs-2022-14>
- Kujawska, J., Kulisz, M., Oleszczuk, P., & Cel, W. (2022). Machine learning methods to forecast the concentration of PM10 in Lublin, Poland. *Energies*, 15(17), 6428. <https://doi.org/10.3390/en15176428>
- Kulisz, M., Kujawska, J., Przysucha, B., & Cel, W. (2021). Forecasting water quality index in groundwater using artificial neural network. *Energies*, 14(18), 5875. <https://doi.org/10.3390/en14185875>
- Lin, H., Chen, Z., Wu, L., Lin, P., & Cheng, S. (2015). On-line monitoring and fault diagnosis of PV array based on BP neural network optimised by genetic algorithm. In B. A & Z. X (Eds.), *Multi-disciplinary Trends in Artificial Intelligence* (Vol. 9426, pp. 102–112). Springer, Cham. [https://doi.org/10.1007/978-3-319-26181-2\\_10](https://doi.org/10.1007/978-3-319-26181-2_10)
- Machrowska, A., Karpiński, R., Maciejewski, M., Jonak, J., & Krakowski, P. (2024a). Application of eemd-dfa algorithms and ann classification for detection of knee osteoarthritis using vibroarthrography. *Applied Computer Science*, 20(2), 90–108. <https://doi.org/10.35784/acs-2024-18>
- Machrowska, A., Karpiński, R., Maciejewski, M., Jonak, J., Krakowski, P., & Syta, A. (2024b). Application of recurrence quantification analysis in the detection of osteoarthritis of the knee with the use of vibroarthrography. *Advances in Science and Technology Research Journal*, 18(5), 19–31. <https://doi.org/10.12913/22998624/189512>
- Mahesh, T. R., Vinoth, K. V., Dhilip, K. V., Geman, O., Margala, M., & Guduri, M. (2023). The stratified K-folds cross-validation and class-balancing methods with high-performance ensemble classifiers for breast cancer classification. *Healthcare Analytics*, 4, 100247. <https://doi.org/10.1016/j.health.2023.100247>

- Mediakom, R. (2024). *Saatnya Mengatur Si Manis*. Sehat Negeriku - Biro Komunikasi & Pelayanan Publik Kementerian Kesehatan RI. <https://sehatnegeriku.kemkes.go.id/baca/blog/20240110/5344736/saatnya-mengatur-si-manis/>
- Nissar, I., Mir, W. A., Shaikh, T. A., Areen, T., Kashif, M., Khiani, S., & Hussain, A. (2024). An intelligent healthcare system for automated diabetes diagnosis and prediction using machine learning. *Procedia Computer Science*, 235, 2476–2485. <https://doi.org/10.1016/j.procs.2024.04.233>
- Oladimeji, O. O., Oladimeji, A., & Oladimeji, O. (2024). Classification models for likelihood prediction of diabetes at early stage using feature selection. *Applied Computing and Informatics*, 20(3/4), 279–286. <https://doi.org/10.1108/ACI-01-2021-0022>
- Polly, Y. T. (2022). *Model pembelajaran quadratic interpolation flower pollination neural network studi kasus identifikasi penyakit babi [Gajah Mada]*. <http://etd.repository.ugm.ac.id/penelitian/detail/218587>
- Polly, Y. T., Hartati, S., -, S., & Sumiarto, B. (2023). A novel approach to multi-layer-perceptron training using quadratic interpolation flower pollination neural network on non-binary datasets. *International Journal of Advanced Computer Science and Applications*, 14(6), 497–504. <https://doi.org/10.14569/IJACSA.2023.0140653>
- Polly, Y. T., Hartati, S., Suprpto, & Sumiarto, B. (2021). Modified flower pollination algorithm for disease identification in swine. *International Journal of Intelligent Engineering and Systems*, 14(6), 616–628. <https://doi.org/10.22266/ijies2021.1231.55>
- Prusty, S., Patnaik, S., & Dash, S. K. (2022). SKCV: Stratified K-fold cross-validation on ML classifiers for predicting cervical cancer. *Frontiers in Nanotechnology*, 4. <https://doi.org/10.3389/fnano.2022.972421>
- Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. *ArXiv, abs/1811.12808*. <https://doi.org/10.48550/arXiv.1811.12808>
- Shaik, R., & Siddique, I. (2024). Novel multi-modal obstruction module for diabetes mellitus classification using explainable machine learning. *Applied Computer Science*, 20(4), 39–62. <https://doi.org/10.35784/acs-2024-39>
- UCI. (2020). *Early Stage Diabetes Risk Prediction [Dataset]*. UCI Machine Learning Repository. <https://doi.org/10.24432/C5VG8H>
- Yang, X.-S. (2012). Flower pollination algorithm for global optimization. In J. Durand-Lose & N. Jonoska (Eds.), *Unconventional Computation and Natural Computation* (Vol. 7445, pp. 240–249). Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-642-32894-7\\_27](https://doi.org/10.1007/978-3-642-32894-7_27)
- Yang, X.-S. (2014). *Nature-Inspired Optimisation Algorithms*. Elsevier Inc. <https://doi.org/10.1016/B978-0-12-416743-8.00001-4>
- Yang, X.-S. (2020). *Nature-Inspired Optimisation Algorithms* (Second). Academic.