Accredited SINTA 2 Ranking
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online at: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 9 No. 1 (2025) 146 - 152 e-ISSN: 2580-0760

Hybrid Gradient Descent Grey Wolf Optimizer for Machine Learning Performance Enhancement

Sri Rossa Aisyah Puteri Baharie[1*], Sugiyarto Surono[2], Aris Thobirin[3]
[1, 2, 3] Matematika, Sains dan Teknologi Terapan, Universitas Ahmad Dahlan, Yogyakarta, Indonesia.
[1]puteri.baharie10@gmail.com, [2]sugiyarto@math.uad.ac.id, [3]aris.thobi@math.uad.ac.id

Abstract

Advancements in machine learning have enabled the development of more accurate and efficient health prediction models. This study aims to improve diabetes prediction performance using the Support Vector Machine (SVM) model optimized with the Hybrid Gradient Descent Gray Wolf Optimizer (HGD-GWO) method. SVM is a robust machine learning algorithm for classification and regression. Still, its performance depends significantly on selecting appropriate hyperparameters such as regularization (C), kernel coefficient (γ), and polynomial kernel degree (d). The HGD-GWO method synergizes Gradient Descent for local optimization and Gray Wolf Optimizer for global solution exploration. Using the Pima Indians Diabetes dataset, the process includes normalization, hyperparameter optimization, data division, and performance evaluation using accuracy, precision, recall, and F1-score metrics. The optimized SVM achieved an accuracy of 81.17%, with precision, recall, and F1-score values of 75.00%, 57.45%, and 65.06%, respectively, at a data ratio of 80%:20%. These findings highlight the potential of HGD-GWO in enhancing predictive models, particularly for early diabetes detection.

Keywords: *Hybrid Gradient Descent Grey Wolf Optimizer; Hyperparameter Optimization; Diabetes Prediction; Machine Learning; Support Vector Machine (SVM)*

How to Cite: S. R. A. Puteri Baharie, Sugiyarto Surono, and Aris Thobirin, "Hybrid Gradient Descent Grey Wolf Optimizer for Machine Learning Performance Enhancement", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 9, no. 1, pp. 146 - 152, Feb. 2025.
DOI: https://doi.org/10.29207/resti.v9i1.6203

1. Introduction

The development of more accurate and efficient prediction models, especially for complex data analysis, has been made possible by advances in science and technology. One area of artificial intelligence known as machine learning allows computers to learn patterns in data without the need for special programs [1]. To maximize the distance between the separating hyperplane and the closest data from each class, Support Vector Machine (SVM) is an ideal margin-based classification method [2]. The advantage of SVMs lies in their ability to use kernel tricks, which allow mapping data to larger dimensions without the need for explicit computation, allowing non-linear data separation with high efficiency. As a result, SVM is often used to solve prediction problems in many fields, such as medical analysis [3].

However, the choice of hyperparameters, such as regulation parameters (C), kernel coefficients (γ), and degree (d) for polynomial kernels, greatly affects the performance of SVM. Hyperparameters determine the generalization and complexity of the model. Failure to optimize it can lead to overfitting or underfitting. Therefore, the main challenge in processing complex data is hyperparameter optimization [4].

Previous studies have used k-Nearest Neighbor (KNN) and Naïve Bayes algorithms to predict diabetes on the Pima Indians Diabetes dataset, and research findings show that Naïve Bayes is very accurate [5]. Nevertheless, this technique can still be improved through the use of more sophisticated machine learning techniques and more efficient hyperparameter optimization.

To overcome this, optimization techniques such as Gradient Descent (GD) can be used. The GD

Received: 01-12-2024 | Accepted: 30-01-2025 | Published Online: 16-02-2025

optimization method is commonly used to minimize the loss function in machine learning by iteratively updating parameters based on the direction of the negative gradient. Thus, GD helps the model converge more quickly and stably [6-7]. However, this technique often experiences slow convergence because it is limited to local data [8]. Consequently, to solve this problem, you can leverage the advantages of metaheuristic optimization methods such as Gray Wolf Optimizer (GWO). This method is based on the hunting habits of gray wolf packs, which can be explored widely and find the best solution [9-10].

The Hybrid HGD-GWO approach is expected to overcome the limitations of each method by combining the global exploration advantages of GWO and the local convergence efficiency of GD [11]. HGD-GWO uses initial global exploration of GWO to explore a wide parameter space, then performs local refinement using GD to reach an optimal solution with faster convergence [11]. This method has shown that it has great potential to improve the accuracy of predictor models in various applications [12].

This research aims to improve the accuracy of diabetes prediction on the Pima Indians Diabetes dataset by integrating the HGD-GWO method in SVM hyperparameter optimization. By leveraging GWO's global exploration capabilities and GD's convergence efficiency, this research has the potential to aid the early detection of diabetes and help develop better machine teaching algorithms.

2. Research Methods

This research was conducted with an experimental approach as shown in Figure 1, using the Pima Indians Diabetes dataset which is publicly available https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database?resource=download.

2.1 Dataset

The dataset used is Pima Indians Diabetes which consists of 768 samples with 8 numerical features. These features include pregnancies (number of pregnancies), glucose (glucose levels), blood pressure (diastolic blood pressure), skin thickness (triceps skinfold thickness), insulin (insulin level), BMI (body mass index), diabetes pedigree function (diabetes genealogy function), and age.

The outcome label indicates diabetes status, with a one (1) for positive diabetes and a zero (0) for negative diabetes. This dataset is sourced from Kaggle and is used to predict a person's likelihood of suffering from diabetes based on their medical characteristics.

2.2 Preprocessing Data

The preprocessing stage begins with identifying and addressing missing values in the dataset. These missing values will be filled using the median of the corresponding column, which helps maintain the stability of the data distribution. The median is selected for this purpose because it effectively reduces distortion that may arise from outliers while keeping the data distribution stable and unbiased [13]. This step is crucial for improving data quality and ensuring the reliability of analysis results, as missing values can negatively impact model performance during the training process. Proper preprocessing is essential for developing machine learning models that are both accurate and capable of generalizing well [14].

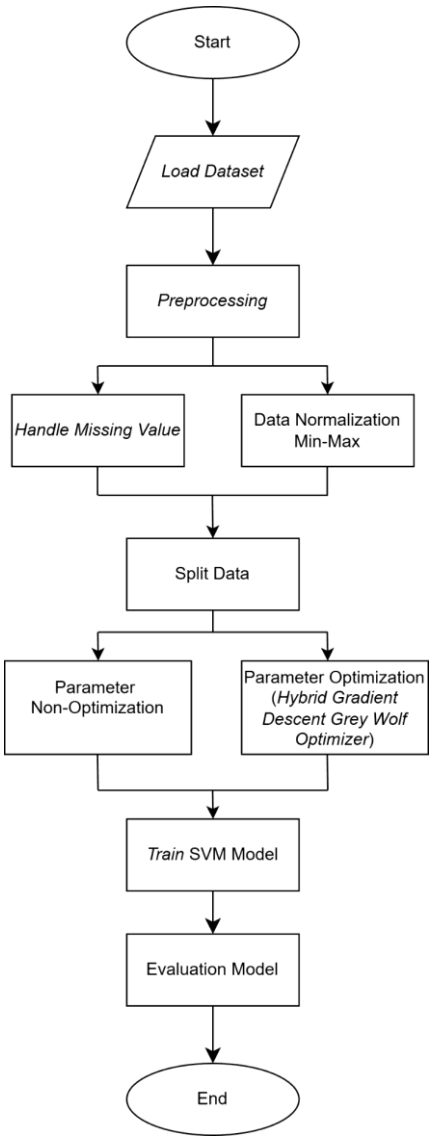

Figure 1. Data Preprocessing Flow

Next, all numerical features are normalized using the Min-Max scaling method to ensure that all features fall within the same range of [0, 1] [15]. This normalization is important to minimize noise and irrelevant data, thereby enhancing performance in the classification process. Additionally, this approach changes the data without losing any information, simplifying the overall data processing [16]. Equation 1 obtains values from normalization results using min-max normalization [17].

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

x' is the normalized data, x is the initial data, $min(x)$ is the minimum value and $max(x)$ is the maximum value. This process aims to reduce the risk of bias due to imbalanced data scales, which can impact model performance during training. A good implementation of normalization allows the model to learn optimally, reduces the risk of overfitting, and increases generalization ability on new data.

2.3 Split Data

After the normalization stage, the dataset is divided into independent variables (X) and dependent variables (Y). The independent variables consist of eight numerical features: pregnancy, glucose, blood pressure, skin thickness, insulin, BMI, diabetes pedigree function, and age. The dependent variable indicates diabetes status, with a value of 1 representing positive diabetes and 0 representing negative diabetes. Separating the independent and dependent variables is a crucial step in machine learning modeling, as it helps prevent data confusion during training and evaluation [18].

The dataset is divided into two parts: training data, which is used to build the model, and testing data, which is used to evaluate the model's performance on previously unseen data [19]. This division is conducted using various training and testing ratios, such as 80%:20%, 70%:30%, and even 10%:90%, to ensure the model's stability across different data distributions. A larger training ratio, like 80%:20%, allows the model to recognize patterns more effectively, though it may reduce its validation on the test data. Conversely, a smaller training ratio, such as 10%:90%, challenges the model to learn patterns from a limited dataset. The primary goal of this division is to ensure the model can generalize well, leading to accurate and reliable predictions. [19].

2.4 Parameter Optimization

To enhance the performance of machine learning models, parameter optimization is a crucial step in the training process. This research focuses on optimizing three key hyperparameters for SVM: regularization (C), kernel coefficient (γ), and the $degree$ (d) of the polynomial kernel [18]. The parameter C is essential for controlling the margin of the hyperplane, which helps prevent overfitting. The kernel coefficient indicates the influence of each data point within the radial basis function (RBF) kernel, while the $degree$ (d) captures the complexity of non-linear data, particularly in polynomial kernels [20].

The optimization process was carried out using the HGD-GWO method, which combines global exploration from GWO with local convergence from GD [21]. GWO exploits gray wolf hunting behavior to iteratively update optimal population positions. Each position is updated based on the average of alpha, beta, and delta positions as formulated in Equation 2 [22].

$$\vec{X}(t+1) = \frac{\vec{X}_\alpha + \vec{X}_\beta + \vec{X}_\delta}{3} \quad (2)$$

Following the global exploration conducted by GWO, the optimal solution (alpha) is refined using GD to minimize the loss function based on negative gradients, as described in Equation 3 [7].

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta J(\theta) \quad (3)$$

Using 20 iterations, with five wolves and a learning rate of $\eta = 0.01$, this algorithm seeks to find the optimal combination of parameters C, γ, and d. These parameters are evaluated based on prediction accuracy on test data, serving as the fitness function.

The HGD-GWO algorithm begins with an initial population that represents various combinations of parameters. Each individual is refined through interactions among alpha, beta, and delta individuals, followed by local refinement using gradient descent. The resulting optimal parameter combinations differ depending on the training data ratio, as shown in Table 1.

These optimal parameters form the foundation for developing an SVM model with a polynomial kernel, which aims to achieve the best performance in detecting diabetes, assessed through metrics such as accuracy, precision, recall, and F1-score.

2.5 Implementation of the SVM model

After determining the optimal parameters C, γ, and d through an optimization process, these values are applied to the SVM model using a polynomial kernel. The polynomial kernel transforms the data into a high-dimensional feature space, enabling the model to identify the optimal hyperplane that separates non-linear data classes [23]. The polynomial kernel formula utilized in this research is shown in Equation 4.

$$K(x, x') = (x \cdot x' + c)^d \quad (4)$$

where x and $x'$are two data vectors, c is the bias, and d is the kernel $degree$.

The parameter C is used to control the balance between the maximum margin and the tolerance for misclassification. The formulation of the SVM objective function with regularization (C) can be seen in Equation 5 [18].

$$min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i \quad (5)$$

with the condition $y_i(w^T x_i + b) \geq 1 - \xi_i$, where ξ_iIs a slack variable that allows the model to accept some classification errors without sacrificing the margin [19].

The parameter γ regulates the influence of each data point in the kernel which aims to determine the shape of the decision boundary in the higher data space. Mathematically, the formula for γ can be seen in Equation 6.

$$\gamma = \frac{1}{number\ of\ features} \quad (6)$$

Meanwhile, $degree\ (d)$ is used to capture the complexity of non-linear data, especially in polynomial kernels. Mathematically, the degree is set at a minimum of 2 ($d \geq 2$).

This parameter optimization process was carried out using the HGD-GWO method, as shown in Table 1.

2.6 Model Evaluation

The confusion matrix is a tool used to assess the performance of a classification model by illustrating the relationship between the model's predictions and the actual values (ground truth). It shows the number of correct and incorrect predictions made by the classification model for each class [24].

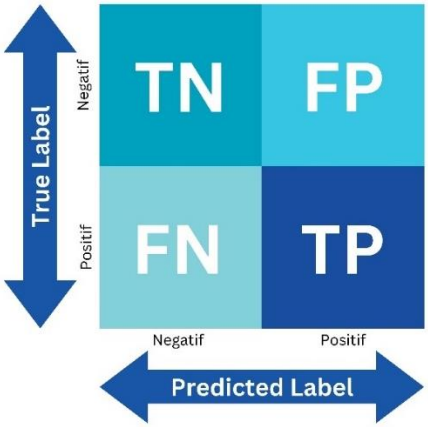

Figure 2. Confusion Matrix

The four main components of model evaluation are illustrated in Figure 2. True Positive (TP) refers to the number of positive cases that the model correctly predicts, while True Negative (TN) indicates the number of negative cases that the model accurately predicts. A False Positive (FP) occurs when the model incorrectly classifies negative cases as positive, and a False Negative (FN) happens when the model fails to identify positive cases [25]. To assess the performance of the classification model, we can use a confusion matrix. This matrix enables us to calculate several important metrics: accuracy, precision, recall, and F1 score. As outlined in Equation 7, accuracy is defined as the proportion of correct predictions made by the model out of all predictions. Precision measures the correctness of positive predictions, while recall evaluates the model's capability to identify all positive cases, as indicated in Equation 8. Equation 9 further explains how recall quantifies the model's effectiveness in detecting all positive instances. Finally, the F1 score, presented in Equation 10, represents the harmonic average of precision and recall [26].

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

$$Presisi = \frac{TP}{TP + FP} \quad (8)$$

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (10)$$

3. Results and Discussions

This research employs the HGD-GWO method to optimize the parameters of a SVM. The study evaluates the model's performance in predicting diabetes using the Pima Indians Diabetes dataset. Performance metrics, including accuracy, precision, recall, and F1-score, are used to assess the model. Additionally, confusion matrix analysis is performed based on the optimal proportion of training data. The primary objective of this process is to enhance the model's performance in predicting diabetes.

3.1 Optimization of SVM Parameters with HGD-GWO

Table 1 shows the results of optimizing parameters C, γ, and d on various ratios of training and testing data.

Table 1. Optimal Parameters

Training Percentage	C	γ	d
80%:20%	2.24	0.55	1.00
70%:30%	5.43	0.40	2.00
60%:40%	4.57	0.53	2.00
50%:50%	2.62	0.78	3.00
40%:60%	7.54	0.79	2.00
30%:70%	2.29	0.38	2.00
20%:80%	3.59	0.80	3.00
10%:90%	2.26	0.99	3.00
Average	3.82	0.65	2.25

Changes in the values of parameters C, γ, and d at various training and testing data ratios demonstrate the impact of data distribution on optimal parameter selection. For instance, with a data ratio of 80% for training and 20% for testing, the optimal parameters are $C = 2.24$, $\gamma = 0.55$, and $d = 1.00$. In contrast, at a 40% training and 60% testing ratio, the values shift to $C = 7.54$, $\gamma = 0.79$, and $d = 2.00$. This indicates that a larger training data ratio tends to result in smaller values for C and d, reflecting the algorithm's stability when handling more training data.

3.2 Model Evaluation

The performance evaluation results of the SVM model, optimized using HGD-GWO, are shown in Table 2. The evaluation included measuring accuracy, precision, recall, and F1-score for each ratio of training and testing data.

Table 2. Evaluation Results After Optimization

Training Percentage	Accuracy	Precision	Recall	F1-Score
80%:20%	81.17%	75.00%	57.45%	65.06%
70%:30%	77.06%	70.59%	48.65%	57.60%
60%:40%	77.60%	76.56%	47.57%	58.68%
50%:50%	77.86%	78.05%	48.86%	60.09%
40%:60%	77.87%	75.21%	54.66%	63.31%
30%:70%	76.77%	77.05%	49.21%	60.06%
20%:80%	76.42%	75.00%	50.68%	60.49%
10%:90%	75.87%	68.75%	58.37%	63.13%
Average	77.58%	74.53%	51.93%	61.05%

Table 2 shows that the model achieves the highest performance at a data ratio of 80%:20% with accuracy of 81.17%, precision of 75.00%, recall of 57.45%, and F1-score of 65.06%. These results show that the larger

the training data used, the better the model's ability to learn data patterns, resulting in more optimal performance. This is to the statement that the more training data, the better the model is at understanding data distribution without relying on testing data.

However, the smaller training data ratio still provides competitive performance. For example, at a data ratio of 40%:60%, the model achieved 77.87% accuracy with 75.21% precision, 54.66% recall, and 63.31% F1-score. This shows that the HGD-GWO algorithm is still able to find optimal parameters that support model performance, even when the amount of training data is limited.

Overall, the average results show that the model achieves 77.58% accuracy, 74.53% precision, 51.93% recall, and 61.05% F1-score across all data ratios. This pattern indicates that the model can perform consistently with different data distributions, which is an advantage of parameter optimization using HGD-GWO.

3.3 Comparison of Evaluation Before and After Optimization

The evaluation results of the SVM model before and after optimization with the HGD-GWO algorithm show significant improvements in various evaluation metrics. Table 3 presents the evaluation results before optimization, while Table 2 presents the evaluation results after optimization. From these two tables, it can be observed that optimizing parameters C, γ, and d succeeded in increasing accuracy, precision, recall, and F1-score at all ratios of training and testing data.

Table 3. Evaluation Results Before Optimization

Training Percentage	Accuracy	Precision	Recall	F1-Score
80%:20%	78.57%	69.44%	53.19%	60.24%
70%:30%	76.62%	70.00%	47.30%	56.45%
60%:40%	76.62%	73.85%	46.60%	57.14%
50%:50%	76.04%	71.91%	48.86%	58.18%
40%:60%	77.22%	75.93%	50.93%	60.97%
30%:70%	76.02%	73.13%	51.31%	60.31%
20%:80%	75.61%	72.55%	50.68%	59.68%
10%:90%	73.99%	65.26%	56.73%	60.70%
Average	76.34%	71.51%	50.70%	59.21%

At a data ratio of 80%:20%, accuracy increased from 78.57% to 81.17%, while precision increased from 69.44% to 75.00%. Recall also increased from 53.19% to 57.45%, reflecting the model's ability to detect positive cases of diabetes. The F1-score increases from 60.24% to 65.06%, indicating a better balance between precision and recall after optimization.

The comparison graph of the results before and after optimization shows a consistent pattern, where the increase in model performance occurs at all data ratios. This indicates that the HGD-GWO algorithm is not only adaptive to changes in data distribution but also provides advantages in balancing sensitivity (recall) and prediction accuracy (precision).

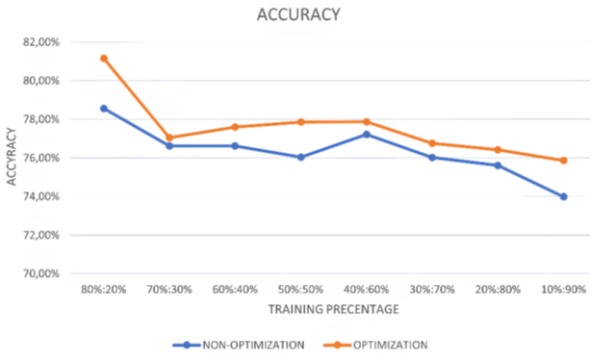

Figure 3. Accuracy Comparison

The graph in Figure 3 illustrates that the model's accuracy after optimization (represented by the orange line) surpasses its accuracy prior to optimization (shown by the blue line) in nearly all training and testing data ratios. The most significant improvement was observed at a data ratio of 80% training to 20% testing, indicating that the HGD-GWO algorithm effectively enhanced the model's ability to learn data patterns.

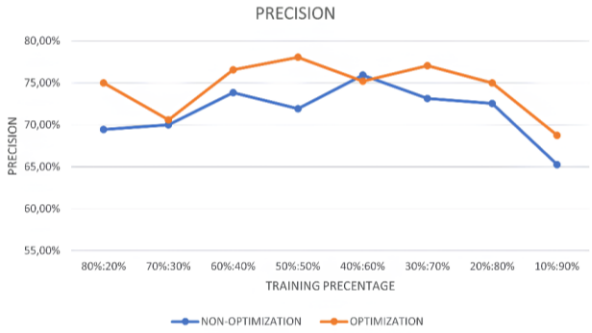

Figure 4. Precision Comparison

Figure 4 shows an increase in precision in all ratios of training and testing data. For example, at a data ratio of 80%:20%, precision increases from 69.44% (blue line) to 75.00% (orange line), which means that the model after optimization is more effective in identifying positive samples, with an error rate in classification lower positive than before optimization.

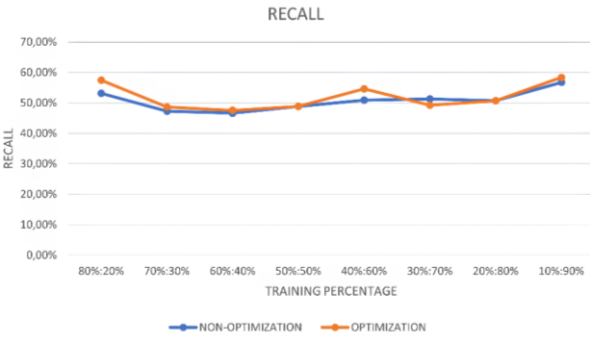

Figure 5. Recall Comparison

Figure 5 shows the increase in recall in almost all ratios of training and testing data after optimization. For example, at a data ratio of 80%:20%, recall increases from 53.19% (blue line) to 57.45% (orange line).

Although this improvement is not as large as in other metrics, the consistency of this improvement indicates that the optimized model can detect positive samples better, thereby increasing the sensitivity of the model.

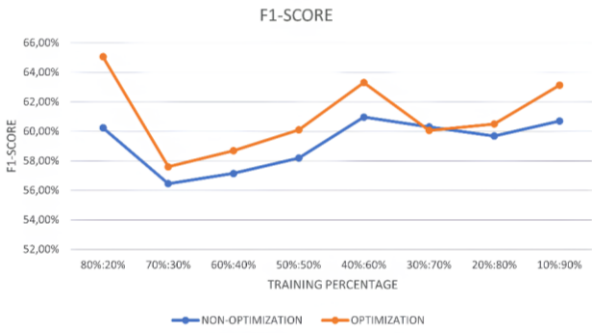

Figure 6. F1-score Comparison

Figure 6 illustrates the improvement in the F1-score, indicating a better balance between precision and recall after optimization. For instance, with a data ratio of 80%:20%, the F1-score increases from 60.24% (shown by the blue line) to 65.06% (represented by the orange line). This demonstrates that the optimized model exhibits a more stable and balanced performance in data classification.

The results of this evaluation confirm that the HGD-GWO algorithm effectively enhances the performance of the SVM model across various metrics, including accuracy, precision, recall, and the F1-score. This improvement highlights the algorithm's capability to optimize model parameters, thereby maximizing generalization abilities. Furthermore, the success of the HGD-GWO algorithm in enhancing model performance on medical datasets, such as those related to diabetes, opens up significant opportunities for its application in other classification problems.

3.4 Confusion Matrix

For the best data rasio (80%:20%), the SVM model produces a confusion matrix that shows the distribution of predictions for the positive (diabetes) and negative (no-diabetes) classes. This matrix provides a visualization of the model's performance in classifying data accurately.

Based on Figure 7, the model achieved a TN count of 98, meaning it correctly predicted 98 negative cases (no diabetes). In contrast, the model had a FP count of 9, indicating that 9 negative cases were incorrectly classified as positive (diabetes). For the positive class, the model reported a FN count of 20, which represents the number of positive cases (diabetes) that were mistakenly classified as negative (no diabetes). Lastly, the model achieved a TP count of 27, showing that it correctly predicted 27 positive cases (diabetes).

The results indicate that the model demonstrates strong classification capabilities, particularly in identifying the negative class (no diabetes). Confusion matrices are valuable tools for understanding model performance across different prediction scenarios. This is especially important in medical applications, such as early diabetes detection, where sensitivity to positive cases is a critical factor.

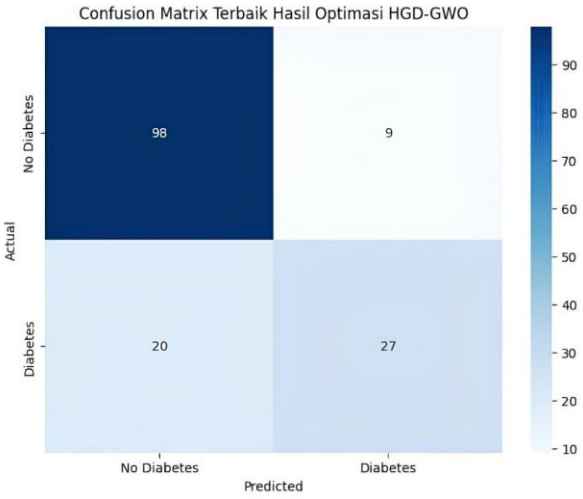

Figure 7. Confusion Matrix

4. Conclusions

This research demonstrates the effectiveness of the HGD-GWO optimization method in enhancing the performance of the SVM model for diabetes prediction. By integrating the global exploration capability of the GWO algorithm with the local exploitation efficiency of GD, this approach generates optimal parameters that quantitatively improve model performance. The best results were achieved with a training data ratio of 80%:20%, yielding an accuracy of 81.17%, a precision of 75.00%, a recall of 57.45%, and an F1-score of 65.06%. This represents an improvement over the pre-optimization results, which showed 78.57% accuracy, 69.44% precision, 53.19% recall, and an F1-score of 60.24%. Specifically, there were increases of 3.3% in accuracy, 5.6% in precision, 4.3% in recall, and 4.8% in the F1-score. Consistent results were observed with other training data ratios, such as 70%:30% and 60%:40%, with average increases in accuracy of 2.7% and 3.0%, respectively, compared to the pre-optimization results. The HGD-GWO approach also demonstrates advantages in balancing predictive ability (precision) with model sensitivity (recall), which is crucial for medical applications, particularly for the early detection of diabetes. By optimizing parameters such as C, γ, and d, the model captures more complex data patterns without overfitting, as evidenced by performance improvements across all evaluation metrics. Future research could examine the application of the HGD-GWO method to high-dimensional datasets or those with unbalanced distributions. Additionally, this method can be integrated with modern techniques such as ensemble learning or deep learning to address challenges in other medical prediction scenarios. Overall, the results of this study indicate that the HGD-GWO method is an effective and promising approach

for enhancing the performance of classification models on complex medical datasets.

References

[1] M. Kumar and Bhawna, *Introduction to Machine Learning*, vol. 1169. 2024. doi: 10.1007/978-981-97-5624-7_2.

[2] L. D. Avendaño-Valencia and S. D. Fassois, "Natural vibration response based damage detection for an operating wind turbine via Random Coefficient Linear Parameter Varying AR modelling," *J. Phys. Conf. Ser.*, vol. 628, no. 1, pp. 273–297, 2015, doi: 10.1088/1742-6596/628/1/012073.

[3] S. Surono, T. Nursofiyani, and A. E. Haryati, "Optimization of Fuzzy Support Vector Machine (FSVM) Performance by Distance-Based Similarity Measure Classification," *HighTech Innov. J.*, vol. 2, no. 4, pp. 285–292, 2021, doi: 10.28991/HIJ-2021-02-04-02.

[4] M. Zulfiqar, M. Kamran, M. B. Rasheed, T. Alquthami, and A. H. Milyani, "Hyperparameter optimization of support vector machine using adaptive differential evolution for electricity load forecasting," *Energy Reports*, vol. 8, pp. 13333–13352, 2022, doi: 10.1016/j.egyr.2022.09.188.

[5] M. E. Febrian, F. X. Ferdinan, G. P. Sendani, K. M. Suryanigrum, and R. Yunanda, "Diabetes prediction using supervised machine learning," *Procedia Comput. Sci.*, vol. 216, no. 2022, pp. 21–30, 2022, doi: 10.1016/j.procs.2022.12.107.

[6] S. Surono, Z. A. R. Hsm, D. A. Dewi, A. E. Haryati, and T. T. Wijaya, "Implementation of Takagi Sugeno Kang Fuzzy with Rough Set Theory and Mini-Batch Gradient Descent Uniform Regularization," *Emerg. Sci. J.*, vol. 7, no. 3, pp. 791–798, 2023, doi: 10.28991/ESJ-2023-07-03-09.

[7] A. Tapkir, "A Comprehensive Overview of Gradient Descent and its Optimization Algorithms," *Iarjset*, vol. 10, no. 11, pp. 37–45, 2023, doi: 10.17148/iarjset.2023.101106.

[8] Y. Tian, Y. Zhang, and H. Zhang, "Recent Advances in Stochastic Gradient Descent in Deep Learning," *Mathematics*, vol. 11, no. 3, pp. 1–23, 2023, doi: 10.3390/math11030682.

[9] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: a review of recent variants and applications," *Neural Comput. Appl.*, vol. 30, no. 2, pp. 413–435, 2018, doi: 10.1007/s00521-017-3272-5.

[10] S. S. Bagalkot, H. A. Dinesha, and N. Naik, "Novel grey wolf optimizer based parameters selection for GARCH and ARIMA models for stock price prediction," *PeerJ Comput. Sci.*, vol. 10, 2024, doi: 10.7717/peerj-cs.1735.

[11] P. M. Kitonyi and D. R. Segera, "Hybrid Gradient Descent Grey Wolf Optimizer for Optimal Feature Selection," *Biomed Res. Int.*, vol. 2021, 2021, doi: 10.1155/2021/2555622.

[12] M. Das, B. R. Mohan, R. M. R. Guddeti, and N. Prasad, "Hybrid Bio-Optimized Algorithms for Hyperparameter Tuning in Machine Learning Models: A Software Defect Prediction Case Study," *Mathematics*, vol. 12, no. 16, pp. 1–31, 2024, doi: 10.3390/math12162521.

[13] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, "A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data," *Front. Energy Res.*, vol. 9, no. April, 2021, doi: 10.3389/fenrg.2021.652801.

[14] O. Sami, Y. Elsheikh, and F. Almasalha, "The Role of Data Pre-processing Techniques in Improving Machine Learning Accuracy for Predicting Coronary Heart Disease," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 6, pp. 816–824, 2021, doi: 10.14569/IJACSA.2021.0120695.

[15] A. Ambarwari, Q. J. Adrian, and Y. Herdiyeni, "Analisis Pengaruh Data Scaling Terhadap Performa Algoritme Machine Learning untuk Identifikasi Tanaman," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 1, pp. 117–122, 2020.

[16] G. A. B. Suryanegara, Adiwijaya, and M. D. Purbolaksono, "Peningkatan Hasil Klasifikasi pada Algoritma Random Forest untuk Deteksi," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 1, no. 10, pp. 114–122, 2021.

[17] M. Nishom, "Perbandingan Akurasi Euclidean Distance, Minkowski Distance, dan Manhattan Distance pada Algoritma K-Means Clustering berbasis Chi-Square," *J. Inform. J. Pengemb. IT*, vol. 4, no. 1, pp. 20–24, 2019, doi: 10.30591/jpit.v4i1.1253.

[18] A. Géron, *Hands-on Machine Learning whith Scikit-Learing, Keras and Tensorfow*. 2019.

[19] S. Raschka, Y. (Hayden) Liu, and V. Mirjalili, *Machine Learning with PyTorch and Scikit Learn*. 2022.

[20] J. A Ilemobayo *et al.*, "Hyperparameter Tuning in Machine Learning: A Comprehensive Review," *J. Eng. Res. Reports*, vol. 26, no. 6, pp. 388–395, 2024, doi: 10.9734/jerr/2024/v26i61188.

[21] A. M. Helmi, M. A. A. Al-Qaness, A. Dahou, R. Damaševičius, T. Krilavičius, and M. A. Elaziz, "A novel hybrid gradient-based optimizer and grey wolf optimizer feature selection method for human activity recognition using smartphone sensors," *Entropy*, vol. 23, no. 8, 2021, doi: 10.3390/e23081065.

[22] Y. Hou, H. Gao, Z. Wang, and C. Du, "Improved Grey Wolf Optimization Algorithm and Application," *Sensors*, vol. 22, no. 10, pp. 1–19, 2022, doi: 10.3390/s22103810.

[23] P. Mazumder and D. S. Baruah, "A Hybrid Model for Predicting Classification Dataset based on Random Forest, Support Vector Machine and Artificial Neural Network," *Int. J. Innov. Technol. Explor. Eng.*, vol. 13, no. 1, pp. 19–25, 2023, doi: 10.35940/ijitee.a9757.1213123.

[24] J. Xu, Y. Zhang, and D. Miao, "Three-way confusion matrix for classification: A measure driven view," *Inf. Sci. (Ny).*, vol. 507, pp. 772–794, 2020, doi: 10.1016/j.ins.2019.06.064.

[25] K. Riehl, M. Neunteufel, and M. Hemberg, "Hierarchical confusion matrix for classification performance evaluation," *J. R. Stat. Soc. Ser. C Appl. Stat.*, vol. 72, no. 5, pp. 1394–1412, 2023, doi: 10.1093/jrsssc/qlad057.

[26] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, pp. 1–14, 2020, doi: 10.1186/s12864-019-6413-7.

