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. 8 No. 6 (2024) 846 - 852 e-ISSN: 2580-0760

Modified Particle Swarm Optimization on Feature Selection for Palm Leaf Disease Classification

Veri Julianto[1*], Ahmad Rusadi Arrahimi[2], Oky Rahmanto[3], Mohammad Sofwat Aldi[4]
[1,2,3,4] Department of Information Technology, Politeknik Negeri Tanah Laut, Pelaihari, Indonesia
[1]veri@politala.ac.id, [2]ahmadrusadi@politala.ac.id, [3]oky.rahmanto@politala.ac.id,
[4]mohammad.sofwat.aldi@mhs.Politala.ac.id

Abstract

Palm oil plantations in Indonesia face challenges in enhancing productivity and profitability, notably due to pest attacks that reduce production. Early identification and classification of plant conditions, particularly palm oil leaves, are crucial for mitigating losses. This study explores the application of artificial intelligence, specifically computer vision and machine learning, for disease detection. Various machine learning techniques, including Local Binary Pattern (LBP), K-Nearest Neighbors (KNN), and Support Vector Machine (SVM), have been used in different studies with varying accuracy. This research focuses on modifying Particle Swarm Optimization (PSO) for feature selection in identifying diseases in palm oil leaves. The PSO modification combined with logistic regression and Bayesian Information Criterion (BIC) significantly enhances KNN performance. Accuracy improved from 95.75% to 97.85%, while precision, recall, and F1-score reached approximately 98.80%. Additionally, the modified KNN+PSO achieved the shortest computation time of 0.0872 seconds, indicating high computational efficiency. These results demonstrate that the PSO modification not only improves accuracy but also computational efficiency, making it an effective method for enhancing KNN performance in detecting palm oil leaf diseases.

Keywords: *Disease Identification; Leaf Classification; K-Nearest Neighbors (KNN); Particle Swarm Optimization (PSO); Computational Efficiency*

How to Cite: V. Julianto, Ahmad Rusadi Arrahimi, Oky Rahmanto, and Mohammad Sofwat Aldi, "Modified Particle Swarm Optimization on Feature Selection for Palm Leaf Disease Classification", *J. RESTI (Rekayasa Sist. Teknol. Inf.)* , vol. 8, no. 6, pp. 846 - 852, Dec. 2024.
DOI: https://doi.org/10.29207/resti.v8i6.6049

1. Introduction

Oil palm plantations in Indonesia are focused on meeting environmental requirements that ensure the quality of oil palm production [1]. Plantation management based on effectiveness and efficiency is an important part of increasing the productivity and profitability of the oil palm business with the concept of precision agriculture that focuses on land management. [2]. One of the obstacles to oil palm productivity lies in pest attacks that result in the emergence of diseases that make production decline. Therefore, it is necessary to identify and classify early on to determine whether oil palm plants are healthy or infected with disease. How to detect oil palm disease, one of which can be seen from the condition of the leaves [3]. One way to identify diseases in agriculture is to use the help of artificial intelligence [2].

Computer vision as one of the fields in artificial intelligence and machine learning (ML) has been developed for crop and land monitoring [4], determining fruit maturity [5] and detecting diseases[6]. In previous research, ML was successfully implemented to detect diseases in leaves [7] and fruit[8]. In general, this method has stages such as preprocessing, segmentation, feature extraction and classification [9]. ML successfully performed disease classification on orange leaves by applying Local Binary Pattern (LBP) and color histogram approach based on RGB and HSV. Furthermore, the classification process uses a bagge tree and produces 99.9% accuracy using 99 infected images and 100 healthy images[9].

Other research related to disease classification in grape plants using k-nearest neighbors (KNN) resulted in an accuracy of 98.75%[10]. Furthermore, it was also successful in detecting diseases on jackfruit by using

Received: 16-09-2024 | Accepted: 04-12-2024 | Published Online: 29-12-2024

SVM as a classifier and Exponential Spider Monkey Optimization as its feature selection resulting in an accuracy of 90%. Research related to plant diseases was also carried out by detecting diseases on oil palm leaves using color histograms and supervised classifiers [5]. The studies used PCA as feature extraction and ANN for classification with 99.67% accuracy. However, the studies that have been done have problems with large dimensions when performing the classification process. This dimension greatly affects the performance of machine learning algorithms in the classification process [11]. One of the techniques used to perform feature reduction is optimization using Swarm Intelligence (SI).

The use of SI to aid the feature selection process has been demonstrated in several previous studies. In[12] feature selection was developed using Particle Swarm Optimization (PSO) to improve the accuracy of diagnosing diseases in the human body. PSO is also used in feature reduction in the cancer classification process[13]. The use of the PSO algorithm is powerful because studies conducted by [14] , compared with other swarm algorithms, it is still better. In this study, PSO combined with SVM and Adaboost as a classification algorithm seems to provide much better accuracy.

Research conducted by[15] successfully used PSO with a combination of logistic regression models and *Bayesian Information Criterion (BIC)* in reducing dimensions and increasing dataset accuracy. Further research conducted by [16] using self-adaptive parameters and PSO succeeded in reducing computation time tested on datasets with dimensions of more than 600.

Based on these studies, this research will focus on disease identification in oil palm by identifying it through leaves by modifying the feature selection process using PSO. PSO is modified with a combination of logistic regression models and Bayesian Information Criterion (BIC) as a fitness evaluation function that will be applied in the palm leaf feature reduction process and then tested for accuracy. Features obtained through PSO will be classified using SVM and KNN.

2. Research Methods

This research method involves six consecutive stages, including image acquisition, segmentation, morphology, feature extraction, feature selection and classification with the trained model.

At this point, a camera was used to transform analogue photos of oil palm leaves into digital photographs in order to gather the dataset. A total of 1500 photographs were collected, of which 750 belonged to each class — that is the sick class and the healthy class.

Table 1. Smartphone camera specs and settings

Specifications	Description
Camera Resolution	108 MP
Image Size	2944 x 2944
Zoom Flash Mode	None
Auto Focus	None
Camera Distance	30-40 cm
Exposure Value	0
ISO	400
White Balancing	Auto
Camera Resolution	108 MP

Once the digital image of the oil palm leaves has been acquired, the subsequent step is preprocessing, which is undertaken with the objective of enhancing the quality of the image that has been obtained. Subsequently, the cropping process is applied with the intention of focusing the image on the area of the palm leaves that is of interest.

In the feature extraction stage, the image utilized is the result of the pre-processing stage. In this research, the distinction between the various classes is based on color and texture. In order to ascertain the color features, the RGB, HSV, LAB, and HSI color spaces are utilized, with the pixel value calculated for each color channel of the image object. Moreover, for each color channel, the mean pixel value is calculated, while for texture features, contrast and energy parameters are employed. The results of this feature extraction will be utilized as input for the classification process of training and test data. Subsequently, the values are normalized using the min-max method.

Feature selection procedures reduce or minimize the number of features by selecting a subset of the original features. This method is applied during the pre-processing stage to identify relevant attributes, which are typically unknown beforehand, and eliminate irrelevant or redundant features that do not contribute significantly to classification. Feature selection is applied in various fields, especially to improve classification accuracy.

Logistic Regression Model (LRM) is a powerful statistical method for modeling binary target variables. This approach is useful when the outcome or dependent variable is binary, having two possible outcomes. A common example of its use is in medical diagnosis, such as cancer classification, where the target variable indicates whether the tumor is malignant (1) or benign (0). In this study, the method is applied to classify oil palm leaves, with diseased leaves assigned a value of 0 and healthy leaves assigned a value of 1 [15].

A feature matrix can be mathematically described as $\boldsymbol{F} = \left(f_g\right)_{nxd}$, with each column representing a feature and each row representing an instance (observation). The numerical value of f_{ij} represents the measurement of a given feature $j(j = 1,2, \dots, d)$ in a specific instance $i(i = 1,2, \dots, n)$. Provided a dataset for training $\{(\boldsymbol{f_i}, y_i)\}_{i=1}^{n}$, where $\boldsymbol{f_i} = \left(f_{i,1}, f_{i,2}, \dots, f_{i,d}\right)$ represents a vector of the i^{th}feature values, and $y_i \epsilon \{0,1\}$ for $i = 1,2, \dots, n$, where $y_i = 1$ specifies the i^{th} sample is in class 1 and $y_i = 0$ specifies the i^{th} sample is in class 2. Overall, the goal is to categorize the new instance and locate the important attributes with high classification accuracy. Assuming $p(\boldsymbol{f_i})$ reflects the class-conditional probability for instance i when $y_i = 1, p(f_i) = \Pr(y_i = 1|f_i)$, then the LRM is shown in Equation 1.

$$\ln\left[\frac{p(f_i)}{1-p(f_i)}\right] = \alpha_0 + f_i{}^t\alpha, i = 1,2,\dots,n \quad (1)$$

$\alpha = (\alpha_1,\dots,\alpha_d)^T \epsilon R^d$ represents a vector of unknown feature coefficients. The negative log likelihood function in Equation 1 is Equation 2.

$$\ell(\alpha) = -\sum_{i=1}^{n}\left[y_i \ln\big(p(f_i)\big) + (1-y_i)\ln\big(1-p(f_i)\big)\right]$$

$$= -\sum_{i=1}^{n}\left[y_i\big(\alpha_0 + f_i^T\alpha\big) - \ln\big(1+\exp\,(\alpha_0 + f_i^T\alpha\big)\right] \quad (2)$$

Minimizing Equation 2 provides the maximum likelihood estimator (MLE) for α. The advantage of LRM is that it allows you to estimate the probability $p(f_i)$ and $1 - p(f_i)$ for each class while also classifying cases. The classification rule states that an instance belongs to class 1 if $\Pr(y_i = 1|f_i) \geq 0.5$ and it belongs to class 2 if $\Pr(y_i = 1|f_i) \leq 0.5$ [17].

The practical swarm optimization (PSO) algorithm combines a population-based global search technique with a substitution solution. PSO's initial population consists of a swarm of randomly produced particles. Every particle has two properties: velocity and location. These are applied in the context of a search problem, where $X_i = \{x_{i1}, x_{i2},\dots,x_{im}\}, i = 1,2,3,\dots,n$ represent the position vector. $V_i = \{v_{i1}, v_{i2},\dots,v_{im}\}, \quad i = 1,2,3,\dots,n$ represent the velocity vector and $Pbest_i = \{p_{i1}, p_{i2},\dots,p_{im}\}, i = 1,2,3,\dots,n.$ represent the personal best position attained for the i_{th} particle in a swarm with m particles[15].

The optimum position for each particle is determined by its velocity, while the best position for the entire swarm is determined by $Gbest = \{g_1, g_2,\dots,g_m\}$. In the search space, the movement of particles is guided by position and velocity updating, where the velocity updating equation is provided in Equation 3 [18].

$$v_{ij}^{t+1} = wv_{jt}^t + k_1r_1\big(p_{ij}^t - x_{ij}^t\big) + k_2r_2\big(g_j^t - x_{ij}^t\big) \quad (3)$$

Additionally, the position will be calculated as shown in Equation 4.

$$x_{ij}^t = x_{ij}^t + x_{ij}^{t+1} \quad (4)$$

In this model, the variable t represents the number of iterations, while w denotes the inertia weight, which ranges between 0 and 1. The cognitive memory and social learning factors, k_1 and k_2, respectively, are represented by the variables k_1 and k_2. Finally, the random numbers r_1 and r_2 are uniformly selected between 0 and 1. During each iteration, the particles in the swarm are updated with the positions and velocities. Additionally, the Pbest (new personal best) and $Gbest$ (global best) values and corresponding particles are identified.

The PSO algorithm was originally introduced for continuous optimization. Subsequently, Kennedy and Eberhart proposed a binary PSO (BPSO), which can be used as a search space in discrete problems. In BPSO, the position of each particle is encoded by a binary string, restricted to 0 or 1, and the velocity represents the probability of an element with a value of 1. The sigmoid function can be introduced to transform velocity into the range of 0 and 1, where applicable. [19]. The particle's position in BPSO is updated using Equations 5 and 6.

$$x_{ij}^t = \begin{cases} 1 & if \text{ rand () } < f(v_{ij}^t) \\ 0 & if\ 0.w \end{cases} \quad (5)$$

$$f\big(v_{ij}^t\big) = \frac{1}{1+e^{v_{ij}^{t+1}}} \quad (6)$$

In this context, "rand" represents a random number selected from a uniform distribution within the range [0,1], while x_{ij}^t is transformed into [0,1] through the application of a sigmoid function. Equations 5 and 6 normalize velocities t_{ij} into range [0, 1] and obtain the solution from the binary vector of particle positions by selecting the set of features with the position set to 1[15]

The proposed algorithm is based on two principal concepts. The initial concept entails the utilisation of a logistic regression model as a classifier, in conjunction with the employment of Particle Swarm Optimization (PSO) for the identification of the optimal set of features. The second concept entails the utilisation of the Bayesian Information Criterion (BIC) as a fitness quality measure, with the objective of identifying the most relevant features during the PSO search process. Several studies used the following two fitness functions [20]–[22].

$$Fit1 = \delta \times CA + (1-\delta) \times \left(\frac{p-q}{p}\right) \quad (7)$$

$$Fit1 = \delta \times CA + (1-\delta) \times \left(\frac{1}{q}\right) \quad (8)$$

In this context, CA represents the degree of accuracy achieved by the classifier in classifying the data. In this context, the variable p represents the total number of features in the dataset, while q denotes the number of features that have been selected. The parameter δ, which lies within the interval [0,1], indicates the relative quality between CA and q. In practice, δ is typically set to 0.9. In the proposed PSO-LRBIC algorithm, classification accuracy is defined as the number of correctly classified instances using a logistic regression model as the classifier. The number of selected features, q, is determined through the use of particle swarm optimization (PSO). For each particle, the fitness functions described in Equations 3 and 4 are calculated and their maximum value is compared with the global best fitness. In the event that the current fitness value is superior, the global best fitness value is updated with this value.

Although Equations 7 and 8 are widely utilized, their efficacy may be compromised due to the dependency of both fitness functions on the value of δ. Modifications to the value of δ will consequently affect the performance of the aforementioned functions. Furthermore, the use of classification accuracy as a measure is not advised for imbalanced datasets[23]. Consequently, the Bayesian Information Criterion (BIC) is put forth as a fitness function, designated as Fit-BIC, which is defined as shown in Equation 9.

$$Fit - BIC = -2 \times logL(\theta) + q \times log(n) \quad (9)$$

In this context, the log-likelihood function, represented by the symbol θ, is given by Equation 2. It should be noted that the BIC fit integrates the loss function with the number of selected features. The optimal global fitness value can be obtained by minimizing the objective function given in Equation 5.

2.5 Classification

Support Vector Machine (SVM) is a supervised learning method used for classification and regression. The basic idea of SVM is to maximize the margin of the hyperplane. By selecting the hyperplane that has the maximum margin, this classification method provides better generalization. Simply put, SVM tries to find the best hyperplane that separates two classes of data in the input space. The hyperplane, which is the decision boundary between the two classes, is found by measuring the margin and searching for the maximum. The margin is the distance between the hyperplane and the closest data from each class, while the data is called the support vector. The first boundary plane in Equation 10 separates the first class, and the second boundary plane in Equation 11 separates the second class.

$$x_i . w + b \geq +1 \; for \; y_i = +1 \quad (10)$$

$$x_i . w + b \leq -1 \; for \; y_i = -1 \quad (11)$$

w is the plane normal and is the position of the plane relative to the coordinate center. The margin value (distance) between the bounding planes is shown in Equation 12.

$$\frac{1-b-(-1-b)}{w} = \frac{2}{||w||} \quad (12)$$

The two constraint fields in equations (6) and (7) can be represented as in Equation 13.

$$y_i(x_i . w + b) - 1 \geq 0 \quad (13)$$

The optimal margin can be calculated by maximizing the distance between the hyperplane and the closest data where this distance can be formulated by equation (8) where is the weight vector. Furthermore, this problem is formulated into a Quadratic Programming (QP) problem by minimizing the inverse of equation (3). This formula for finding the best separating field is a quadratic programming problem, so the global maximum value of can always be found. Once the solution of the quadratic programming problem is found (value), the class of the test data can be determined based on the value of the decision function:

$$(x_d) = \sum_{i=1}^{N_s} \alpha_i y_i x_i x_d + b \quad (14)$$

x_i is the support vector, Ns is the number of support vectors, and x_d is the data to be classified.

The K-nearest neighbor (KNN) method is the simplest classification technique. The KNN algorithm is a supervised learning method. The KNN algorithm employs a neighborly classification approach to predict the value of a new test sample. This is in accordance with the principles set forth in references [24], [25]. The classification process of the K-Nearest Neighbor (KNN) is comprised of the following steps: Enter the calculated convolution image value; Sets the parameter k (number of nearest neighbors); Using the Euclidean distance model, calculate proximity to supplied training data using Equation 15.

$$D(x, y) = \left||x - y|\right|_2 \sqrt{\sum_{j=1}^{N} |x - y|^2} \quad (15)$$

Sort distance results in ascending order (from highest to lowest value); Determine the number of each class based on k nearest neighbors; The majority class is used to classify the test data.

2.6 Evaluation

To evaluate the classification performance of the proposed method and the other two methods, three performance metrics are used: classification accuracy (CA) shown in Equation 16, where TP represents the number of true positives, FP represents false positives, TN represents the number of true negatives, and FN represents false negatives.

$$\text{CA} = \frac{\text{TP+TN}}{\text{TP+FP+FN+TN}} \times 100\% \quad (16)$$

To assess the performance of a classification model, particularly in binary classification problems, the confusion matrix components are used. True Positives (TP) are the number of positive cases that are accurately forecasted as positive, whereas False Positives (FP) are negative instances that are wrongly projected as positive. True Negatives (TN) are negative cases that have been successfully classified as such, while False Negatives (FN) are positive examples that have been incorrectly projected as negative as shown in Equations 17-20.

$$recall = \frac{TP}{TP+FN} \quad (17)$$

$$precision = \frac{TP}{TP+FP} \quad (18)$$

$$f1 - score = \frac{2 \times precision \times recall}{precision + recall} \quad (19)$$

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (20)$$

3. Results and Discussions

In this study, we used a dataset consisting of classification images of palm leaf disease. The focus is on two main classes of palm leaf condition: diseased (750 images) and healthy (750 images). The images underwent preprocessing using a Gaussian filter, and the features from both the diseased and healthy images were then extracted. The color features employed in this study include RGB, HSV, LAB, and HIS, with pixel values for each channel calculated for the objects in the images. For each channel, the mean pixel value was determined, while contrast and energy were used as texture features. In total, there are 14 features, comprising 12 color features and 2 texture features. All features were normalized using min-max normalization before undergoing feature selection with a modified PSO algorithm. The classification was then tested using SVM and KNN classifiers.

Based on the feature extraction results, several experimental scenarios were conducted to classify the training and test data by applying the 10-fold cross-validation method. Model performance was evaluated using accuracy, precision, recall, and F1-score metrics. The PSO algorithm used has a population size of 20 and a maximum of 20 iterations. The test results can be seen in Table 2.

Table 2 Summary table of calculations for accuracy, precision, recall

Algorithm	Accuracy	Precision	Recall	F1-score
SVM	0.9965	0.9964	0.9966	0.9965
SVM+PSO	0.9923	0.9925	0.9922	0.9923
SVM+PSO +Modified	0.9965	0.9964	0.9966	0.9965
KNN	0.9575	0.9572	0.9578	0.9574
KNN+PSO	0.9616	0.9614	0.9620	0.9616
KNN+PSO Modified	0.9785	0.9880	0.9882	0.9881

The table presents a comparative analysis of different algorithms applied to a classification problem, showcasing metrics such as accuracy, precision, recall, and F1-score. The base SVM model exhibits an impressive degree of accuracy, with a score of 99.65%. This is accompanied by similarly high precision (99.64%), recall (99.66%), and F1-score (99.65%). The application of PSO to SVM resulted in a slight decline in performance, with an observed decrease in accuracy to 99.23% and a similar decline in all other metrics (ranging from 99.22% to 99.25%). This indicates that the feature selection process using PSO in this instance had a slight impact on the performance of SVM. However, the modified SVM+PSO approach recovers the lost ground, achieving identical results to the base SVM model. This suggests that the modifications successfully enhanced the model's robustness.

In contrast, KNN demonstrates a lower baseline accuracy of 95.75%, with precision, recall, and F1-score also in the range of 95.72% to 95.78%. Upon integrating PSO, KNN+PSO shows a slight improvement, raising the accuracy to 96.16% and other metrics to around 96.14% to 96.20%. Interestingly, the modified KNN+PSO significantly outperforms both the base KNN and the PSO-enhanced KNN, achieving 97.85% accuracy and dramatic improvements in precision, recall, and F1-score, all reaching approximately 98.80%. This indicates that the modification has a substantial positive impact on KNN, leading to more accurate and balanced classifications.

Overall, while SVM demonstrates consistent high performance, KNN benefits more from the modifications, showcasing a notable upward trend in all evaluation metrics after PSO modification. The trend suggests that PSO and its modifications have a more pronounced effect on KNN than on SVM.

Table 3. Summary table of calculations for Accuracy Sensitivity Specificity Computation Time (second)

Algorithm	Accuracy	Sensitivity	Specificity	Computation Time (second)
SVM	0.9965	0.9947	0.9985	1.4133
SVM+PSO	0.9923	0.9960	0.9883	0.8931
SVM+PSO +Modified	0.9965	0.9947	0.9985	1.2992
KNN	0.9575	0.9507	0.9649	0.2011
KNN+PSO	0.9616	0.9547	0.9693	0.1772
KNN+PSO Modified	0.9785	0.9867	0.9898	0.0872

Table 3 summarizes the evaluation of various classification algorithms, including SVM, KNN, and their variations involving Particle Swarm Optimization (PSO) and related modifications. The analysis includes metrics of accuracy, sensitivity, specificity, as well as the computation time required by each algorithm.

Support Vector Machine (SVM) showed excellent performance with an accuracy of 99.65%, sensitivity of 99.47%, and specificity of 99.85%. The computation time for SVM was 1.4133 seconds, which is moderate. When PSO was applied to SVM, there was a small decrease in accuracy to 99.23%, while sensitivity increased to 99.60% and specificity decreased to 98.83%. The application of PSO reduced the computation time to 0.8931 seconds, which indicates higher efficiency in feature selection. The modified SVM+PSO again showed identical accuracy performance to the basic SVM model, but the computation time increased slightly to 1.2992 seconds. Sensitivity and specificity remained at the same level as the base SVM model, indicating that the modification did not significantly affect the classification quality, but slightly extended the computation time.

In the K-Nearest Neighbors (KNN) algorithm, the base model achieved 95.75% accuracy, 95.07% sensitivity, and 96.49% specificity, with the fastest computation time among all models at 0.2011 seconds. KNN+PSO showed an improvement in accuracy to 96.16% and sensitivity and specificity to 95.47% and 96.93%, respectively. The computation time for KNN+PSO is slightly lower than the base KNN model, at 0.1772 seconds, indicating good efficiency in feature selection. The modified KNN+PSO gave the most superior performance with 97.85% accuracy, 98.67% sensitivity, and 98.98% specificity. The computation time for this model is the shortest at 0.0872 seconds, reflecting maximum efficiency in terms of feature selection and data processing.

Overall, SVM remains superior in terms of accuracy and specificity, but KNN with modified PSO shows better performance in terms of sensitivity and accuracy, while maintaining highly efficient computation time. The application of PSO modifications to KNN is shown to provide significant improvements to key evaluation metrics as well as computational efficiency, making it a highly competitive option for this classification task.

The application of PSO modification to the KNN algorithm shows significant improvement in classification performance. Accuracy increased dramatically from 95.75% in the basic KNN model to 97.85% in the modified KNN+PSO. Precision, recall, and F1-score also increased sharply, reaching around 98.80%. In addition, the computation time of the modified KNN+PSO is the shortest, only 0.0872 seconds, showing maximum efficiency. In conclusion, PSO modification not only improves accuracy but also increases computational efficiency, making it highly effective in improving KNN performance.

This research shows that the Particle Swarm Optimization (PSO) algorithm with the Logistic Regression Bayesian Information Criterion (BIC) model produces very good accuracy when applied to the SVM and KNN classification algorithms. In SVM, the results obtained do not have a significant impact on changes in accuracy. In contrast, the combination of modified PSO with KNN provides a more positive increase in accuracy. This is in accordance with the research of [15] in conducting feature selection and increasing the resulting accuracy.

4. Conclusions

The application of PSO algorithm modification using logistic regression models and Bayesian Information Criterion (BIC) resulted in excellent accuracy when tested with SVM and KNN, compared to PSO without modification. The implementation of PSO modification to the KNN algorithm has resulted in notable enhancements in classification performance, with an increase in accuracy. The PSO modification results in notable enhancements in both classification metrics and processing time, thereby establishing it as an effective feature selection method. In conclusion, the KNN+PSO approach is a highly competitive method for improving classification tasks, particularly in situations where computational efficiency is of paramount importance. Future research needs to add variations in disease classes in oil palm so that the algorithm modifications that have been developed can be tested on various disease classes. Furthermore, other algorithms such as CNN and ANN will also be applied to assist in classification.

Acknowledgements

This work was supported by a research grant Indonesia ministry of education 2024 with contract number is 88/KWT/D.D4/PPK.01.APTV/III/2024

References

[1] M. Ichsan, W. Saputra, and A. Permatasari, “Oil palm smallholders on the edge: Why business partnerships need to be redefined,” *Br. SPOS Indones.*, pp. 1–12, 2021, [Online]. Available: https://sposindonesia.org/wp-content/uploads/2021/07/Oil-palm-smallholders-on-the-edge-Why-business-partnerships.pdf
[2] G. A. W. Satia, E. Firmansyah, and A. Umami, “Perancangan sistem identifikasi penyakit pada daun kelapa sawit (Elaeis guineensis Jacq.) dengan algoritma deep learning convolutional neural networks,” *J. Ilm. Pertan.*, vol. 19, no. 1, pp. 1–10, Mar. 2022, doi: 10.31849/JIP.V19I1.9556.
[3] L. A. Harahap, R. I. Fajri, M. F. Syahputra, R. F. Rahmat, and E. B. Nababan, “Identifikasi Penyakit Daun Tanaman Kelapa Sawit dengan Teknologi Image Processing Menggunakan Aplikasi Support Vector Machine,” *Talent. Conf. Ser. Agric. Nat. Resour.*, vol. 1, no. 1, pp. 53–59, 2018, doi: 10.32734/anr.v1i1.96.
[4] J. Zheng *et al.*, “Growing status observation for oil palm trees using Unmanned Aerial Vehicle (UAV) images,” *ISPRS J. Photogramm. Remote Sens.*, vol. 173, pp. 95–121, Mar. 2021, doi: 10.1016/J.ISPRSJPRS.2021.01.008.
[5] H. Hamdani, A. Septiarini, A. Sunyoto, S. Suyanto, and F. Utaminingrum, “Detection of oil palm leaf disease based on color histogram and supervised classifier,” *Optik (Stuttg).*, vol. 245, no. July, p. 167753, 2021, doi: 10.1016/j.ijleo.2021.167753.
[6] S. Pandiyan, M. Ashwin, R. Manikandan, K. M. Karthick Raghunath, and G. R. Anantha Raman, “Heterogeneous Internet of things organization Predictive Analysis Platform for Apple Leaf Diseases Recognition,” *Comput. Commun.*, vol. 154, pp. 99–110, Mar. 2020, doi: 10.1016/J.COMCOM.2020.02.054.
[7] V. K. Vishnoi, K. Kumar, and B. Kumar, *Plant disease detection using computational intelligence and image processing*, vol. 128, no. 1. Springer Berlin Heidelberg, 2021. doi: 10.1007/s41348-020-00368-0.
[8] M. Sharif, M. A. Khan, Z. Iqbal, M. F. Azam, M. I. U. Lali, and M. Y. Javed, “Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection,” *Comput. Electron. Agric.*, vol. 150, no. May 2017, pp. 220–234, 2018, doi: 10.1016/j.compag.2018.04.023.
[9] H. Ali, M. I. Lali, M. Z. Nawaz, M. Sharif, and B. A. Saleem, “Symptom based automated detection of citrus diseases using color histogram and textural descriptors,” *Comput. Electron. Agric.*, vol. 138, pp. 92–104, Jun. 2017, doi: 10.1016/J.COMPAG.2017.04.008.
[10] G. Saleem, M. Akhtar, N. Ahmed, and W. S. Qureshi, “Automated analysis of visual leaf shape features for plant classification,” *Comput. Electron. Agric.*, vol. 157, pp. 270–280, Feb. 2019, doi: 10.1016/J.COMPAG.2018.12.038.
[11] Z. Wang, M. Li, and J. Li, “A multi-objective evolutionary algorithm for feature selection based on mutual information with a new redundancy measure,” *Inf. Sci. (Ny).*, vol. 307, pp. 73–88, Jun. 2015, doi: 10.1016/J.INS.2015.02.031.
[12] H. H. Inbarani, A. T. Azar, and G. Jothi, “Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis,” *Comput. Methods Programs Biomed.*, vol. 113, no. 1, pp. 175–185, Jan. 2014, doi: 10.1016/J.CMPB.2013.10.007.
[13] I. Jain, V. K. Jain, and R. Jain, “Correlation feature selection based improved-Binary Particle Swarm Optimization for gene selection and cancer classification,” *Appl. Soft Comput.*, vol. 62, pp. 203–215, Jan. 2018, doi: 10.1016/j.asoc.2017.09.038.
[14] M. Rostami, K. Berahmand, E. Nasiri, and S. Forouzande, “Review of swarm intelligence-based feature selection methods,” *Eng. Appl. Artif. Intell.*, vol. 100, no. September 2020, p. 104210, 2021, doi: 10.1016/j.engappai.2021.104210.

This is an open access article under the CC BY-4.0 license

[15] O. S. Qasim and Z. Y. Algamal, "Feature selection using particle swarm optimization-based logistic regression model," *Chemom. Intell. Lab. Syst.*, vol. 182, pp. 41–46, Nov. 2018, doi: 10.1016/J.CHEMOLAB.2018.08.016.

[16] Y. Xue, T. Tang, W. Pang, and A. X. Liu, "Self-adaptive parameter and strategy based particle swarm optimization for large-scale feature selection problems with multiple classifiers," *Appl. Soft Comput.*, vol. 88, p. 106031, Mar. 2020, doi: 10.1016/J.ASOC.2019.106031.

[17] Z. Y. Algamal and M. H. Lee, "Penalized logistic regression with the adaptive LASSO for gene selection in high-dimensional cancer classification," *Expert Syst. Appl.*, vol. 42, no. 23, pp. 9326–9332, 2015, doi: 10.1016/j.eswa.2015.08.016.

[18] J. Cervantes, F. Garcia-Lamont, L. Rodriguez, A. López, J. R. Castilla, and A. Trueba, "PSO-based method for SVM classification on skewed data sets," *Neurocomputing*, vol. 228, pp. 187–197, Mar. 2017, doi: 10.1016/J.NEUCOM.2016.10.041.

[19] S. M. Vieira, L. F. Mendonça, G. J. Farinha, and J. M. C. Sousa, "Modified binary PSO for feature selection using SVM applied to mortality prediction of septic patients," *Appl. Soft Comput.*, vol. 13, no. 8, pp. 3494–3504, Aug. 2013, doi: 10.1016/J.ASOC.2013.03.021.

[20] M. A. El Aziz and A. E. Hassanien, "Modified cuckoo search algorithm with rough sets for feature selection," *Neural Comput. Appl. 2016 294*, vol. 29, no. 4, pp. 925–934, Jul. 2016, doi: 10.1007/S00521-016-2473-7.

[21] M. Aladeemy, S. Tutun, and M. T. Khasawneh, "A new hybrid approach for feature selection and support vector machine model selection based on self-adaptive cohort intelligence," *Expert Syst. Appl.*, vol. 88, pp. 118–131, Dec. 2017, doi: 10.1016/J.ESWA.2017.06.030.

[22] G. I. Sayed, A. E. Hassanien, and A. T. Azar, "Feature selection via a novel chaotic crow search algorithm," *Neural Comput. Appl. 2017 311*, vol. 31, no. 1, pp. 171–188, Apr. 2017, doi: 10.1007/S00521-017-2988-6.

[23] A. Moayedikia, K. L. Ong, Y. L. Boo, W. G. Yeoh, and R. Jensen, "Feature selection for high dimensional imbalanced class data using harmony search," *Eng. Appl. Artif. Intell.*, vol. 57, pp. 38–49, Jan. 2017, doi: 10.1016/J.ENGAPPAI.2016.10.008.

[24] A. A. Nababan, M. Khairi, and B. S. Harahap, "Implementation of K-Nearest Neighbors (KNN) algorithm in classification of data water quality," *J. Mantik*, vol. 6, no. 1, pp. 30–35, 2022.

[25] A. Damayunita, R. S. Fuadi, and C. Juliane, "Comparative Analysis of Naive Bayes, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) Algorithms for Classification of Heart Disease Patients," *J. Online Inform.*, vol. 7, no. 2, pp. 219–225, 2022, doi: 10.15575/join.v7i2.919.