Jurnal Minfo Polgan (JMP)
Volume 15, Nomor 3, September-Desember 2026
DOI : https://doi.org/10.33395/jmp.v15i3.16603
e-ISSN : 2797-3298
p-ISSN : 2089-9424

Performance Evaluation Of K-Nearest Neighbors And Random Forest On Monkeypox Dataset Using Particle Swarm Optimization (PSO)

[1]Lima Hartimar Rambe, [2]Rika Rosnelly, [3]Wanayumini
[1*,2,3]Universitas Potensi Utama, Medan, Indonesia
*Korespondensi: limahartimarrambe12345@gmail.com

Submit : 29 Jun | **Diterima** : 17 Jul 2026 | **Terbit** : 03 Sept 2026

Abstract

This study aims to evaluate and compare the performance of the K-Nearest Neighbors (KNN) and Random Forest (RF) algorithms in classifying Monkeypox disease by applying the Particle Swarm Optimization (PSO) technique. PSO is used to obtain optimal parameters for each algorithm in order to improve the accuracy and efficiency of the classification models. The data used consist of 10,000 patient records with 11 clinical symptom features representing health conditions related to Monkeypox infection and are numerical in nature. Model performance evaluation was conducted before and after optimization using accuracy, precision, recall, F1-score, and sensitivity metrics. The results show that the application of PSO improves the performance of both algorithms compared to the non-optimized models, particularly in enhancing accuracy and balancing precision and recall. Overall, the combination of classification algorithms with the PSO optimization technique produces more optimal performance in detecting Monkeypox disease. This study is expected to contribute to the development of more accurate machine learning based classification systems and to support effective early diagnosis of Monkeypox.

Keywords: Machine Learning; K-Nearest Neighbors; Random Forest; Particle Swarm Optimization; Monkeypox Classification

Abstrak

Penelitian ini bertujuan untuk mengevaluasi dan membandingkan kinerja algoritma K-Nearest Neighbors (KNN) dan Random Forest (RF) dalam klasifikasi penyakit cacar monyet (Monkeypox) dengan menerapkan teknik optimasi Particle Swarm Optimization (PSO). PSO digunakan untuk memperoleh parameter optimal pada masing-masing algoritma guna meningkatkan akurasi dan efisiensi model klasifikasi. Data yang digunakan terdiri dari 10.000 data pasien dengan 11 fitur gejala klinis, yang merepresentasikan kondisi kesehatan terkait infeksi cacar monyet dan bersifat data numerik. Evaluasi kinerja model dilakukan sebelum dan sesudah optimasi menggunakan metrik akurasi, presisi, recall, F1-score, dan sensitivitas. Hasil penelitian menunjukkan bahwa penerapan PSO mampu meningkatkan kinerja kedua algoritma dibandingkan dengan model tanpa optimasi, terutama dalam meningkatkan akurasi dan keseimbangan antara presisi dan recall. Secara keseluruhan, kombinasi algoritma klasifikasi dengan teknik optimasi PSO menghasilkan performa yang lebih optimal dalam mendeteksi penyakit cacar monyet. Penelitian ini diharapkan dapat berkontribusi dalam pengembangan sistem klasifikasi berbasis pembelajaran mesin yang lebih akurat serta mendukung proses diagnosis dini cacar monyet secara efektif.

***Kata Kunci**: Pembelajaran Mesin; K-Nearest Neighbors; Random Forest; Particle Swarm Optimization; Klasifikasi Cacar Monyet*

INTRODUCTION

K-Nearest Neighbors (KNN) is Wrong One algorithm the most popular machine learning and easy used . The classification method used by the K-Nearest Neighbors (KNN) algorithm is based on the shortest distance based on the data object. The K-Nearest Neighbors (KNN) method uses the appropriate distance to classify new data. The K-nearest neighbor distance is calculated and the class label of the nearest neighbor is predicted as the class label of the new instance. The accuracy of K-Nearest Neighbors (KNN) is greatly influenced by choosing

the number of K nearest neighbors. (Saifur et al., 2021). K-NN has high flexibility because it can be applied to data that cannot be represented as a feature vector directly, as long as there is an adequate similarity measure. This makes K-NN applicable in various data contexts, including text, images, and time-series. In addition, when explanations of classification results are needed, this algorithm is also very useful because decisions can be explained by referring directly to relevant neighboring data.(Cunningham et al., 2021)

Random Forest is one of the most widely used algorithms in machine learning and data science. This algorithm is considered moderated and is widely used in classification problems. One of the advantages of this algorithm is its ability to handle continuous, numeric, categorical data, and even data with many missing values, both in regression and classification problems. The random forest model is a collection of decision trees used in machine learning problems. All nodes and internal branches in the decision tree are associated with test results. The test results are not dependent on a single decision tree but rather compare the results of the trees formed. (Hamdan et al., 2024)

Some studies generally use K-Nearest Neighbors and Random Forest for classification. (Hamdan et al., 2024)conducted research on the monkeypox dataset by comparing the performance of machine learning such as KNN and random forest , testing was carried out by dividing the data into Latin data and test data first, then the accuracy of the data classification was measured using statistical techniques, this technique helps set standards for accuracy, precision, F1-score, and sensitivity for the algorithm used. in the results of the study, it was seen that K Nearest Neighbors (KNN) showed good performance with balanced values in various metrics, while random forest had the lowest values in all the metrics evaluated.

Particle Swarm Optimization (PSO) is a nature-inspired optimization technique introduced in 1995. PSO draws inspiration from the collective behavior of animals, such as bees, birds, and fish, which move in groups. Originally developed for optimizing real number spaces, PSO has found successful applications in a variety of domains, including function optimization, ANN training, and fuzzy system control.

This research (Hadjaidji et al., 2025)aims to evaluate and compare the performance of various machine learning algorithms in detecting Parkinson's disease. The goal of this study was to identify the most accurate classification method and improve model performance through parameter optimization using Particle Swarm Optimization (PSO). The algorithms compared covering K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Gradient Boosting (GB), and Naive Bayes, Good before and after optimized . PSO implementation is carried out on during the training process. PSO plays a role in the feature weighting process , namely determine weight or level interest each feature voice towards the diagnosis of Parkinson's disease . This process is carried out by representing each particle in the PSO as a candidate feature weight, which is applied to the training data. Model performance is measured using cross-validation to obtain fitness values, which are then used by PSO to update particle positions until finding the optimal weight combination. The dataset used consists of 197 sound data with 23 acoustic features. The pre-processing stage is carried out using the Z-Score normalization technique for data standardization, SMOTE (Synthetic Minority Over-sampling Technique) to handle class imbalance, and k-fold cross-validation to obtain a more stable and accurate performance evaluation.

It is interesting to observe, based on previous research conducted by (Hamdan et al., 2024)regarding the performance evaluation of the KNN (K-Nearest Neighbors) and Random Forest algorithms , as well as a comparative evaluation of machine learning + PSO studied by (Hadjaidji et al., 2025)which shows the advantages and performance provided by the methods used as I have described in the previous paragraph. Researchers are interested in conducting a comparative evaluation of the k-nearest neighbor (KNN) and random forest algorithms, both before and after the optimization process is carried out on the monkeypox dataset. Machine learning is used to classify data, while Particle Swarm Optimization (PSO) is used to find the best parameters of each algorithm used. The KNN and Random Forest algorithms can only process data in numeric form, so an encoding process is required to convert categorical data into numeric data. In addition, a cleaning process is also carried out to clean the data from missing or irrelevant values, as well as the process of dividing the data into training data and test data. All of these pre- processing stages are necessary so that the data is ready to be used in building a classification model using the KNN and Random Forest algorithms.

The comparative performance evaluation was analyzed using evaluation metrics such as accuracy, precision, F1 score, recall, and sensitivity. These metrics were used to comprehensively compare the performance of each model and determine which model had the best accuracy and classification performance after the optimization process.

Detecting the symptoms of monkeypox is very important because apart from being dangerous and having spread to various countries, monkeypox also has many similar symptoms to other diseases such as fever, headaches, and muscle pain.

Therefore, the researcher intends to compare the performance evaluation of K-Nearest Neighbors and Random Forest on the Monkeypox dataset using PSO (Particle Swarm Optimization).

RESEARCH METHODOLOGY

On stage data collection, research This using secondary data obtained from site Kaggle And published on 2023. The dataset used consists of from 25,000 patient data with 11 features symptom clinical nature categorical And boolean, includes information disease systemic, various symptom physical, as well as history infection, with feature MonkeyPox as a classification target label. This label shows the patient's status , namely positive or negative smallpox monkeys . The dataset used as base in the classification process For building a detection model disease smallpox monkey machine learning based .

Data pre-processing stages are carried out for increase data quality and model performance. This process covering data cleaning , checking missing value and duplicate data , labeling (encoding), standardization , and selection features . Results checking show that the dataset is not own mark is lost as well as duplicate data. Categorical data changed to form numeric use label encoding, whereas feature type boolean converted use one-hot encoding. Next, data standardization is performed with StandardScaler For equalize scale features, which are very important especially for sensitive K-Nearest Neighbors (KNN) algorithm to difference scale. Selection feature done use Chi-Squared test for evaluate relevance every feature to MonkeyPox target variable .

On stage modeling, data sharing become training data and test data with ratio of 80% and 20%. For overcome imbalance class , applied SMOTE technique on training data so that distribution class become balanced . Research This use four testing models , namely KNN and Random Forest without optimization , as well as KNN and Random Forest with Particle Swarm Optimization (PSO) optimization . PSO is used forget the best parameters on each algorithm. Evaluation model performance is carried out use metric accuracy, precision, recall, and F1-score, in order to determine the most optimal classification model in detect disease smallpox monkey.

RESULTS AND DISCUSSION

Data Pre-Processing

Data Cleansing

The initial pre-processing stage involved checking for missing values and duplicate data to ensure the quality and consistency of the dataset. This inspection was performed on all 11 clinical symptom features. The results showed no missing values across all attributes.

```
Jumlah nilai yang hilang di setiap kolom:
```

	0
Patient_ID	0
Systemic Illness	0
Rectal Pain	0
Sore Throat	0
Penile Oedema	0
Oral Lesions	0
Solitary Lesion	0
Swollen Tonsils	0
HIV Infection	0
Sexually Transmitted Infection	0
MonkeyPox	0

Figure 1. Checking Missing Values

Encoding

At this stage, the encoding process is performed on features with categorical and Boolean data types. This process is crucial because the machine learning algorithms used cannot process non-numeric data directly. Some encoding techniques used are as follows:

1. Label encoding (0/1) Features with boolean or binary values such as Rectal Pain, Sore Throat, Penile Oedema, Oral Lesions, Solitary Lesion, Swollen Tonsils, HIV Infection, and Sexually Transmitted Infection are converted into numeric form with a representation of 0 for the condition “no” and 1 for the condition “yes”.

Kolom boolean/biner berhasil dikonversi ke numerik (0/1).

	Patient_ID	Systemic Illness	Rectal Pain	Sore Throat	Penile Oedema	Oral Lesions	Solitary Lesion	Swollen Tonsils	HIV Infection	Sexually Transmitted Infection	MonkeyPox
0	P3574	Swollen Lymph Nodes	0	1	1	1	1	0	1	0	positiv
1	P22261	Swollen Lymph Nodes	0	1	0	1	1	1	1	1	positiv
2	P15466	Fever	0	0	0	1	1	0	0	0	positiv
3	P1238	Fever	0	1	1	1	1	0	1	1	positiv
4	P244	Swollen Lymph Nodes	1	1	0	1	1	1	0	0	positiv

Figure 2. Encoding Kategorik Data

2. One-Hot Encoding Categorical features such as Systemic Illnesses with more than two categories require one-hot encoding. This technique produces several new columns representing each unique category in binary form (0/1).

Kolom boolean/biner berhasil dikonversi ke numerik (0/1).

	Patient_ID	Systemic Illness	Rectal Pain	Sore Throat	Penile Oedema	Oral Lesions	Solitary Lesion	Swollen Tonsils	HIV Infection	Sexually Transmitted Infection	MonkeyPox
0	P3574	Swollen Lymph Nodes	0	1	1	1	1	0	1	0	positiv
1	P22261	Swollen Lymph Nodes	0	1	0	1	1	1	1	1	positiv
2	P15466	Fever	0	0	0	1	1	0	0	0	positiv
3	P1238	Fever	0	1	1	1	1	0	1	1	positiv
4	P244	Swollen Lymph Nodes	1	1	0	1	1	1	0	0	positiv

Figure 3. Encoding Boolean Data

3. Target
Encoding In the monkeypox target column, a label encoding process is performed by mapping categories to numeric values. "Negative" values are mapped to 0, while "positive" values are mapped to 1. This encoding is necessary to facilitate the model in learning the classification patterns between the two classes.

Kolom 'MonkeyPox' berhasil di-encode.

	Patient_ID	Rectal Pain	Sore Throat	Penile Oedema	Oral Lesions	Solitary Lesion	Swollen Tonsils	HIV Infection	Sexually Transmitted Infection	MonkeyPox	Systemic_Illness_Fever	Systemic_Illness_Muscle Aches and Pain	Systemic_Illness_Swollen Lymph Nodes
0	P3574	0	1	1	1	1	0	1	0	1	False	False	True
1	P22261	0	1	0	1	1	1	1	1	1	False	False	True
2	P15466	0	0	0	1	1	0	0	0	1	True	False	False
3	P1238	0	1	1	1	1	0	1	1	1	True	False	False
4	P244	1	1	0	1	1	1	0	0	1	False	False	True

Figure 4. Target Encoding

Feature Selection

In the feature selection stage, a chi-squared test approach is used to assess the relevance of each feature to the MonkeyPox target variable. Categorical features that have been encoded in binary form are applied to the Chi-square test to measure the correlation/relationship of features with the target class. The initial step of this method is to build a contingency table between the features and the target variable. Then, a Chi-Squared test is performed to obtain the test statistic and p-value of each feature. The test results are then sorted by p-value to facilitate the analysis of the significance level of each feature. A p-value <0.5, which has been set as a threshold, is considered to have a significant relationship

with the target variable and is declared relevant. Conversely, features with a p-value greater than 0.5 are considered less relevant and are removed from the dataset.

	MonkeyPox
MonkeyPox	1.000000
Systemic_Illness_Muscle Aches and Pain	0.199703
HIV Infection	0.119110
Systemic_Illness_Fever	0.098724
Rectal Pain	0.098636
Systemic_Illness_Swollen Lymph Nodes	0.091971
Sexually Transmitted Infection	0.091227
Sore Throat	0.071119
Oral Lesions	0.039803
Penile Oedema	0.038008
Solitary Lesion	0.016902
Swollen Tonsils	0.010401

Figure 5. Feature Selection Results

From this table, it can be seen that each feature has a different value. Of the 11 features whose correlation was checked, there were 4 features that had a low p-value. Therefore, the features were manually removed based on the predetermined p-value.

Split Data

The dataset was divided into two parts using a train-test split technique: 80% of the data was for training and 20% for testing, with 6,400 positive and 1,600 negative data categories for training. This proportion was chosen to allow the model to obtain a larger amount of training data and thus be able to learn patterns more optimally, given the dataset's diverse clinical symptom feature complexity.

Resampling

With a large amount of data, and an unbalanced number of class categories, it is necessary to use the SMOTE (Synthetic Minority Over-sampling Technique) technique to balance the number of samples between positive and negative classes. to avoid bias towards the majority class. SMOTE is applied to the training data (x_train, y_train) by synthesizing new samples in the minority class through interpolation between similar data. After this process, the class distribution in the training data is balanced with the target class number of MonkeyPox 6400/6400.

Modeling

K-Nearest Neighbors (KNN)

k-nearest neighbors (KNN) model, the model is built using the default parameters used in distance-based classification research. The main parameters used are:

1. N_neighbors = 5
   The number of nearest neighbors used to determine the class of a sample. This value is a standard setting often used as a starting point in KNN modeling.
2. Distance (p) = 2
   p = 2 in the distance metric used, The value p = 2 indicates that the model uses Euclidean distance, which is the most commonly used metric in KNN because it is able to measure the closeness between data geometrically in multidimensional space. The model's performance visualization for $k = 5$and $p = 2$can be seen in Figure 6.

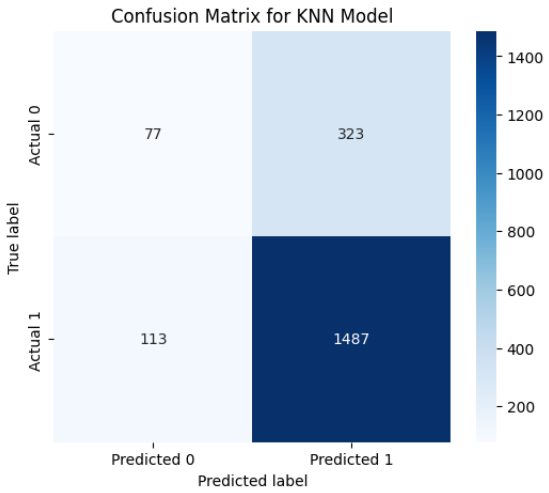


Figure 6. KNN

The confusion matrix provides a detailed visualization of the model's prediction distribution, where:
a) True Negative (TN): 77
b) False Positive (FP): 323
c) False Negative (FN): 113
d) True Positive (TP): 1,487

The evaluation of the K-Nearest Neighbors (KNN) model yielded an accuracy of 0.78 (78%). The precision was 0.82 (82%), indicating that the majority of the positive predictions made by the model were correct, resulting in a relatively low number of false positives. Meanwhile, the recall was 0.93 (93%), demonstrating that the model was able to identify almost all positive cases in the dataset, minimizing the risk of missing monkeypox cases. The combination of precision and recall produced an F1-score of 0.87 (87%), confirming that the model maintains a well-balanced performance in detecting and verifying positive cases.

Random Forest

In the classification stage, the Random Forest model was constructed using several parameters, including random_state(42) to ensure the training process is reproducible, max_depth(none) to limit the maximum depth of the trees and maintain consistent experimental results, as well as other default parameters such as n_estimators (100), min_samples_split(2), and min_samples_leaf(2). After completing the data pre-processing stage, the Random Forest model was evaluated using the pre-defined parameters. To better understand the model's performance, a visualization was created using the confusion matrix, as shown in Figure 7.

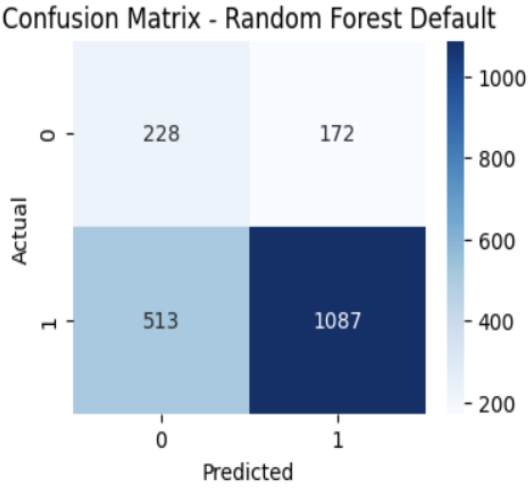


Figure 7 Random Forest

The confusion matrix provides a detailed visualization of the model's prediction distribution as follows:

1. Confusion matrix:
   a) True Negatives (TN): 228
   b) False Positives (FP): 172
   c) False Negatives (FN): 513
   d) True Positives (TP): 1087

The evaluation of the Random Forest model yielded an accuracy of 0.65 (65%), indicating that the model is able to correctly classify the majority of the data, although there is still room for improvement. A precision of 0.86 (86%) demonstrates that most of the positive predictions made by the model are correct, resulting in a relatively low number of false positives. Meanwhile, a recall of 0.68 (68%) reflects the model's ability to capture a majority of the positive cases, although it is not as high as the precision. The combination of these metrics produces an F1-score of 0.76 (76%), reflecting a balanced performance between prediction accuracy and the model's capability to detect positive cases.

K-Nearest Neighbors (KNN) With PSO

In the optimization process of the k-nearest neighbors (KNN) algorithm, there are three main parameters that are optimized. n_neighbors: 18, p : 1 , weights: distance. These results indicate that the KNN model achieves the best performance when using 18 nearest neighbors in the classification process. A p-value of 1 indicates that the model uses Manhattan distance as a distance measure which is considered more appropriate to describe the distribution of data in this dataset than Euclidean. Meanwhile, the use of weights = " uniform " indicates that each neighbor is weighted based on its distance, where closer neighbors have a greater influence on the classification decision than more distant neighbors. The following are the best parameters of KNN optimized with 40 iterations:

```
=== Hyperparameter Terbaik Hasil Optimasi PSO ===
n_neighbors : 18
p           : 1
weights     : uniform
```

Figure 8. KKN

The optimization results showed that k = 18 provided the best performance, meaning the model achieved a balance between variance and bias. In other words, k = 18 provided a fairly stable representation for the model in determining the class of each new sample. After obtaining optimal KNN parameters, the model was retested using the optimal parameters with an accuracy of 80%, a precision of 81%, a recall of 93%, and an f1-score of 89%.

To better understand the model performance, visualization was performed using the following confusion matrix.

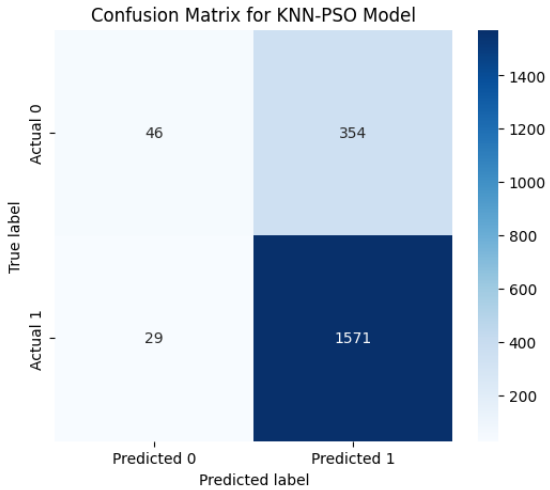


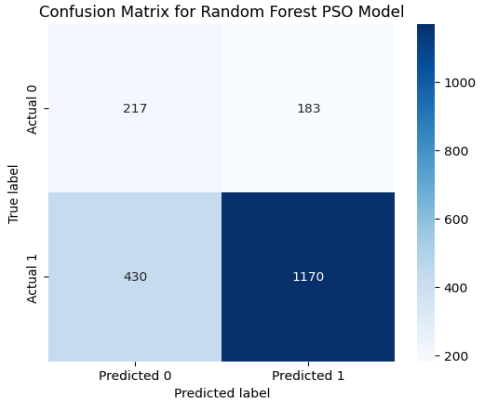
Figure 9. KNN PSO Optimization

The model evaluation results show a significant performance improvement compared to the model without optimization. An accuracy of 80.90% indicates that the model is capable of better and more consistent classification. A precision value of 81% indicates that most of the positive predictions generated by the model are correct, resulting in a relatively low false positive rate. A high recall of 93% demonstrates the model's ability to capture almost all positive cases, making it highly effective in detecting Monkeypox cases. The combination of precision and recall resulted in an F1-score of 89%, illustrating a strong performance balance between the model's accuracy and sensitivity.

Random Forest With PSO

In this study, PSO was used to optimize four main parameters of random forest, namely n_estimators, max_depth, min_samples_split, and min_samples_leaf . Each parameter was given a specific search range by determining lower bounds (50, 2, 2, 1) and upper bounds (150, 8, 6, 3). The optimization process was run for 40 iterations, where each particle in the swarm carried a combination of the values of the four parameters as its position in the search space. At each iteration, PSO evaluated the performance of the resulting model using an objective function. Then the particle position was updated based on the best individual experience (pBest) and the best experience of the entire swarm (gBest), so that the particles gradually moved towards the most optimal parameter solution. The obtained matrix values showed an increase in several matrices with an accuracy value of 0.69(69%), precision 0.86(86%), recall 0.73,(73%), and f1 score 0.79 (79%).

To better understand the model performance, visualization was performed using the following confusion matrix.



Picture 10 Random Forest PSO Optimization

The evaluation results show a more stable performance improvement compared to the default model. An accuracy of 69% indicates that the model is able to correctly classify data in most cases, although the improvement is not significant. A high precision of 86% indicates that the model's positive predictions are mostly correct, thus maintaining a low false positive rate . A recall of 73% indicates that the model is able to consistently recognize the majority of positive cases, thus reducing the risk of missing positive cases. The combination of these two metrics results in an F1-score of 79%, indicating a good balance between accuracy and sensitivity.

Evaluation results

After applying optimization using particle swarm optimization (PSO), the model obtained optimal results for both KNN and random forest parameters. The optimal parameters obtained by the model were retested. The comparison results can be seen in the Model Comparison Table below.

Table 1. KKN

Model	Akurasi	Presisi	Recall	F1-Score
KNN	78%	82%	93%	87%
KNN + PSO	80%	81%	87%	89%
Random Forest	65%	86%	68%	76%
RF + PSO	69%	86%	73%	79%

The performance comparison of the K-Nearest Neighbor (KNN) model before and after Particle Swarm Optimization (PSO) demonstrates variations across several evaluation metrics. In terms of accuracy, the KNN model shows an improvement from 78% to 80% after PSO optimization. This increase indicates that PSO-based parameter optimization enhances the model's overall classification capability. However, with respect to precision, the standard KNN model achieves a value of 82%, while the KNN optimized with PSO slightly decreases to 81%. This decline suggests that the PSO-optimized model tends to produce a slightly higher number of false positives. In contrast, the recall metric of the standard KNN model is higher at 93% compared to 87% for the KNN+PSO model, indicating that the non-optimized KNN is more effective in capturing all positive instances. Nevertheless, the most notable improvement is observed in the F1-score, which increases from 87% to 89%. This result demonstrates that PSO optimization improves the overall balance between precision and recall, leading to more stable model performance.

Similarly, the application of PSO optimization to the Random Forest model also results in performance improvements. In terms of accuracy, the standard Random Forest model achieves an accuracy of 65%, which increases to 69% after PSO optimization. This finding suggests that PSO contributes to improving the model's overall predictive accuracy. Regarding precision, both the standard Random Forest and the PSO-optimized Random Forest obtain the same value of 86%, indicating that PSO optimization does not significantly affect the model's ability to correctly predict positive instances. However, for the recall metric, the PSO-optimized Random Forest shows an improvement from 68% to 73%, indicating that PSO helps the model capture more true positive cases and reduce false negatives. This improvement is also reflected in the F1-score, which increases from 76% to 79%, demonstrating an enhanced balance between precision and recall following PSO optimization.

CONCLUSION

Based on results implementation, testing, and analysis to K-Nearest Neighbors (KNN) and Random Forest algorithms, both before and after done optimization using Particle Swarm Optimization (PSO), can concluded that KNN algorithm in general show better performance superior compared to Random Forest on the MonkeyPox dataset which is not balanced, especially in detect class positive, which is reflected from higher recall and F1-score values high. Implementation proven PSO optimization capable increase performance second algorithm, however most significant improvement happen on the KNN model, especially on metric accuracy, recall, and F1-score. The KNN model with PSO optimization to become the best model in study This with accuracy of 80%, class recall positive of 93% and F1-score of 089% which indicates effectiveness tall in detect case positive MonkeyPox as well as minimize error false negative, which is very crucial in context medical. Meanwhile that, Random Forest shows level greater precision tall And more capabilities Good in recognize class negative, but relative recall value low make this model less than optimal for application detection disease that requires sensitivity tall.

REFERENCES

Bengesi, S., Oladunni, T., Olusegun, R., & Audu, H. (2023). A Machine Learning-Sentiment Analysis on Monkeypox Outbreak: An Extensive Dataset to Show the Polarity of Public Opinion From Twitter Tweets. IEEE Access, 11, 11811–11826. doi: 10.1109/ACCESS.2023.3242290

Cunningham, P., & Delany, S. J. (2021). K-Nearest Neighbour Classifiers-A Tutorial. In ACM Computing Surveys (Vol. 54, Issue 6). Association for Computing Machinery. doi: 10.1145/3459665

Darmawan, R., & Surahmat, A. (2022). *Optimalisasi* Support Vector Machine (SVM) *Berbasis* Particle Swarm Optimization (PSO) *Pada Analisis Sentimen Terhadap* Official Account *Ruang Guru Di* Twitter. In *Jurnal Kajian Ilmiah* (Vol. 22, Issue 2). Retrieved from http://ejurnal.ubharajaya.ac.id/index.php/JKI

Freitas, D., Lopes, L. G., & Morgado-Dias, F. (2020). Particle Swarm Optimisation: A historical review up to the current developments. In Entropy (Vol. 22, Issue 3). MDPI AG. doi: 10.3390/E22030362

Gad, A. G. (2022). Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. Archives of Computational Methods in Engineering, 29(5), 2531–2561. doi: 10.1007/s11831-021-09694-4

Hadjaidji, E., Korba, M. C. A., & Khelil, K. (2025). Improving detection of Parkinson's disease with acoustic feature optimization using particle swarm optimization and machine learning. Machine Learning: Science and Technology, 6(1). doi: 10.1088/2632-2153/adadc3

Hamdan, A., & Ekmekci, D. (2024). Design of Monkeypox Disease Diagnosis Model Using Classical Machine Learning Algorithm. Journal of Soft Computing and Artificial Intelligence. doi: 10.55195/jscai.1461849

Hoang Huong, L., Hoang Khang, N., Nhat Quynh, L., Huu Thang, L., Minh Canh, D., & Phuoc Sang, H. (n.d.). A Proposed Approach for Monkeypox Classification. In IJACSA) International Journal of Advanced Computer Science and Applications (Vol. 14, Issue 8). Retrieved from www.ijacsa.thesai.org

Homepage, J., Cholil, S. R., Handayani, T., Prathivi, R., & Ardianita, T. (2021). IJCIT (Indonesian Journal on Computer and Information Technology) *Implementasi Algoritma Klasifikasi* K-Nearest Neighbor (KNN) *Untuk Klasifikasi Seleksi Penerima Beasiswa*. In IJCIT (Indonesian Journal on Computer and Information Technology) (Vol. 6, Issue 2).

Jain, A., Somwanshi, D., Joshi, K., & Bhatt, S. S. (2022). A Review: Data Mining Classification Techniques. Proceedings of 3rd International Conference on Intelligent Engineering and Management, ICIEM 2022, 636–642. doi: 10.1109/ICIEM54221.2022.9853036

Jatmiko, Y. A., Padmadisastra, S., & Chadidjah, A. (2019). *Analisis Perbandingan Kinerja* Cart Konvensional, Bagging *Dan Random* Forest *Pada Klasifikasi Objek: Hasil Dari Dua Simulasi. Media Statistika*, 12(1), 1. doi: 10.14710/medstat.12.1.1-12

Kesehatan, J., Meditory, S., Gumandang, H. P., *Kedokteran*, F., & Lampung, U. (n.d.). Monkeypox Disease: *Wabah Multi-Nasional* Monkeypox Disease: *Multi-Nasional* Outbreak. Retrieved from https://jurnal.syedzasaintika.ac.id

Mangukiya, M. (2022). Breast Cancer Detection with Machine Learning. International Journal for Research in Applied Science and Engineering Technology, 10(2), 141–145. doi: 10.22214/ijraset.2022.40204

Nur Afifah, I. A., & Nurdiyanto, H. (2023). Data Mining Clustering *Dalam Pengelompokan Buku Perpustakaan Mengunakan Algoritma* K-Means. JIPI *(Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 8(3), 802–814. doi: 10.29100/jipi.v8i3.3891

Octaviani, A., & Dewi, P. (2020). Big *Data di Perpustakaan dengan Memanfaatkan Data* Mining. ANUVA, 4(2), 223–230.

Rahmaddeni, Wulandari Denok, Ramadhan Gilang, & Sari Retno. (2024). Machine Learning (R. M. Sari, Ed.). Serasi Media Teknologi.

Rahman Ramli, A., & Khalil Gibran, A. (n.d.). *Analisis Kinerja Algoritma* Machine Learning *Untuk Deteksi Penyakit Daun Teh Dengan* Particle Swarm Optimization. Retrieved from https://www.kaggle.com/code/arafathhmm/tea-leaf-

Raut, R., Borole, Y., Patil, S., Khan, V., & Takale, D. G. (n.d.). Skin Disease Classification Using Machine Learning Algorithms. doi: 10.14704/nq.2022.20.10.NQ55940

Salman, K., & Sonuc, E. (2021). Thyroid Disease Classification Using Machine Learning Algorithms. Journal of Physics: Conference Series, 1963(1). doi: 10.1088/1742-6596/1963/1/012140

Setegn, G. M., & Dejene, B. E. (2025). Explainable AI for Symptom-Based Detection of Monkeypox: a machine learning approach. BMC Infectious Diseases, 25(1). doi: 10.1186/s12879-025-10738-4. *Tampilan Gambaran Umum Algoritma* Random Forest. (n.d.).

Wang, X., & Lun, W. (2023). Skin Manifestation of Human Monkeypox. In Journal of Clinical Medicine (Vol. 12, Issue 3). MDPI. doi: 10.3390/jcm12030914

Wardana, A., Ula, M., & Retno, S. (n.d.). *Senastika Universitas Malikussaleh Analisis Kinerja Kombinasi Algortihma* K-Nearest Neighbor (Knn) Dan Principal Componen Analysis (Pca) *Pada Klasifikasi Gambar Spesies Burung*.