Jurnal Ilmiah KURSOR
Menuju Solusi Teknologi Informasi
Vol. 12, No. 4, Desember 2024

ISSN 0216 – 0544
e-ISSN 2301– 6914

EVALUATION OF PARTICLE SWARM ALGORITHM MODIFICATIONS ON SUPPORT VECTOR MACHINE HYPERPARAMETER OPTIMIZATION TUNING FOR RAIN PREDICTION

[a]Aina Latifa Riyana Putri, [b]Joko Riyono, [c]Christina Eni Pujiastuti, [d]Supriyadi

[a]Telkom University, Jl. DI Pandjaitan No. 128 Purwokerto
[b,c,d]Universitas Trisakti, Jl Kiyai Tapa No 1 Grogol-Jakarta
E-mail: ainaqp@telkomuniversity.ac.id,
jokoriyono@trisakti.ac.id,christina.eni@trisakti.ac.id,supri@trisakti.ac.id

Abstract

The Particle Swarm Optimization (PSO) algorithm, though simple and effective, faces challenges like premature convergence and local optima entrapment. Modifications in the PSO structure, particularly in acceleration coefficients (C_1 and C_2), are proposed to address these issues. Techniques like Time Varying Acceleration Coefficients (TVAC), Sine Cosine Acceleration Coefficients (SCAC), and Nonlinear Dynamics Acceleration Coefficients (NDAC) have been implemented to enhance convergence speed and solution quality. This research evaluates various PSO modifications for improving convergence and robustness in rainfall potential prediction using Support Vector Machine (SVM) classification. The UAPSO-SVM algorithm C=0.82568 and γ=0.01960 excels in initial exploration, discovering more optimal global solutions with smaller variability. In contrast, TVACPSO-SVM shows gradual improvement but requires more iterations for stability, while SBPSO-SVM achieves the fastest convergence at iteration 14 but risks overfitting. Robustness analysis reveals all PSO-SVM variants maintain stable performance despite variations in dataset subset sizes, with accuracy stabilizing after a spike at 20%.. Therefore, PSO modifications enhance convergence speed and resilience to data fluctuations, improving their effectiveness for rainfall prediction.

Key words: Convergence, Modified Particle Swarm Optimization, Robustness, Support Vector Machine.

INTRODUCTION

The Particle Swarm Optimization (PSO) algorithm, first proposed by Kennedy and Eberhart in 1995, is a population-based optimization algorithm that utilizes the concept of "swarm intelligence." This algorithm is inspired by the social behavior of bird flocks or fish schools in searching for food [1]. The Particle Swarm Optimization (PSO) algorithm works by optimizing a group of candidate solutions (particles) through the solution space based on the best positions and velocities found. The Particle Swarm Optimization (PSO) algorithm has several advantages mentioned in research [2], such as being able to explore the search space more intelligently due to its population-based approach, resulting in more efficient computation and better performance in handling high-dimensional problems [3] compared to other algorithms like Grid Search [4], Random Search [5], and Bayesian Optimization [6].

Although the Particle Swarm Optimization (PSO) algorithm is known for its simplicity and effectiveness, it also has some disadvantages. In research by [7], it is mentioned that the PSO algorithm tends to generate divergent paths, indicating that the algorithm does not always converge adequately. Particles in the PSO algorithm tend to converge prematurely to a stable state (premature convergence), and solutions get trapped in local optima [8], [9], [10], [11]. Convergence behavior in a PSO algorithm can be achieved when there is a proper balance between exploration and exploitation of the search space, guiding particles toward the global optimum solution [12]. To address these issues, the PSO algorithm has undergone many structural modifications over time, especially regarding the acceleration coefficients (C_1 and C_2), which play a role in the exploration and exploitation process of the search space [13], improving the convergence speed and the quality of the solutions obtained. Research by [14] proposed a modification to the acceleration coefficients with unbalanced values (C_1 = 0.5 and C_2 = 2.5), which resulted in better convergence for the given problem. Research by [15] found that the acceleration coefficients should change with each iteration to approach the global optimum solution. This has been implemented in methods like Time Varying Acceleration Coefficients [16], Sine Cosine Acceleration Coefficients [17], Nonlinear Dynamics Acceleration Coefficients [17], and Sigmoid-based Acceleration Coefficients [13].

This study will analyze the performance of several modifications to the Particle Swarm Optimization (PSO) algorithm. These modifications will be used for optimizing the hyperparameters (C and Gamma) of the Support Vector Machine algorithm in the case of rain potential prediction. The Support Vector Machine (SVM) algorithm is one of the algorithms used to solve classification problems by separating different classes [18]. The Support Vector Machine (SVM) algorithm is based on the concept of finding the best hyperplane to separate data into different classes. The Support Vector Machine (SVM) algorithm has a set of configuration variables that can be adjusted by the user before training a model, and these variables can have various effects on the resulting model training [18]. These configuration variables are known as hyperparameters. Hyperparameters are used to optimize the model by controlling parameters (weights and biases in the case of the Support Vector Machine algorithm), which will affect the speed and quality of model training. According to [19] and [18], the Support Vector Machine (SVM) algorithm has several hyperparameters such as C and Gamma (γ). Choosing the right combination of hyperparameter values is a challenge. In cases with the same dataset size, different hyperparameter values can yield very good predictions or potentially very poor ones [20], which affects the performance of the Support Vector Machine (SVM) algorithm [21] and [22]. As [23] wrote, a high C hyperparameter will risk overfitting the Support Vector Machine (SVM) model to the training samples, and similarly, the gamma (γ) hyperparameter will also affect the decision boundary. Choosing the right hyperparameter values will provide the best and most optimal prediction model [24]. Therefore, finding an effective and efficient way to determine the optimal hyperparameter values, while ensuring prediction performance without overfitting for the dataset used to train the Support Vector Machine (SVM) algorithm, is an important issue to research [20].

The aim of this research is to evaluate how well the developments in the Particle Swarm Optimization (PSO) algorithm modifications can overcome the weaknesses of the original PSO algorithm (C_1 = 2.0 and C_2 = 2.0) in terms of convergence speed and robustness across different data variations for rain potential prediction classification problems. This study seeks to answer the following questions: How do various PSO modifications improve convergence speed and robustness in optimizing SVM hyperparameters for rainfall prediction? and Which PSO modification achieves the best performance across different dataset sizes?. By understanding the shortcomings and limitations of the existing modifications, future research can further develop more efficient and robust PSO algorithm modifications for optimizing the Support Vector Machine hyperparameters for rain prediction, leading to more accurate and reliable predictions.

MATERIAL AND METHODS

In this research, the data used are secondary data with a literature study to gather sources related to the research topic, obtained from previous research journals. Additionally, data from the Online Database Center of the Meteorology, Climatology, and Geophysics Agency (BMKG), specifically from the Maritime Meteorology Station in Tanjung Priok, North Jakarta (http://dataonline.bmkg.go.id/home), were used. BMKG data were selected due to their reliability, comprehensive coverage of meteorological parameters, and their relevance to rainfall prediction, which aligns with the research objectives. The data contains daily climate data from the area, consisting of 1,410 data entries from January 1, 2019, to November 10, 2022, with 10 variables, as seen in Appendix 1. The variables used in this study are as follows.

Table 1. Identification of Variables

No.	Variable	Description
1	ddd_x	Wind direction at maximum speed
2	ddd_car	Most frequent wind direction
3	ff_x	Maximum wind speed
4	ff_avg	Average wind speed
5	RH_avg	Average humidity
6	ss	Duration of sunlight
7	Tx	Maximum Temperature
8	Tn	Minimum Temperature
9	Tavg	Average Temperatue
10	RR	Rainfall

To evaluate the performance of the Particle Swarm Optimization (PSO) modifications in optimizing Support Vector Machine (SVM) hyperparameters, the study used key validation metrics, including average and standard deviation to measure variability in the results, and convergence speed (iterations to achieve stability). These metrics ensure a robust evaluation of the algorithm's ability to achieve optimal solutions while maintaining stability across various dataset sizes. Parameter selection for PSO (e.g., population size, maximum iterations, and acceleration coefficients (C_1) and (C_2) was based on values commonly used in prior studies, with adjustments to suit the dataset characteristics.

This section explains the research process to evaluate the extent to which modifications of the Particle Swarm Optimization (PSO) algorithm can address the weaknesses of the original Particle Swarm Optimization (PSO) algorithm (C_1 = 2.0 and C_2 = 2.0) in terms of convergence speed and robustness against variations in classification data cases for the Support Vector Machine algorithm in predicting rainfall potential. The steps are as follows:

Literature Review

The first step in this research is to study the fundamental concepts and related literature on Particle Swarm Optimization (PSO) and its modifications. The previous research will be reviewed to understand the various approaches that have been proposed, including studies on the original Particle Swarm Optimization algorithm (C_1 = 2.0 dan C_2 = 2.0) and five modified variations to address the convergence issues related to PSO.

a) Unbalanced Acceleration Coefficient PSO (UACPSO)
The research conducted by [14] developed modifications to the cognitive and social parameters (c_1 and c_2) to accelerate the convergence of the Particle Swarm Optimization (PSO) algorithm. In the tests conducted, two sets of parameters were used: Balanced Acceleration ($C_1 = 1.5$ and $C_2 = 1.5$) and Unbalanced Acceleration Coefficient ($C_1 = 0.5$ and $C_2 = 2.0$). The test results indicated that using the Unbalanced Acceleration Coefficient accelerated convergence.

b) Time Varying Acceleration Coefficients PSO (TVAC-PSO)
This research [16] also proposed modifications to the cognitive and social parameter values (C_1 and C_2) based on [15] study, which utilized a second-order linear decrease in both acceleration coefficients over time to achieve better solutions. The modification involved decreasing the cognitive component (C_1) and increasing the social component (C_2) over time, as described by the following equations:

$$C_1 = (C_{1f} - C_{1i})\frac{iter}{MAXITR} + C_{1i} \quad (1)$$

$$C_2 = \left(C_{2f} - C_{2i}\right)\frac{iter}{MAXITR} + C_{2i} \quad (2)$$

where $C_{1f}, C_{1i}, C_{2f}, C_{2f}$ are constants, $iter$ is the current iteration number, and $MAXITR$ is the predetermined maximum number of iterations. Simulations were conducted to find the optimal ranges for C_1 and C_2. The results showed that changing the values from 2.5 to 0.5 for C_1 and from 0.5 to 2.5 for C_2 across the entire search range led to better optimal solutions.

c) Sine Cosine Acceleration Coefficients PSO (SCAC-PSO)
Inspired by TVAC-PSO and [15], this study also proposed using time-varying acceleration coefficients where individuals in the population are expected to explore the entire search space during the early stages of the optimization process while enhancing convergence capabilities toward the global optimum in the later stages. The study proposed sine cosine acceleration coefficients (SCAC) as a new parameter adjustment strategy for cognitive and social components [17]:

$$C_1 = \partial \; x \sin\left(\left(1 - \frac{iter}{MAXITR}\right) x \; \frac{\pi}{2}\right) + \delta (3)$$

$$C_2 = \partial \; x \cos\left(\left(1 - \frac{iter}{MAXITR}\right) x \; \frac{\pi}{2}\right) \delta \quad (4)$$

where ∂ and δ are constants (∂ = 2; δ = 0.5). SCAC is more effective than TVAC at maintaining a balance between extensive exploration in the initial phases and steady convergence towards the end. The PSO algorithm modified with these sine cosine acceleration coefficients is referred to as PSO-SCAC.

d) Nonlinear Dynamics Acceleration Coefficients PSO (NDAC-PSO)
This research modifies the PSO acceleration coefficients using a nonlinear dynamic (NDAC) method as a parameter updating mechanism to adjust the cognitive component (C_1) and social component (C_2). In the PSO algorithm, individuals in a swarm are expected to explore the solution space in the early stages of the search. Additionally, in the subsequent search phases, Improving local search ability is essential for efficiently and rapidly finding the global optimum. By incorporating the NDAC mechanism into the PSO algorithm, this method successfully balances global and local search effectiveness, thus the modified algorithm is called PSO-NDAC. The equations representing NDAC are as follows [17]:

$$C_1 = -\left(C_{1f} - C_{1i}\right)\left(\frac{iter}{MAXITR}\right)^2 + C_{1f} \quad (5)$$

$$C_2 = C_{1i} \; x \left(1 - \frac{iter}{MAXITR}\right)^2 + C_{1f} \; x \; \frac{iter}{MAXITR} \quad (6)$$

where C_{1f} and C_{1i} are positive constants (with values $C_{1f} = 2.5$ and $C_{1i} = 0.5$), iter is the current iteration, and $MAXITR$ is the maximum iteration. The cognitive component (C_1) and social component (C_2) change from 2.5 to 0.5 and from 0.5 to 2.5, respectively, as the iterations progress.

e) Sigmoid-based Acceleration Coefficient PSO (SAC-PSO)
This research [13] presents a sigmoid-based acceleration coefficient (SBAC) with the following equations:

$$C_1 = \frac{1}{1+e^{\left(-\lambda \cdot \frac{iter}{MAXITR}\right)}} + 2\left(C_{1f} - C_{1i}\right)\left(\frac{iter}{MAXITR} - 1\right)^2 \quad (7)$$

$$C_2 = \frac{1}{1+e^{\left(-\lambda \cdot \frac{iter}{MAXITR}\right)}} + \left(C_{1f} - C_{1i}\right)\left(\frac{iter}{MAXITR}\right)^2 \quad (8)$$

where λ is a control parameter used to regulate the sigmoid-based acceleration coefficient (λ = 0.0001), and C_{1f} and C_{1i} are two positive constants ($C_{1f} = 2.5$ and $C_{1i} = 0.5$). The values of C_1 decrease from 2.5 to 0.5 while C_2 increases from 0.5 to 2.5 under the conditions λ = 0.0001, $C_{1f} = 2.5$, and $C_{1i} = 0.5$.

Input Data and Data Preprocessing

The data used in this study has been mentioned in the previous section. The next step is to preprocess the data. This involves data cleaning to remove missing values or outliers, as well as normalizing or standardizing the data to make it suitable for use in the Support Vector Machine algorithm. This step is crucial to ensure that the data used in the experiment is of high quality and ready for analysis.

Implementation of Particle Swarm Optimization (PSO) Variations

The third step is to implement five modified versions of the PSO algorithm with various acceleration coefficients. Each PSO variation will be implemented according to the specifications described in the literature. This implementation includes parameter adjustments and the algorithm's structure for each variation, which will then be used to optimize the SVM parameters.

Testing and Validation

The next step is to determine the SVM parameters to be optimized, such as the parameters C and gamma. Each experiment will be run with a fixed number of iterations and population size for each PSO variant. The convergence results for each iteration of each PSO variation and data subset will be recorded. Additionally, the dataset will be split into several subsets of different sizes, such as 10%, 20%, 30%, 40%, and so on. This is done to test the robustness of each PSO variation against various data sizes.

Analysis of Results

After the experiment is complete, the convergence speed of each PSO variation will be evaluated based on the number of iterations required to reach the optimal solution. In addition, the performance of each PSO variation on different data subsets will be evaluated to determine their robustness. This analysis will include evaluating how well each PSO variation can handle different data sizes in the classification task using SVM. Based on the convergence speed and robustness analysis, conclusions will be drawn about the effectiveness and reliability of each PSO variation.

RESULT AND DISCUSSION

Despite its simplicity and effectiveness, the Particle Swarm Optimization (PSO) algorithm has a known drawback regarding convergence. In [7] study, the convergence of PSO toward a global optimal solution is discussed as follows:

Theorem 2.1. A particle in the Particle Swarm Optimization algorithm converges to a stable point, which is $\frac{c_1 p_{(i,lb)}^{(t)} + c_2 p_{gb}^{(t)}}{c_1 + c_2}$, if and only if $max\{||\lambda_1||, ||\lambda_2||\} < 1$, where λ_1 and λ_2 are eigenvalues representing the dynamics of the simple PSO system with inertia (ω) [7].

In his analysis, the trajectory of the particle with inertia approaches a stable point, which is the weighted average of $p_{(i,lb)}^{(t)}$ and $p_{gb}^{(t)}$, i.e., $\frac{c_1 p_{(i,lb)}^{(t)} + c_2 p_{gb}^{(t)}}{c_1 + c_2}$, if and only if $\max\{\|\lambda_1\|, \|\lambda_2\|\} < 1$ is satisfied. Subsequently, the parameter selection for the Particle Swarm Optimization algorithm is made based on these findings. It is required that

$0 < c_1 + c_2$

$\frac{c_1 + c_2}{2} - 1 < \omega.$

If $\omega, c_1,$ and c_2 are chosen, then the system ensures convergence $(\max\{\|\lambda_1\|, \|\lambda_2\|\} < 1)$ to a stable point.

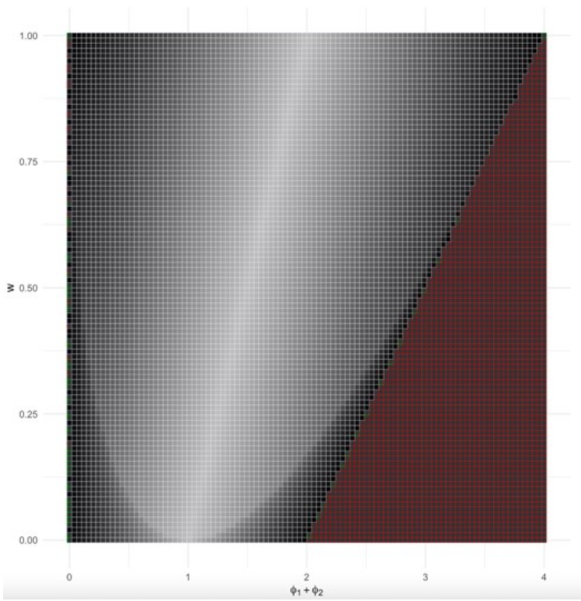

Fig 1. Visualization of φ_1, φ_2, and ω values leading to convergence and divergence

Figure 1 illustrates the visualization of values c_1, c_2, and w that influence the convergence or divergence of the algorithm. The plot is created by sampling c_1, c_2, and values of w are obtained on a grid by using 100 samples horizontally and 100 samples vertically. The red triangle region indicates that $\max\{\|\lambda_1\|, \|\lambda_2\|\} > 1$, suggesting that the particle trajectory will diverge. The green area represents $\max\{\|\lambda_1\|, \|\lambda_2\|\} = 1$, while the area representing $\max\{\|\lambda_1\|, \|\lambda_2\|\} < 1$ is shaded from black to light gray, with lighter colors indicating faster convergence.

In searching for the optimal hyperparameter combination for Support Vector Machine using the original Particle Swarm Optimization, values of $c_1 = 2, c_2 = 2$, and $w = 1$ were used. It is essential to check first whether the original PSO can satisfy the convergence criteria discussed previously. Given that c_1 and c_2 are the upper bounds for φ_1 and φ_2. It can be observed that these values do not meet the convergence criteria for the PSO parameters, as $c_1 + c_2 = 2 + 2 = 4 > 0$ and $\frac{c_1+c_2}{2} - 1 = \frac{2+2}{2} - 1 = 1 = w$. Thus, this seems to imply that the original PSO equation produces a divergent trajectory [7]. This raises concerns about the use of the original PSO in real-world problems. The divergence of the trajectory indicated by equations (1) and (2) suggests that the original PSO does not provide adequate convergence. This weakness of the original PSO algorithm has been discussed extensively, as seen in [8], [9], [10], and [11], where it is stated that particles in the PSO algorithm tend to prematurely converge to a stable state (premature convergence) and solutions get trapped in local optima. In [12], It is also noted that in population-based algorithms like PSO, achieving convergence behavior requires an appropriate balance between exploring and exploiting the search space, thereby driving the particles toward the global optimal solution.

To observe the performance of various PSO modifications proposed by previous studies, experimental tests were conducted using the rainfall data mentioned earlier. The optimization problem of PSO modifications to obtain the SVM hyperparameter values will be defined as follows:

"Given SVM is a Support Vector Machine machine learning algorithm with N hyperparameters. A Support Vector Machine model $\mathcal{M}$, which maximizes some accuracy function $acc(\mathcal{M}|X^{(te)}$ on a given test dataset $X^{(te)}$, is obtained by applying the SVM algorithm using the training dataset $X^{(tr)}$ and solving an optimization problem. The SVM machine learning algorithm can be parameterized by a set of hyperparameters λ (i.e., C and γ), for example, $\mathcal{M} = SVM(X^{(tr)}|\lambda)$. The hyperparameter search aims to find an optimal set of hyperparameters, λ^*, such that the algorithm produces an optimal model $\mathcal{M}^*$ that maximizes $acc(\mathcal{M}|X^{(te)})$.

$$\lambda^* = arg \max_{\lambda} acc\big(\text{SVM}\big(X^{(tr)}\big|\lambda\big)\big|X^{(te)}\big) = arg \max_{\lambda} f(\lambda|\text{SVM}, X^{(tr)}, X^{(te)}, acc) \quad (9)$$

Where f is the objective function; λ is a tuple of hyperparameters (optimization variables); and the sets of datasets $X^{(tr)}$ and $X^{(te)}$ are known."

Specifically, the experimental setup is standardized for comparison purposes as follows: swarm size (N): 40 [11], inertia weight (ω): 0.7298 [7], search space limits are set to C: [0.1, 1.0] and γ: [0.001, 0.9] through trial and error considering computational accuracy and convergence speed. For each test, 30 independent tests [9] are conducted by each PSO variation, and each test is run for 100 iterations. All PSO variations will stop iterating when the maximum number of allowed iterations is reached. The average solution and standard deviation, as shown in the Table 2, are used to evaluate the algorithm's convergence.

Table 2. Average Performance and Standard Deviation of Each PSO Modification

Algorithm	Average		Standard Deviation	
	C	γ	C	γ
UAPSO-SVM	0.82568	0.01960	0.17431	0.00216
TVACPSO-SVM	0.55597	0.06935	0.34138	0.06262
SCACPSO-SVM	0.73396	0.03765	0.25649	0.04168
NDACPSO-SVM	0.63438	0.05577	0.29029	0.05698
SPO-SVM	0.60270	0.05234	0.34854	0.05881
UAPSO-SVM	0.82568	0.01960	0.17431	0.00216

In the Table 2, to obtain a more comprehensive picture of convergence, variability in the results produced by the algorithm must also be considered. Variability refers to the extent of dispersion of variable values from their central tendency in a distribution, reflecting how much the values differ from the central tendency, mainly the mean or average. Measuring variability allows us to see an overview of the variation, range, as well as the heterogeneity or homogeneity of a data group's measurements. Measures of variability include range, mean deviation, standard deviation, and so on. If variability is small, then each score will accurately represent the entire distribution. Conversely, if the sample distribution variability is large, then each score or set of scores does not accurately represent the entire distribution [26].

Standard deviation is one of the most widely used statistical tools. The standard deviation measures the dispersion of data in a sample to determine how far or close the data values are from the mean. The formula for the standard deviation is as follows:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} \quad (10)$$

σ = Standard Deviation
n = Sample Size
$\bar{x}$ = Mean Value

The final standard deviation results for C and Gamma in each PSO modification experiment can be seen in the Table 2.

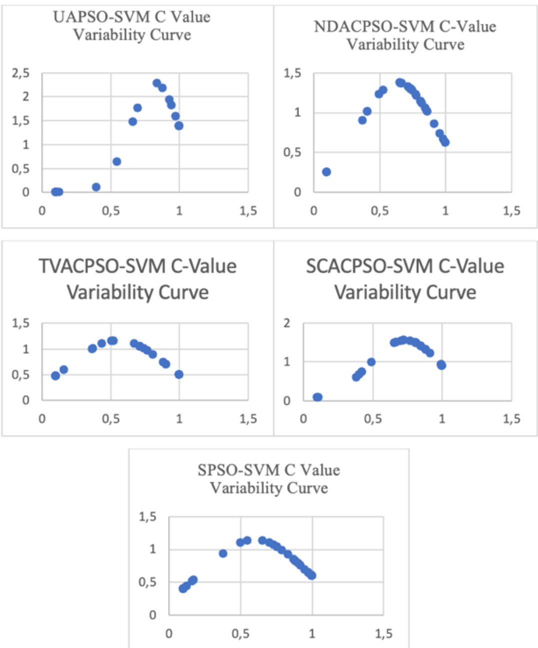

Fig 2. Variability curve

Variability measurement can be illustrated in the form of a curve in the Fig 2. The TVACPSO-SVM and SPSO-SVM algorithms may show the same mean value for the optimal C value. Although the mean values are almost the same, the dispersion of the optimal C value from TVACPSO-SVM is smaller than that of SPSO-SVM. In this case, it can be said that the optimal C value obtained by SPSO-SVM is heterogeneous, while the optimal C value obtained by TVACPSO-SVM is homogeneous. Furthermore, from the figure, it can be seen that the optimal C value obtained by SPSO-SVM is spread quite far from the average optimal C value. In statistical terms, it is said that the optimal C value obtained by SPSO-SVM has greater variability than the optimal C value obtained by VACPSO-SVM. This also reflects that SPSO-SVM has less flexibility in exploring the solution space, which can lead to the risk of overfitting or getting stuck in a local optimum. The optimal C value obtained by UAPSO-SVM has the smallest variability curve compared to other algorithms. This phenomenon indicates that the solution obtained by the algorithm is increasingly closer to the mean value. This reflects that UAPSO-SVM shows an advantage in broad initial exploration, allowing this algorithm to find a more optimal global solution. Therefore, UAPSO-SVM is an indication of good algorithm convergence.

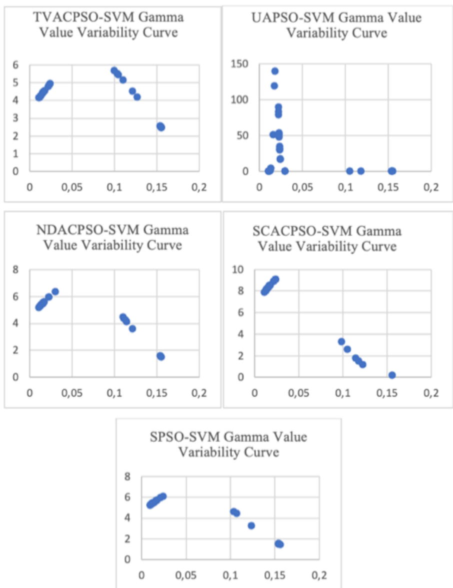

Fig 3. Gamma value variability curve

The variability measurement of the Gamma value can also be illustrated in the form of a curve in the Figure. The NDACPSO-SVM and SPSO-SVM algorithms may also show the same mean value for the optimal Gamma value. Although the mean values are almost the same, the dispersion of the optimal Gamma value from NDACPSO-SVM is larger than that of SPSO-SVM. In this case, it can be said that the optimal Gamma value obtained by SPSO-SVM is heterogeneous, while the optimal Gamma value obtained by NDACPSO-SVM is homogeneous. Furthermore, from the figure, it can be seen that the optimal Gamma value obtained by SPSO-SVM is spread quite far from the average optimal Gamma value. In statistical terms, it is said that the optimal Gamma value obtained by SPSO-SVM has greater variability than the optimal Gamma value obtained by NDACPSO-SVM. This also reflects that SPSO-SVM has less flexibility in exploring the solution space, which can lead to the risk of overfitting or getting stuck in a local optimum. The optimal Gamma value obtained by UAPSO-SVM also has the smallest variability curve compared to other algorithms. The UAPSO-SVM algorithm demonstrated the smallest variability in both C and γ values, as shown in Fig 2 and Fig 3, reflecting its consistent performance across multiple runs. This consistency is critical in scenarios requiring reliable predictions, such as disaster preparedness. Another interesting observation,

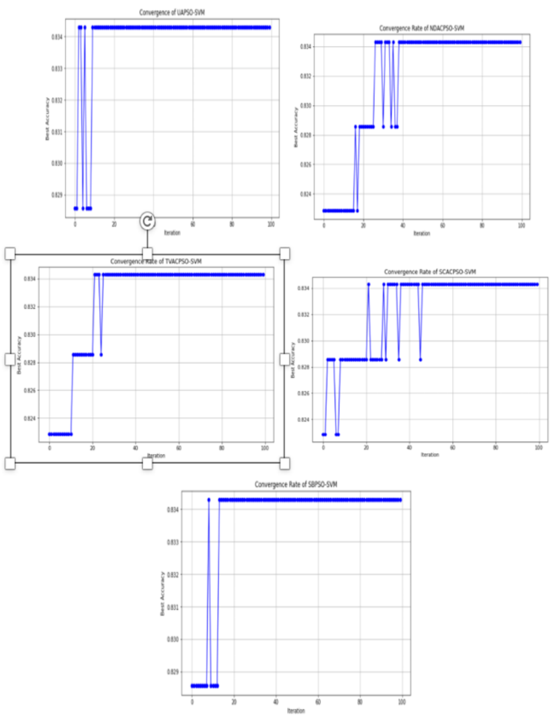

Fig 4. Convergance rate

The five variations of the PSO-SVM algorithm show different convergence graphs in achieving the best accuracy. In practice, the convergence rate provides an understanding when using the algorithm with iterative methods as a tool for calculating numerical approximations. The goal of convergence rate analysis in PSO is to improve the overall performance of PSO, which can drive faster global convergence, higher-quality solutions, and greater robustness. One of the efforts that must be made is to also pay attention to the occurrence of early convergence, which indicates optimization stagnation (as seen in NDACPSO-SVM). UAPSO-SVM reaches optimal accuracy but requires more iterations with more frequent fluctuations before finally stabilizing at the same accuracy (iteration 10), indicating slower convergence. TVACPSO-SVM also experiences gradual improvement and reaches stability after about 25 iterations, slightly slower than UAPSO-SVM. On the other hand, SBPSO-SVM shows the fastest convergence (iteration 14) by achieving optimal accuracy in the early iterations and remaining stable afterward, making it the algorithm with the highest convergence speed among the variations. This makes SBPSO-SVM particularly suited for time-critical tasks, such as real-time weather forecasting. However, its greater variability suggests potential trade-offs in stability over larger or noisier datasets.

These findings are in line with the study by [13], where, when the two PSO parameters (C_1 and C_2) have the same value, the particle's next position is typically between the individual best position ($p_{(i,lb)}^{(t)}$) and the global best position ($p_{gb}^{(t)}$). This results in more iterations being needed for the particle to approach the global best position ($p_{gb}^{(t)}$), affecting the search efficiency, as shown in the following Fig 5.

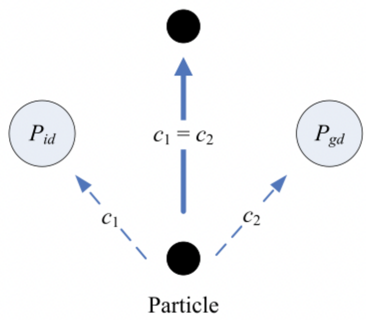

Fig 5. Particle movement behavior ($C_1 = C_2$)

In contrast, in the five PSO modifications discussed, where the value of C_2 is greater than C_1, the new position of a particle will be closer to the global best position ($p_{gb}^{(t)}$), as also illustrated in Figure 6. With such modifications, particle convergence is accelerated.

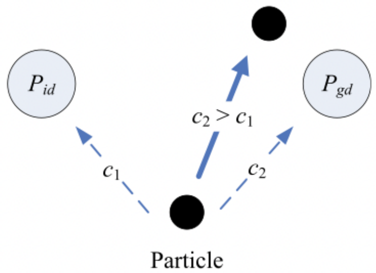

Fig 6. Particle movement behavior ($C_1 > C_2$)

Furthermore, changing the value from 2.5 to 0.5 for C_1 and from 0.5 to 2.5 for C_2 throughout the entire search range also results in a better optimal solution. At the start, a large cognitive component and small social component allow particles to explore the search space. In contrast, a smaller cognitive component combined with a larger social component helps the particles converge to the global optimum towards the end of the optimization.

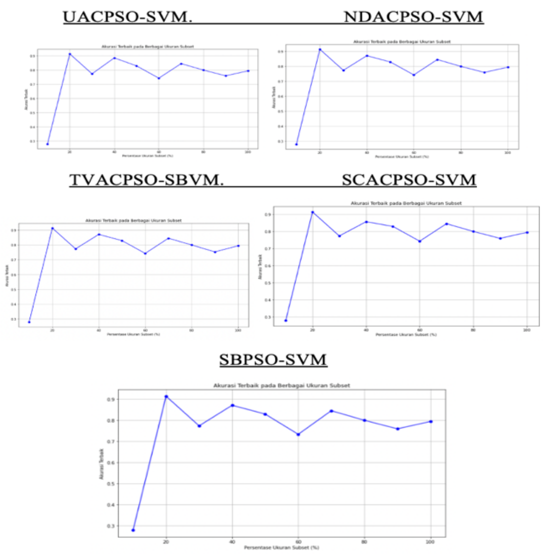

Fig 7. Best accuracy over various subset sizes

Next, every machine learning model that has been designed must meet the criterion that the model does not contain disturbances in the data. This criterion is known as the concept of "robustness." Robustness refers to a model's ability to continue functioning well or providing good results even in the presence of uncertainty, disturbances, or variations in input data or the environment.

In this study, robustness will be examined concerning data variations. A machine learning model will be evaluated on its ability to provide good and consistent results even when the input data experiences significant variations or changes in data distribution. Data variations can be caused by various factors, such as changing trends, environmental changes, or changes in data characteristics. A good model is one that is robust against data variations, capable of producing reliable predictions or results in various situations.

Based on the graphs on Fig 7, all variations of the PSO-SVM algorithms (UACPSO-SVM, NDACPSO-SVM, TVACPSO-SVM, SCACPSO-SVM, and SBPSO-SVM) show a similar pattern regarding resilience to changes in the subset dataset size. In smaller subsets, particularly at 20%, there is a significant spike in accuracy. However, after that, accuracy tends to stabilize despite slight fluctuations at various subset sizes, especially above 60%. Generally, each algorithm is able to maintain relatively stable performance as the subset size increases, indicating that the robustness of each algorithm is quite good. Although there are some variations in terms of fluctuations, there is no significant decrease in accuracy, indicating that all algorithms can adapt well to larger datasets. The findings suggest that while UAPSO-SVM excels in producing stable results, it may require more iterations to reach the optimal solution. In contrast, SBPSO-SVM offers rapid convergence, making it ideal for applications with stringent time constraints but potentially less suitable for highly dynamic environments.

CONCLUSION

This study has provided a comprehensive evaluation of various PSO-SVM modifications in the context of rainfall prediction, highlighting the strengths and weaknesses of each approach. The UAPSO-SVM algorithm demonstrated superior early exploration and convergence, delivering more optimal and homogeneous solutions compared to other algorithms. Conversely, while SPSO-SVM exhibited the fastest convergence speed, other algorithms like TVACPSO-SVM and NDACPSO-SVM required more iterations to reach stability. Additionally, UAPSO-SVM and TVACPSO-SVM showed lower hyperparameter variability, indicating more consistent results across different runs.

All algorithms demonstrated robust performance, maintaining stable accuracy despite significant changes in dataset size. Based on these findings, UAPSO-SVM stands out as the most effective approach for achieving optimal global solutions, whereas SPSO-SVM excels in convergence speed.

Future research could focus on further enhancing the scalability of UAPSO-SVM for larger datasets and exploring hybrid PSO-SVM models to leverage the strengths of multiple variants. It would also be valuable to incorporate alternative performance metrics, such as runtime efficiency, to fully assess the trade-offs between these algorithms. Finally, real-world validation in diverse domains would help to assess the generalizability and practicality of these findings in various application contexts.

REFERENCES

[1] Bimantara, I. M. S., & Widiartha, I. M. (2023). Optimization of K-Means Clustering Using Particle Swarm Optimization Algorithm for Grouping Traveler Reviews Data on Tripadvisor Sites. *Jurnal Ilmiah Kursor*, *12*(1), 1-10.

[2] Ramírez-Ochoa, D. D., Pérez-Domínguez, L. A., Martínez-Gómez, E. A., & Luviano-Cruz, D. (2022). PSO, a swarm intelligence-based evolutionary algorithm as a decision-making strategy: A review. Symmetry, 14(3), 455.

[3] Nur, A. Z., Suyono, H., & Aswin, M. (2020). APPLICATION OF HYBRID GA-PSO TO IMPROVE THE PERFORMANCE OF DECISION TREE C5. 0. *Jurnal Ilmiah Kursor*, *10*(4).

[4] Abbaszadeh, M., Soltani-Mohammadi, S., & Ahmed, A. N. (2022). Optimization of support vector machine parameters in modeling of Iju deposit mineralization and alteration zones using particle swarm optimization algorithm and grid search method. Computers & Geosciences, 165, 105140.

[5] Deligkaris, K. (2024). Particle Swarm Optimization and Random Search for Convolutional Neural Architecture Search. IEEE Access.

[6] Siivola, E., Paleyes, A., González, J., & Vehtari, A. (2021). Good practices for Bayesian optimization of high dimensional structured spaces. Applied AI Letters, 2(2), e24.

[7] Van den Bergh, F., & Engelbrecht, A. P. (2006). A study of particle swarm optimization particle trajectories. Information sciences, 176(8), 937-971.

[8] Freitas, D., Lopes, L. G., & Morgado-Dias, F. (2020). Particle swarm optimisation: a historical review up to the current developments. Entropy, 22(3), 362.

[9] Houssein, E. H., Gad, A. G., Hussain, K., & Suganthan, P. N. (2021). Major

advances in particle swarm optimization: theory, analysis, and application. Swarm and Evolutionary Computation, 63, 100868.

[10] Tian, D., Zhao, X., & Shi, Z. (2019). Chaotic particle swarm optimization with sigmoid-based acceleration coefficients for numerical function optimization. Swarm and Evolutionary Computation, 51, 100573.

[11] Wu, L., Zhao, D., Zhao, X., & Qin, Y. (2023). Nonlinear Adaptive Back-Stepping Optimization Control of the Hydraulic Active Suspension Actuator. Processes, 11(7), 2020.

[12] Gad, A. G. (2022). Particle swarm optimization algorithm and its applications: a systematic review. Archives of computational methods in engineering, 29(5), 2531-2561.

[13] Tian, D., Zhao, X., & Shi, Z. (2019). Chaotic particle swarm optimization with sigmoid-based acceleration coefficients for numerical function optimization. Swarm and Evolutionary Computation, 51, 100573.

[14] Tsai, C. Y., & Kao, I. W. (2011). Particle swarm optimization with selective particle regeneration for data clustering. Expert Systems with Applications, 38(6), 6565-6576.

[15] Kardani, N., Bardhan, A., Samui, P., Nazem, M., Asteris, P. G., & Zhou, A. (2022). Predicting the thermal conductivity of soils using integrated approach of ANN and PSO with adaptive and time-varying acceleration coefficients. International Journal of Thermal Sciences, 173, 107427.

[16] Ratnaweera, A., Halgamuge, S. K., & Watson, H. C. (2004). Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. IEEE Transactions on evolutionary computation, 8(3), 240-255.

[17] K. Chen, F. Zhou, L. Yin, et al., A hybrid particle swarm optimizer with sine cosine acceleration coefficients, Inf. Sci. 422 (2018) 218–241.

[18] Fernando, W., Jollyta, D., Priyanto, D., & Oktarina, D. (2024). THE INFLUENCE OF DATA CATEGORIZATION AND ATTRIBUTE INSTANCES REDUCTION USING THE GINI INDEX ON THE ACCURACY OF THE CLASSIFICATION ALGORITHM MODEL. Jurnal Ilmiah Kursor, 12(3), 111-122.

[19] Kalita, D. J., & Singh, S. (2020). SVM hyper-parameters optimization using quantized multi-PSO in dynamic environment. Soft Computing, 24(2), 1225-1241.

[20] Candelieri, A., Giordani, I., Archetti, F., Barkalov, K., Meyerov, I., Polovinkin, A., ... & Zolotykh, N. 2019. Tuning hyperparameters of a SVM-based water

demand forecasting system through parallel global optimization. Computers & Operations Research, 106, 202-209. https://doi.org/10.1016/j.cor.2018.01.013.

[21] Schratz, P., Muenchow, J., Iturritxa, E., Richter, J., & Brenning, A. (2019). Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data. Ecological Modelling, 406, 109-120.

[22] Nalepa, J., & Kawulok, M. (2019). Selecting training sets for support vector machines: a review. Artificial Intelligence Review, 52(2), 857-900.

[23] Sajedi, S. O., & Liang, X. (2020). A data-driven framework for near real-time and robust damage diagnosis of building structures. Structural Control and Health Monitoring, 27(3), e2488.

[24] Shih, H. C., Stow, D. A., & Tsai, Y. H. (2019). Guidance on and comparison of machine learning classifiers for Landsat-based land cover and land use mapping. International Journal of Remote Sensing, 40(4), 1248-1274.

[25] Bouchene, M. M., & Boukharouba, A. (2022). Features extraction and reduction techniques with optimized SVM for Persian/Arabic handwritten digits recognition. Iran Journal of Computer Science, 5(3), 247-265.

[26] Irianto, H. A. (2021). Statistik untuk Ilmu Sosial: Aplikatif untuk Ilmu-ilmu Sosial. Prenada Media.

.