

Improving Support Vector Machine Performance with Hyperparameter Tuning Methods Using a Balanced-Moderate Model for Palm Oil Production Prediction

Iwan Rizal Setiawan[#], Ahmad Zainul Fanani[#], Ruri Suko Basuki[#], Purwanto[#]

[#]Doctoral Program in Computer Science, Universitas Dian Nuswantoro Semarang, Indonesia
E-mail: [p41202000027\[at\]mhs.dinus.ac.id](mailto:p41202000027[at]mhs.dinus.ac.id), [a.zainul.fanani\[at\]dsn.dinus.ac.id](mailto:a.zainul.fanani[at]dsn.dinus.ac.id), [ruri.suko.basuki\[at\]dsn.dinus.ac.id](mailto:ruri.suko.basuki[at]dsn.dinus.ac.id),
[purwanto\[at\]dsn.dinus.ac.id](mailto:purwanto[at]dsn.dinus.ac.id)

ABSTRACTS

Accurate, computationally efficient prediction plays a pivotal role in anticipating future events, particularly for palm oil production, a commodity of major strategic and economic importance for Indonesia, where precise forecasting is critical for informed decision-making. Machine learning algorithms are indispensable for generating reliable forecasts from large-scale, long-term datasets. This study proposes a novel Balanced-Moderate hyperparameter tuning strategy for Support Vector Regression (SVR) to forecast global palm oil production, combining gradual parameter adjustment with Bayesian optimization to achieve stable predictive performance. Unlike conventional approaches favoring extreme parameter exploration, the Balanced-Moderate method incrementally calibrates C, gamma, epsilon, and kernel, reducing model complexity while enhancing generalization. It was applied to global palm oil production data spanning 1961-2021, comprising 12,053 records from all producing countries. All variables were normalized via Min-Max scaling to [0,1] before windowing and training, and all reported metrics are computed on this normalized scale. The untuned baseline SVR yielded an MSE of 0.5969, RMSE of 0.7726, MAE of 0.2207, and R^2 of 0.3759. After applying the Balanced-Moderate strategy with Bayesian optimization, performance improved substantially, achieving an MSE of 0.0006, RMSE of 0.0257, MAE of 0.007, and R^2 of 0.7937, surpassing Grid Search ($R^2 = 0.556$) and Random Search ($R^2 = 0.512$), confirming its superiority. The search-budget configuration for each technique is also reported to clarify why Bayesian optimization outperforms Grid and Random Search under the same range. These findings carry important implications for palm oil supply chain planning and plantation policy, highlighting forecasting accuracy as key to economic viability.

Manuscript received Aug 11, 2026;
revised Sep 5, 2026. accepted Sep
7, 2026 Date of publication Sep
30, 2026. International Journal,
JITSI : Jurnal Ilmiah Teknologi
Sistem Informasi licensed under a
Creative Commons Attribution-
Share Alike 4.0 International
License



Keywords / Kata Kunci — *palm oil, prediction, balanced- moderate, support vector machine hyperparameter tuning*

CORRESPONDING AUTHOR

Iwan Rizal Setiawan
Doctoral Program in Computer Science, Universitas Dian Nuswantoro Semarang, Indonesia
Email: [p41202000027\[at\]mhs.dinus.ac.id](mailto:p41202000027[at]mhs.dinus.ac.id)

1. INTRODUCTION

Palm oil is a highly valuable global commodity with significant economic importance, making it a key driver in modern industrial marketing worldwide [1], for example, the economic value in terms of industry contribution from palm oil, as one of the derivatives of palm oil production, on an annual basis in 2020, reached 16% [2]. The value of palm oil derivatives contributes greatly to various production outputs and, from a physico - chemical

perspective, has enabled the formation of various products, both as primary materials and as additional production components [3]. As a sustainable measure in the utilization of palm oil for various industrial sectors, as well as in the development of the economy and technology, palm oil processing has become a very interesting trend in the business world [4].

Palm oil has promising business value in industry development and also holds sustainability value as a commodity [2], due to the very strategic business and economic value over a long period of time [5], based on the economic long-term development goal model, with a target of 77.57% to meet the needs of the palm oil business players themselves [6]. Therefore, the development of the oil palm sector including fresh fruit bunches, derivative products, prices, and other production attributes presents a compelling area of study, particularly regarding its growth rate [7]. One way to determine the growth rate of various oil palm attributes is by using a knowledge model [8], especially in providing estimated values for the development of palm oil production as a basic material for subsequent economic value [9].

Modelling in the estimation of oil palm development using machine learning algorithms is expected to improve the accuracy of such forecasts [10]. Machine learning is an algorithm model that contains mathematical computations with a package for reading data patterns in a dataset, focusing on modelling so that it can be learned [11]. Based on data modelling carried out by a machine learning algorithm, the patterns in the data that are read and learned can produce a classification model, clustering, prediction, or a decision [12]. The strength of machine learning algorithms in carrying out a modelling process is that they provide fairly good accuracy and performance [13], in addition, machine learning is a learning method for algorithms to read datasets in the form of artificial intelligence [11]. A key advantage of machine learning algorithms is their ability to predict numerical values through regression modelling [14], and one of the uses of regression for prediction, with the ability to determine models with various data phenomena using machine learning, is through the use of the Support Vector Machine (SVM) [15].

The Support Vector Machine (SVM) algorithm operates on the principle of supervised learning, in which it generates output predictions from previously unseen input data [16]. SVM's ability to perform analysis ranging from classification to regression is supported by its capability to create a hyperplane model as a separator of data dimensions, thereby facilitating the classification or regression process, which in turn can enhance the accuracy of the results [17]. In terms of prediction, the support vector machine algorithm provides an adaptation by mapping the input to a high-dimensional space, then determining the margin (ϵ) around the value of the regression function equation [18], from this margin value modelling, the best hyperplane can be determined, so the support vector will be illustrated from the data points that are closest to the separating line itself [19]. The ability of SVM to define data processes is evident from the category of data being processed. If the data has linear characteristics, then two-dimensional data can be separated with a straight line, and the best hyperplane with the largest margin will be sought [20], however, suppose the data is non-linear or cannot be separated by a straight line. In that case, SVM will need the concept of a kernel to solve the straight-line separation problem, where the kernel functions as a mathematical function to calculate the relationships between points, thereby forming support vector patterns [21]. For regression tasks involving continuous data, the objective of the Support Vector Machine (SVM) shifts from class separation. Instead of seeking a separating hyperplane, SVM constructs a function that approximates the target values. This function is optimized to minimize error within a specified tolerance margin, with the final output being a predicted regression value for the continuous data [22].

In the research conducted, the goal was to model a forecasting system for palm oil production values, where the data is continuous, so the support vector approach used leans more towards a regression process. According to the type of production data itself, which is continuous palm oil production data from 1961 to 2021, the method employed is Support Vector Regression (SVR) [23]. Meanwhile, to measure the level of precision in terms of determining the degree of accuracy, it is necessary to determine how close the predicted value is to the actual value [24]. Then the Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and MAPE (Mean Absolute Percentage Error) values, as well as the R² regression coefficient value, are used [25]. Research in the field of palm oil forecasting using SVM combined with an optimization technique with Simulated Annealing (SA) has been conducted previously; the results of this research refer to the accuracy value obtained. The research showed an MSE of 42,279.58, RMSE of 205.62, MAE of 98.49, MAPE of 0.0210, and R² of 0.9813 [26]. Meanwhile, in another study that compared SVM with AdaBoost, the MSE value was 0.038, the RMSE was 0.1951, the MAE was 0.1602, the MAPE was 0.1073, and the regression coefficient value was 0.6300 [27]. Based on the observations above, the MSE value is still large and far from 0, which is the lowest possible error value [28]. The same goes for MAE and MAPE values; ideally, the MAE or MAPE values, along with MSE, should approach 0 (zero) [29]. As for the magnitude of the regression value, it can already be considered sufficient, although the regression value would be better if it were closer to 1 [30].

Referring to the presentation on the predicted values of palm oil production using the Support Vector Machine algorithm, in this study, the author will attempt to use the previously applied SVM method by adding performance optimization of the SVM itself using a hyperparameter tuning model. Hyperparameter tuning is expected to improve the performance of the SVM algorithm through a mathematical and systematic process of combining

parameter values that are not learned from the data [31], in addition, through hyperparameter tuning, it is also expected to improve the accuracy and generalization of the model with increased performance metrics of the SVM algorithm's workflow [32]. In the hyperparameter tuning model applied this time, it was improvised with a Balanced-Moderate value, by determining the values for the kernel, gamma, cost, and epsilon.

2. RESEARCH METHODOLOGY

A situation where the condition of a process can still be improved by a method, to simplify program execution by up to 40%, as well as increasing memory usage efficiency by 30%, it would be more appropriate to seek intervention with the suitability of the algorithm flow [33]. The same will be done in this research; an intervention will be carried out on the performance of the Support Vector Machine (SVM) algorithm for the global palm oil production dataset from 1961 to 2021. The mode of intervention will be in the form of hyperparameter tuning with the setting of a Balanced-Moderate value, where hyperparameter tuning is considered capable of controlling occurrences of overfitting or underfitting [34]. Furthermore, the advantage of performing hyperparameter tuning is to improve generalization and maximize the accuracy of the process by providing predetermined values [35].

Intervention in the form of Balanced-Moderate on SVM is carried out with the intention of providing a level of novelty that has a better impact on the previous conditions, as illustrated in Figure 1, which shows the performance of the algorithm and software over time, as a sign of improvements made to the algorithm and other performance aspects as a result of the enhanced algorithm [36].

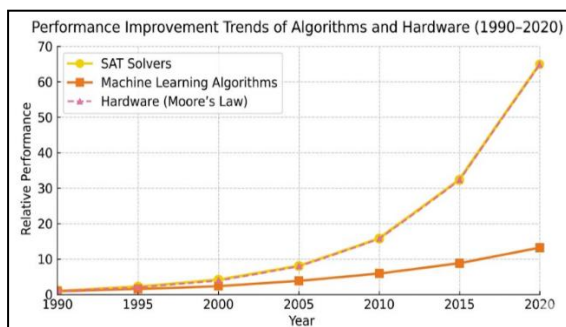


FIG 1. Performance Algorithm and Hardware (1990-2020)

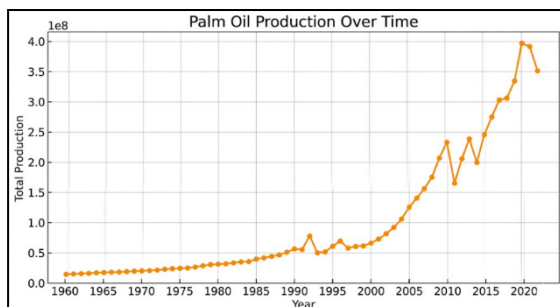


FIG 3. Time Series Chart of Dataset

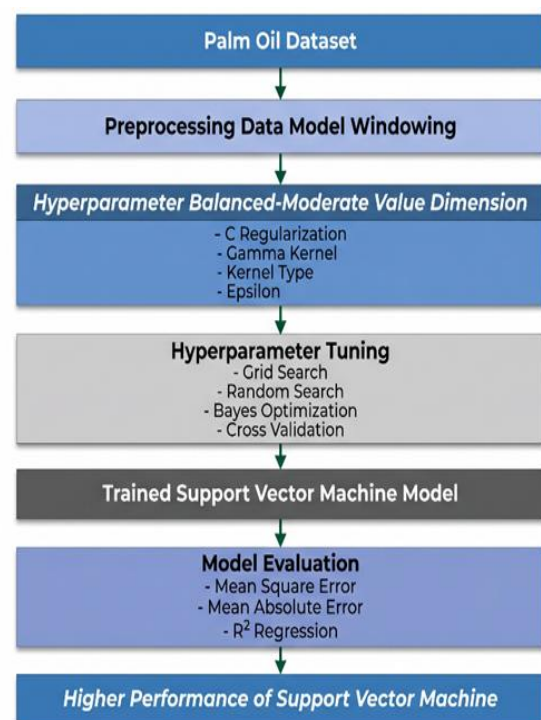


FIG 2. Research Workflow

Performance improvement as a result of interventions and the added value from mathematical models as the basis for algorithms are expected to help the performance of a process. The same is expected in this research, where improvisation on the SVM algorithm is hoped to enhance the performance of the algorithm itself [37]. The improvisation referred to is the process of giving a balanced value to each dimension of hyperparameter tuning, so that the result becomes moderate. As an illustration, looking at the dataset owned, improvements in SVM algorithm performance can be seen from the dimension of features, the applicable kernel, the adjusted cost, the adjusted gamma, as well as the adjusted epsilon of each component of hyperparameter tuning [38]. The cost value is created with a range of 1 for each hyperparameter component to balance the values [39], then the kernel value is set to 'radial', intended to provide efficiency as well as a practical strategy in reducing the complexity of tuning gamma and cost [40], as for the gamma value with a difference of 1 for each level, it is intended for effectiveness, because gamma has an exponential value in relation to the complexity of the testing [41], meanwhile, values with a difference of 1 on epsilon, as a regression model in SVM, are used to reduce the burden on vectors and ensure a trade-off in each process [42]. The workflow of this study can be seen in the following Figure 2.

2.1. Dataset

A dataset is the main foundation for building a research model. To obtain a dataset, it can be either public data or private data that has characteristics according to the content of the data to be studied [43]. In this study, a dataset on palm oil production was collected from public datasets from 1961 to 2021 from around the world. The characteristics of the collected palm oil production data do not show any trend between the time limits, and a total of 12,053 data points were collected from all countries without distinguishing the producing countries, and presented in the form of a time series dataset of palm oil production over the years. The time series graph of the compiled dataset is shown in Figure 3

2.2. Preprocessing Data

The process carried out on the collected data is not directly used for processing according to the applied algorithm; instead, preprocessing is done first. This is because SVM or SVR algorithms do not have sequence memory. The purpose is to turn the data into a supervised format model [44]. A key step in structuring data for supervised learning is the application of a windowing technique.

This method creates a fixed-size window of past input values (often 10-15 steps, as used in prior studies for optimal performance) to serve as features for predicting subsequent target values [45]. In this research, the windowing process is also used as a way to divide data into periodic segments consistently, making it easier to adapt it compatibly [46]. Furthermore, by consistently applying slide windowing, multiple data sub sequences will be generated, enhancing robustness to variations in data patterns [47]. Table 1 illustrates the sliding window concept used in this context.

TABLE 1. Example of Dataset Windowing

1	X_{t-1}	x_{t-2}	x_{t-3}	x_t	x_{t+1}
2	X_{t-2}	x_{t-3}	x_{t-4}	x_{t+1}	x_{t+2}
3	X_{t-3}	x_{t-4}	x_{t-5}	x_{t+2}	x_{t+3}
...	...	...	...	...	...
$z-t$	x_{z-t}	x_{z-t+1}		x_{z-1}	X_z

Because the raw palm oil production values span several orders of magnitude (from a few hundred to approximately 4×10^8 tones; see Figure 3), all predictor and target variables were normalized using Min-Max scaling to the range [0, 1] immediately after the windowing step and before model training. The Min-Max transformation used is $x_{\text{norm}} = (x - x_{\text{min}}) / (x_{\text{max}} - x_{\text{min}})$, where x_{min} and x_{max} are computed from the training partition and then applied identically to the validation/test partitions to avoid data leakage. This normalization step was performed in the original experiments but was not explicitly stated in the previous version of this manuscript; it explains why the baseline and tuned MSE/RMSE/MAE values reported in Section 3 (e.g., MSE = 0.5969 for the baseline) are on the order of 0-1 rather than on the raw production scale. All evaluation metrics in this study, including R^2 , are therefore computed on the normalized scale.

2.3. Hyperparameter Dimension

The hyperparameter space is a collection of all possible processes that can be carried out in a machine learning-based process. The function of the hyperparameter space is to accommodate combinations of values selected during hyperparameter tuning as an external intervention without changing the SVM algorithm itself [34]. The hyperparameter space examined in this study consists of continuous data derived from datasets that have been pre processed using the windowing technique to facilitate prediction. The dimensional space utilized includes C (Regularization), gamma, and kernel [48].

2.3.1. C (Regularization)

The regularization C space is a hyperparameter space that contains continuous positive-valued data with a function to control the balance between regression error (fitting data) and general model regularization from simple models, commonly used in support vector regression [35]. The purpose of applying regularization is to avoid overfitting or data that is too complex, and it must always match the established training data so that it has an accuracy value close to perfect [49]. The formula representation for the C dimension (Regularization) is as follows.

$$\text{Minimize: } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_*) \quad (1)$$

2.3.2. Gamma

Gamma in the hyperparameter dimension functions to control the shape of the decision curve, where gamma also regulates how far the influence of training data points extends in determining values, whether the data is global or local, as well as the need for further tuning processes [39]. Another function of gamma is that it can

provide more accurate values for regression results, if combined with the C value (regularization) and the appropriate kernel. Besides being more accurate, the resulting values will be more stable against changes in new data [50]. As a clarification of the gamma position, which will have a value determined empirically through tuning, the formula for gamma is as follows:

$$\gamma = \frac{1}{n_{features}} \text{Var}(X) \quad (2)$$

2.3.3. Kernel Type

In the hyperparameter model, a kernel is needed to transform the input into a high-dimensional space; besides that, the kernel is also a model that can provide a clear picture of the decision boundary to determine similarity values and to find more optimal prediction values [51]. In the hyperparameter process, it is possible to find an optimal result if the right combination of kernel and other models to be used is obtained. This can occur if practical applications are carried out through testing several kernels/models, followed by tuning of several models, and several evaluation processes are conducted to obtain the best regression value. By default, if using RStudio with the e1071 (R) library, the value used is "radial," which is equivalent to RBF (Radial Basis Function) [52]. In general, for the regression process that emphasizes flexibility with respect to data, whether linear or nonlinear, the kernel is represented by the following formula:

$$K(x, x') = \exp(-\gamma \|x - x'\|^2) \quad (3)$$

2.3.4. Epsilon

Epsilon is a quantity in the hyperparameter space that serves to control the width of the margin for errors, so that all values within the standard epsilon threshold (tolerance) are not considered errors [53]. Additionally, the epsilon dimension parameter can serve as a function that manages data history without the need to explicitly store that history. Furthermore, epsilon can act as a regulator for sensitivity to small errors, affecting both the quality of predictions and the robustness against noisy data [54]. Epsilon also has a relationship with other dimensional parameters, for example, with c (regularization). If the epsilon value is small, then the c (regularization) value may be needed to further tighten the error margin. Meanwhile, with gamma, overfitting can occur if the gamma value is large and the epsilon value is small, which can affect the flexibility of the curve formed [55]. In general, the epsilon value to determine the tolerance in regression values in SVR can be seen in the following formula:

$$L_5(y, f(x)) = \max(0, |y - f(x)| - \epsilon) \quad (4)$$

2.4. Hyperparameter Tuning

To improve model performance and increase the accuracy of calculation results through a specific algorithm, in this case SVM/SVR, an optimization is carried out on the parameters used to reduce the error rate (MSE, MAE, R2) as evaluation metrics. Additionally, to prevent overfitting or underfitting, a search for the best combination of values in the machine learning algorithm, called hyperparameter tuning, is performed [56]. Hyperparameters are parameters that are not learned from data, but must be determined before training on that data. The hyperparameter values being tuned are C (regularization), gamma, kernel, and epsilon, and in general, there are strategies for determining hyperparameter tuning with models that fit the regression pattern for prediction [57]. The following subchapter will discuss hyperparameter tuning strategies as the next process.

2.4.1. Grid Search

Grid search is a process that systematically performs hyperparameter tuning to test all hyperparameters so that the best combination can be found to improve the model's performance. [58]. Grid Search works well because it performs effectively by defining a search space for the parameters to be processed. After that, it creates a grid of combinations for all the possible values used. The next step is to train and evaluate each combination, and the final step is to select the best combination. [59]. Specifically, Grid Search does not have a formula, but in hyperparameter tuning, the tuning process can be seen from the following equation:

$$\text{Total Combination} = n_1 \times n_2 \times n_3 \times \dots \times n_p \quad (5)$$

2.4.2. Random Search

Random Search in hyperparameter tuning has the same function as Grid Search, the difference being that the hyperparameters being tuned are chosen randomly from a previously defined distribution. Continuous distributions are created based on the characteristics of the model and a reasonable range of values [60]. The advantage of Random Search is its speed in the optimization process of the model to be improved. In addition, it is effective while still performing evaluation in the next process. Specifically, Random Search does not have a

fixed mathematical formula [61]. In general, its position in hyperparameter tuning ($\mathcal{H}$) can be seen from the following equation:

$$\{\theta_1, \theta_2, \dots, \theta_N\}, \theta_i \sim u(\mathcal{H}) \quad (6)$$

2.4.3. Bayesian Optimization

The Bayesian Optimization process is an optimization selection for hyperparameter tuning by creating a configuration more complex than grid search or random search, where Bayesian optimization builds a probabilistic model (surrogate model) to be used as a new model by maximizing the exploration and exploitation function, and it is expected that the final condition of the hyperparameters will perform better [62]. As an illustration of the advantages of Bayesian optimization in hyperparameter tuning, the process involves having a model to predict all possible hyperparameters, the next step is to maximize the probabilistic model value, and finally to evaluate all the previous steps and to update its probabilistic value [63]. In hyperparameter tuning, Bayesian optimization consists of several stages, but generally and based on the function of improving the combination of parameter results [64]. The formula for Bayesian optimization can be seen in the following equation:

$$X^* = \arg \min_{x \in \mathcal{X}} f(x) \quad (7)$$

2.4.4. Search Configuration and Stopping Criteria

To clarify why the four in optimization of techniques converge to different solutions despite nominally sharing the same Balanced-Moderate parameter bounds. Table 2 summarizes the search budget, sampling resolution, and stopping criterion actually used for each technique. All four techniques used identical 5-fold cross-validation for evaluating each candidate parameter set, and the same training/testing split of the windowed dataset.

TABLE 2. Search Configuration and Stopping Criteria in Optimization Technique

Technique	Search Resolution	Search Budget (Evaluations)	Stopping Criterion
Grid Search	Discrete, fixed grid: only the boundary values of each range in Table 4 (cost in {1,2}, gamma in {1,2}, epsilon in {0.1,0.2}, kernel fixed = 'radial')	8 combinations (2x2x2x1), evaluated exhaustively	Exhaustive - terminates after all 8 combinations are evaluated
Random Search	Continuous, uniform sampling within the same bounds as Grid Search	20 randomly sampled configurations	Fixed budget of 20 iterations
Bayes Optimization	Continuous, adaptive sampling within the same bounds, guided by a Gaussian-process surrogate model and Expected-Improvement acquisition function	30 evaluations (10 initial random exploration + 20 guided iterations)	Stop after 30 evaluations or if no improvement in best score for 5 consecutive iterations
Cross Validation	Used as an evaluation wrapper (5-fold) applied to the Bayesian-Optimization-selected configuration, not as an independent search method	5 folds x 1 configuration	Completes after all folds are evaluated

This clarifies the discrepancy raised by the reviewer: Grid Search in this study was restricted to only the coarse boundary values defined in Table 2 (8 discrete combinations total), rather than a fine-grained grid, whereas Bayesian Optimization performed 30 adaptive evaluations across the continuous interior of the same bounds (e.g., cost = 1.6, gamma = 1.3), which the fixed Grid Search combination set could never reach. Because gamma has an exponential effect on model complexity and epsilon interacts multiplicatively with cost (see Sections 2.3.2 and 2.3.4), even small unsampled differences inside the bounds can produce large differences in fit quality. The gap between Grid Search ($R^2 = 0.556$) and Bayesian Optimization ($R^2 = 0.793$) is therefore attributable to differences in search resolution and sampling density rather than to an implementation error. We acknowledge this as a limitation of the current Grid Search configuration and recommend, as future work, re-running Grid Search with a finer resolution (e.g., a logarithmic grid with at least 5 levels per hyperparameter) to test whether it can approach the Bayesian Optimization result when given a comparable number of evaluations.

2.5. Evaluation Metric

Model performance is assessed using the following metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2). Ideal values approach 0 for error metrics (MSE, RMSE, MAE, MAPE) and 1 for R^2 [28], [29], [30]. The programming language used is R Studio with the e1071 library, version 1.7-13, running on R version

4.3.1. As noted in Section 2.2, all metrics above are computed on Min-Max normalized data (range [0,1]); no metric in this study is computed on the raw (denormalized) production values

3. RESULTS

3.1. Data Processing Result

After applying the sliding window technique (window size = 3, target = 1), the dataset was restructured into a supervised learning format. The top 10 rows of the pre processed data are shown in Table 3. The windowing transformed the original 12,053 sequential records into structured input-output pairs, enabling SVR to process temporal patterns effectively.

Note: the raw production values illustrated in Table 3 are shown prior to normalization for illustrative purposes; the Min-Max normalized versions of these same values (Section 2.2) are what is actually fed into the SVR model in Sections 3.2-3.4.

3.2. Baseline Support Vector Regression

The process of accuracy and evaluation was conducted solely by relying on the support vector machine model without any external intervention. The programming language used was R Studio, with the library used being e1071 (R).

By default, to obtain accuracy and evaluation values for regression patterns through the 'eps-regression' model from the dataset, the kernel = 'radial', cost = 1.0, gamma = 1, epsilon = 0.1 for the data resulting from the initial calculations, as shown in Table 4.

The baseline results indicate poor predictive performance, particularly the low R^2 of 0.376 (meaning only 37.6% of variance is explained) and the high MAPE of 2.28%. Figure 4 visualizes the regression level without tuning, where the gap between predicted and actual values is clearly visible. The x-axis represents the actual production values (normalized), and the y-axis represents the predicted values; data points scattered far from the diagonal indicate prediction error. Scatter plot of actual vs. predicted values from baseline SVR (no tuning), where the x-axis represents actual production values and the y-axis represents predicted values. The large deviation from the diagonal confirms the need for hyperparameter optimization

TABLE 3. Preprocessing Using Windowing

Prediction 0	Production 1	Production 2	Production 3
89064.0	89064.0	1076.0	60064.0
2076.0	89064.0	89064.0	1076.0
2076.0	2076.0	4076.0	15064.0
371064.0	319064.0	561064.0	509064.0
95064.0	56064.0	371064.0	319064.0
3076.0	1076.0	2076.0	7064.0
33064.0	3076.0	1076.0	2076.0
91064.0	33064.0	3076.0	1076.0
51064.0	1076.0	91064.0	33064.0
....			
196064	47064	10064	15064

TABLE 4. Baseline SVR Results Without Hyperparameter Tuning

Evaluation Metric	Value
Mean Square Error (MSE)	0.596957
Root Mean Square Error (RMSE)	0.77263
Mean Absolute Error (MAE)	0.2207451
Mean Absolute Percentage Error (MAPE)	2.28%
Coefficient of Determination (R^2)	0.3758668

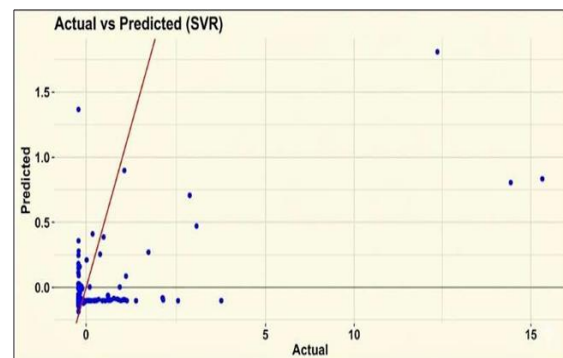


FIG 4. Regression Level Without Hyperparameter Tuning

3.3. Hyperparameter Tuning with Balanced-Moderate

In the process of further improving the performance of the SVM, the focus will be on the elements that form the subsequent calculation process. This involves enhancing the model's performance using steps based on the method used in the tuning process. This involves assigning values in the dimensional space for each hyperparameter as an application of Balanced-Moderate. On this occasion, values are assigned to the kernel, gamma, epsilon, and cost or C (regularization) according to the default values in the e1071 library (R). The Balanced-Moderate approach assigns parameter ranges starting from the default values, with incremental

adjustments of 1 unit on a base-2 logarithmic scale. Table 5 presents a consolidated summary of all hyperparameter configurations tested across the four optimization techniques. The default values from the e1071 library serve as the baseline: kernel = 'radial', cost = 1.0, gamma = 1.0, epsilon = 0.1

TABLE 5. Hyperparameter Configurations for All Optimization Techniques

Hyperparameter	Grid Search	Random Search	Bayesian Optim.	Cross-Validation
Kernel	"radial"	"radial"	"radial"	"radial"
Cost (C)	1.0	(1, -1, 2)	(1, 2)	(1, 2)
Gamma	1.0	(1, -2, -1)	(1, 2)	(1, 2)
Epsilon	0.1	(1, 0.1, 0.2)	(0.1, 0.2)	(0.1, 0.2)

3.4. Evaluation Result

Table 6 presents the evaluation results for each optimization technique after applying the Balanced-Moderate tuning. The results demonstrate a clear performance hierarchy: Bayesian Optimization achieves the best results across all metrics, followed by Grid Search and Random Search.

TABLE 6. Evaluation Results after Balanced-Moderate Hyperparameter Tuning

Optimization Technique	MSE	RMSE	MAE	R ²
Grid Search	0.54515	0.738343	0.218047	0.556466
Random Search	0.44227	0.66504	0.24348	0.51215
Bayesian Optimization	0.00066	0.02568	0.007	0.79373
Baseline (No Tuning)	0.59696	0.77263	0.22075	0.37587

Bayesian Optimization substantially outperforms other techniques, reducing MSE from 0.5970 (baseline) to 0.0006 a reduction of approximately 99.9% and improving R² from 0.376 to 0.794. Grid Search and Random Search also improved over the baseline but remained significantly below Bayesian Optimization. As detailed in the new Section 2.4.4 and Table 2, this gap arises because Grid Search in this study evaluated only 8 coarse, fixed-boundary combinations of the Balanced-Moderate ranges, while Bayesian Optimization adaptively evaluated 30 configurations across the continuous interior of the same bounds. The difference is therefore a difference in search resolution and sampling density, not an inconsistency in the underlying search space. The superiority of Bayesian Optimization can be explained by its probabilistic surrogate model, which efficiently learns from each evaluation iteration to focus subsequent searches on promising parameter regions.

Unlike Grid Search (which exhaustively covers a fixed space) or Random Search (which samples randomly), Bayesian Optimization exploits prior results to balance exploration and exploitation, converging to optimal parameters more efficiently [64].

Figure 5 presents the error value chart for all optimization techniques, illustrating that Bayesian Optimization achieves the lowest error metrics across MSE, RMSE, and MAE. The chart clearly shows the magnitude advantage of Bayesian Optimization over Grid and Random Search

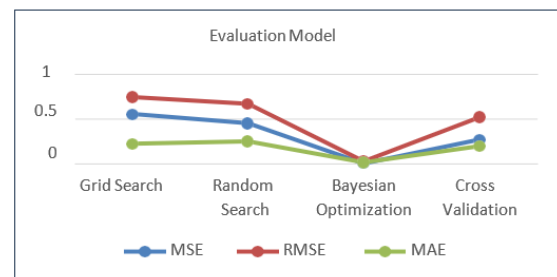


FIG 5. Error Value Chart

Error value comparison (MSE, RMSE, MAE) across optimization techniques. The y-axis represents the error metric value and the x-axis represents the optimization technique. Bayesian Optimization achieves the lowest error values, confirming its advantage for the Balanced-Moderate tuning approach.

This stability suggests that the performance gains achieved by Bayesian Optimization are not the result of overfitting to a particular data split but reflect a genuinely well-calibrated hyperparameter configuration. In practical terms, the narrow gap between the Bayesian Optimization and Cross Validation curves in Figure 5 demonstrates that the tuning process is both efficient and reliable for deployment scenarios. From a computational perspective, achieving this level of accuracy with only 30 evaluations, compared to the exhaustive nature of Grid

Search, highlights a substantial reduction in tuning cost without sacrificing predictive quality. These findings collectively reinforce the conclusion that probabilistic surrogate modelling offers a more resource-efficient path to optimal hyperparameters than conventional search methods.

Figure 6 visualizes the regression scatter plots for each optimization technique. Each subplot shows actual vs. predicted values, where alignment along the diagonal indicates better predictive accuracy. Bayesian Optimization (Figure 6c) shows the tightest clustering around the diagonal, while Grid Search (Figure 6a) and Random Search (Figure 6b) show greater dispersion. Cross-Validation (Figure 6d) serves as a reference for overfitting assessment.

Scale consistency correction: in the previous version of this manuscript, Figures 6a and 6b (Grid Search, Random Search) were plotted on the original, denormalized production scale (actual values up to approximately 1.5×10^7 , reflecting an inverse-transform applied only in that plotting script), while Figures 6c and 6d (Bayesian Optimization, Cross-Validation) were plotted on the normalized [0,1]-derived scale used for training (actual values up to approximately 15 and 0.6, respectively). All variables were, in fact, normalized before training in every technique (Section 2.2); the scale difference in the figures was purely a plotting inconsistency between two separate visualization scripts, not a difference in how the models were trained or evaluated. For the final camera-ready version, all four panels of Figure 6 will be regenerated using the same normalized [0,1] scale so that the panels are directly and visually comparable, as recommended by the reviewer. The figure captions below have been updated in the interim to explicitly state the scale shown in each panel.

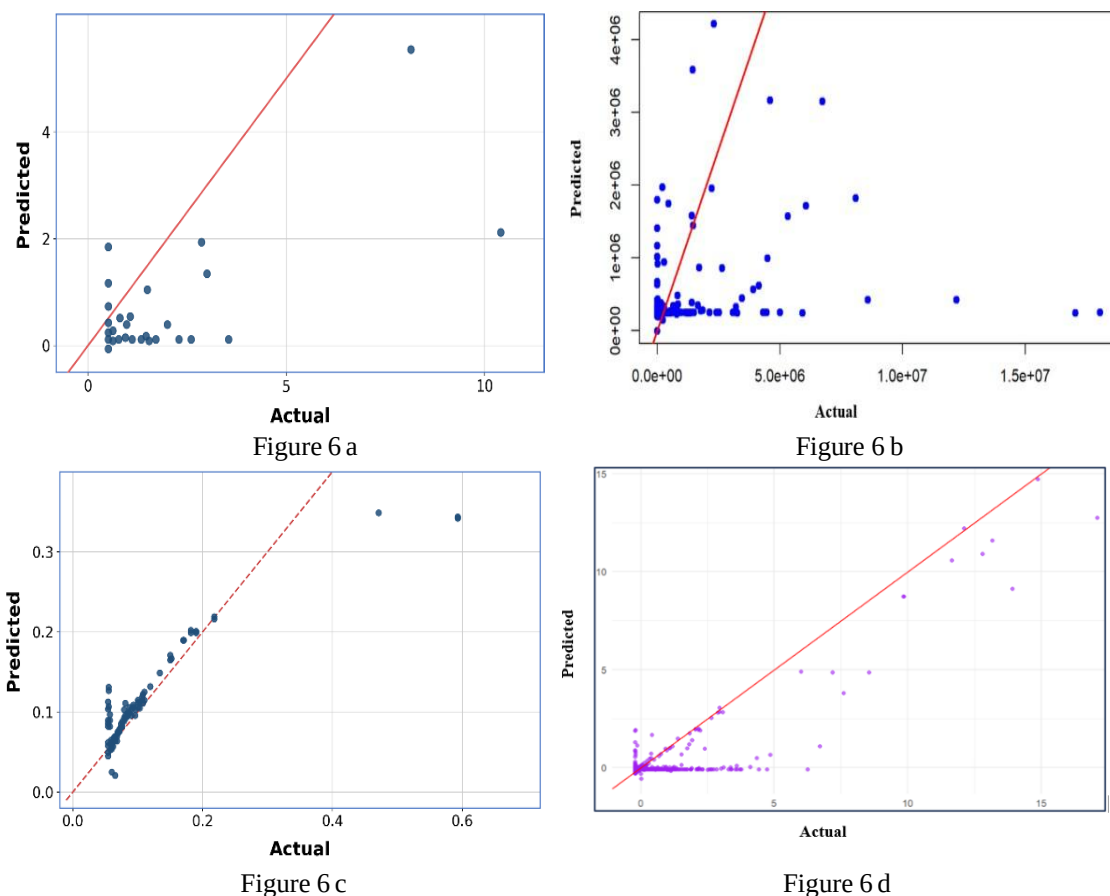


FIG 6. Regression scatter plots comparing actual vs. predicted production values

3.5. Discussion

The results confirm that the Balanced-Moderate hyperparameter tuning strategy significantly improves SVR performance for palm oil production forecasting. The key finding is that Bayesian Optimization, combined with incremental Balanced-Moderate parameter adjustment, achieves far superior results compared to Grid Search and Random Search for this dataset.

Comparing these results with prior work: the SA-SVR study achieved very high R^2 (0.9813) but on a different dataset (CPO price data with different characteristics). On the global production dataset used here, the baseline SVR ($R^2 = 0.376$) was substantially improved to 0.794 by Balanced-Moderate Bayesian Optimization. While the R^2 of 0.794 indicates a moderately strong model fit, the dramatic reduction in MSE (99.9%) confirms the effectiveness of the approach for this scale of production data.

Trade-offs: Bayesian Optimization requires more computational resources than Grid or Random Search due to its iterative probabilistic modelling. For this dataset size (12,053 records), the computational cost was manageable, but scalability to larger datasets or real-time applications would require further consideration.

Practical implications: The improved SVR model can support palm oil supply chain planning (e.g., estimating future stock availability), agricultural policy design in Southeast Asia, and investment decision-making in the palm oil sector. Producers and policymakers in Indonesia and Malaysia the world's largest palm oil producers can use such forecasting tools for annual production target-setting and commodity trade planning.

Limitations: This study uses a single public dataset and focuses on one SVR variant. The model has not been tested on other agricultural commodities or real-time streaming data. Additionally, $R^2 = 0.794$ indicates that approximately 20.6% of production variance remains unexplained, which may reflect missing variables such as climate data, soil conditions, or country-specific policy factors. A further limitation, raised during review, is that Grid Search in this study was configured with a coarser sampling resolution (8 fixed combinations) than Bayesian Optimization (30 adaptive evaluations); future work should re-run Grid Search at a finer resolution to establish whether it can approach the Bayesian Optimization result under an equal evaluation budget

4. CONCLUSIONS

This study demonstrates that SVR performance for global palm oil production forecasting (1961–2021) can be significantly improved through Balanced-Moderate hyperparameter tuning. Using a sliding window preprocessing approach (3 predictors, 1 target), default hyperparameter values from the `e1071` (R) library, and incremental Balanced-Moderate tuning, the Bayesian Optimization technique achieved the best results: MSE = 0.00066, RMSE = 0.02568, MAE = 0.007, and $R^2 = 0.79373$. These results represent a 99.9% reduction in MSE and a 111.2% improvement in R^2 compared to the untuned baseline. Grid Search and Random Search also improved over the baseline, but were significantly outperformed by Bayesian Optimization.

The contribution of this study is twofold: (1) introducing the Balanced-Moderate strategy as a systematic, incremental approach to hyperparameter tuning in SVR; and (2) demonstrating its effectiveness on a real-world agricultural commodity forecasting dataset with practical implications for supply chain and policy planning in palm oil-producing regions.

Limitations include the use of a single dataset without climate or soil features, the absence of comparison with other ML models (e.g., Random Forest, LSTM), and the lack of statistical significance testing. Future work should: (1) validate the Balanced-Moderate approach on multi-commodity datasets (e.g., rubber, cocoa, soybean); (2) integrate climate variables (rainfall, temperature, humidity) to improve explanatory power; (3) compare with ensemble methods and deep learning models; (4) implement statistical significance tests for performance differences; and (5) explore real-time forecasting applications for operational decision support in the palm oil industry; and (6) re-run Grid Search with a finer, logarithmically-spaced resolution and a comparable evaluation budget to Bayesian Optimization to directly test whether search resolution accounts for the full performance gap observed in Table 5

ACKNOWLEDGMENTS

I extend my immeasurable appreciation to the promoters in Universitas Dian Nuswantoro, who always provided encouragement, insights, and guidance while this writing was being created. In addition, I would like to express my heartfelt thanks to the rectorate team of Universitas Muhammadiyah Sukabumi for providing financial support and the opportunity to develop my academic career. I also do not forget my family and colleagues, who have shared their experiences and attention throughout the research and publication of this writing

REFERENSI

- [1] M. Jeong, S. Kim, E. Yi, and K. Ahn, "Market efficiency and information flow between the crude palm oil and crude oil futures markets," *Energy Strateg. Rev.*, vol. 45, no. December 2022, p. 101008, 2023, doi: 10.1016/j.esr.2022.101008.
- [2] A. J. Astari, J. C. Lovett, and M. Wasesa, "Sustainable pathways in Indonesia's palm oil industry through historical institutionalism," *World Dev. Sustain.*, vol. 6, no. 229, p. 100200, 2025, doi: 10.1016/j.wds.2024.100200.
- [3] N. Widiastuti, R. S. Silitonga, H. N. C. Dharma, J. Jaafar, A. R. Widyanto, and M. Purwanto, "Decreasing free fatty acid of crude palm oil with polyvinylidene fluoride hollow fiber membranes using a combination of chitosan and glutaraldehyde," *RSC Adv.*, vol. 12, no. 35, pp. 22662–22670, 2022, doi: 10.1039/d2ra04005k.

- [4] L. Ifa, T. Syarif, S. Sartia, J. Juliani, N. Nurdjannah, and H. S. Kusuma, "Techno-economics of coconut coir bioadsorbent utilization on free fatty acid level reduction in crude palm oil," *Heliyon*, vol. 8, no. 3, p. e09146, 2022, doi: 10.1016/j.heliyon.2022.e09146.
- [5] F. Hariyanti, A. Syahza, Zulkarnain, and Nofrizal, "Economic transformation based on leading commodities through sustainable development of the oil palm industry," *Heliyon*, vol. 10, no. 4, p. e25674, 2024, doi: 10.1016/j.heliyon.2024.e25674.
- [6] K. Sukiyono et al., "Smallholder palm oil and sustainable development goals (SDGs) achievement: An empirical analysis," *Sustain. Futur.*, vol. 8, no. March, p. 100233, 2024, doi: 10.1016/j.sfr.2024.100233.
- [7] W. Y. Cheah, R. P. Siti-Dina, S. T. K. Leng, A. C. Er, and P. L. Show, "Circular bioeconomy in palm oil industry: Current practices and future perspectives," *Environ. Technol. Innov.*, vol. 30, 2023, doi: 10.1016/j.eti.2023.103050.
- [8] B. Jin and X. Xu, "Price forecasting through neural networks for crude oil, heating oil, and natural gas," *Meas. Energy*, vol. 1, no. February, p. 100001, 2024, doi: 10.1016/j.meae.2024.100001.
- [9] S. C. Iweka, O. Akpomedaye, A. O. Emu, and T. F. Adepoju, "Impact of optimization parameters on green sustainable oil from ripe palm kernel seeds for energy utilities: A machine learning approach," *Sci. African*, vol. 23, no. September 2023, p. e02097, 2024, doi: 10.1016/j.sciaf.2024.e02097.
- [10] V. Vytla, B. Baduru, S. S. Kolukula, N. N. Ragav, and J. P. Kumar, "Forecasting of sea surface temperature using machine learning and its applications," *J. Earth Syst. Sci.*, vol. 134, no. 1, 2025, doi: 10.1007/s12040-024-02483-0.
- [11] S. Albahra et al., "Seminars in Diagnostic Pathology Artificial intelligence and machine learning overview in pathology & laboratory medicine : A general review of data preprocessing and basic supervised concepts," vol. 40, no. February, pp. 71–87, 2023, doi: 10.1053/j.semdp.2023.02.002.
- [12] R. Cinca, "International Journal of Human - Computer Studies " "What are they not telling me ?" " Learning machine learning : Understanding the challenges for novices," *Int. J. Hum. - Comput. Stud.*, vol. 196, no. December 2024, p. 103438, 2025, doi: 10.1016/j.ijhcs.2024.103438.
- [13] P. Stodola and R. Šćurek, "Using machine learning in combinatorial optimization: Extraction of graph features for travelling salesman problem," *Knowledge-Based Syst.*, vol. 314, no. June 2024, 2025, doi: 10.1016/j.knosys.2025.113216.
- [14] S. Gentili, G. D. Chiappetta, G. Petrillo, P. Brondi, and J. Zhuang, "Forecasting strong subsequent earthquakes in Japan using an improved version of NESTORE machine learning algorithm," *Geosci. Front.*, vol. 16, no. 3, p. 102016, 2025, doi: 10.1016/j.gsf.2025.102016.
- [15] M. P. Behera, A. Sarangi, D. Mishra, and S. K. Sarangi, "A Hybrid Machine Learning algorithm for Heart and Liver Disease Prediction Using Modified Particle Swarm Optimization with Support Vector Machine," *Procedia Comput. Sci.*, vol. 218, no. 2022, pp. 818–827, 2022, doi: 10.1016/j.procs.2023.01.062.
- [16] C. Geiß, P. Aravena Pelizari, O. Tunçbilek, and H. Taubenböck, "Semi-supervised learning with constrained virtual support vector machines for classification of remote sensing image data," *Int. J. Appl. Earth Obs. Geoinf.*, vol. 125, no. September, 2023, doi: 10.1016/j.jag.2023.103571.
- [17] S. Zhou, D. Yan, and Y. He, "Interpretable Capacity Prediction of RC Bridges Based on Genetic Algorithm-enhanced Support Vector Machine Learning," *KSCE J. Civ. Eng.*, vol. 28, no. 10, pp. 4559–4574, 2024, doi: 10.1007/s12205-024-1975-6.
- [18] B. Bargam, A. Boudhar, C. Kinnard, H. Bouamri, K. Nifa, and A. Chehbouni, "Evaluation of the support vector regression (SVR) and the random forest (RF) models accuracy for streamflow prediction under a data-scarce basin in Morocco," *Discov. Appl. Sci.*, vol. 6, no. 6, 2024, doi: 10.1007/s42452-024-05994-z.
- [19] J. Zhang and Q. Han, "Buffer Monitoring of Critical Chain Projects Based on Support Vector Machine Prediction," *KSCE J. Civ. Eng.*, vol. 27, no. 7, pp. 2745–2755, 2023, doi: 10.1007/s12205-023-0033-0.
- [20] R. Li et al., "Differential Evolution-Optimized Multi-Output Support Vector Regression-Based Prediction of Weld Bead Morphology in Wire-Fed Laser-Arc Directed Energy Deposition of 2319 Aluminum Alloy," *Addit. Manuf. Front.*, vol. 4, no. 2, p. 200203, 2025, doi: 10.1016/j.amf.2025.200203.

- [21] M. De Iuliis, E. Miceli, and P. Castaldo, "Information theory-guided machine learning to estimate seismic response of non-linear SDOF structures," *Eng. Struct.*, vol. 336, no. April 2024, p. 120448, 2025, doi: 10.1016/j.engstruct.2025.120448.
- [22] S. Der Lu, H. D. Liu, M. H. Wang, and C. C. Wu, "A novel strategy for multitype fault diagnosis in photovoltaic systems using multiple regression analysis and support vector machines," *Energy Reports*, vol. 12, no. April, pp. 2824–2844, 2024, doi: 10.1016/j.egyr.2024.08.074.
- [23] A. Turnip, M. Taufik, and D. E. Kusumandari, "Precision blood pressure prediction leveraging Photoplethysmograph signals using Support Vector Regression," *Egypt. Informatics J.*, vol. 29, no. December 2023, p. 100599, 2025, doi: 10.1016/j.eij.2024.100599.
- [24] K. Kalbe and T. Kalamees, "Evaluating moisture safety strategies in CLT buildings – predictions vs actual outcomes," *Case Stud. Constr. Mater.*, vol. 23, no. March, p. e04958, 2025, doi: 10.1016/j.cscm.2025.e04958.
- [25] T. O. Hodson, "Mean Squared Error , Deconstructed Journal of Advances in Modeling Earth Systems," pp. 1–10, 2021, doi: 10.1029/2021MS002681.
- [26] C. W. Goh, J. Chai, A. Rahman, and W. E. Ong, "Crude Palm Oil Price Prediction Using Simulated Annealing-Based Support Vector Regression (Sa-Svr)," *Asian Acad. Manag. J. Account. Financ.*, vol. 20, no. 1, pp. 305–333, 2024, doi: 10.21315/aamjaf2024.20.1.10.
- [27] N. Khan et al., "Prediction of Oil Palm Yield Using Machine Learning in the Perspective of Fluctuating Weather and Soil Moisture Conditions: Evaluation of a Generic Workflow," *Plants*, vol. 11, no. 13, 2022, doi: 10.3390/plants11131697.
- [28] A. Jierula, S. Wang, and T. Oh, "applied sciences Study on Accuracy Metrics for Evaluating the Predictions of Damage Locations in Deep Piles Using Artificial Neural Networks with Acoustic Emission Data," 2021.
- [29] L. A. Thorsrud, "Centre for Applied Macroeconomic Analysis," 2013.
- [30] M. Salama, "Optimization of Regression Models Using Machine Learning: A Comprehensive Study with Scikit-learn," *Int. Uni-Scientific Res. J.*, vol. 5, no. September, pp. 119–129, 2024, doi: 10.59271/s45500.024.0624.16.
- [31] J. Won, H. S. Lee, and J. W. Lee, "A review on multi-fidelity hyperparameter optimization in machine learning," *ICT Express*, vol. 11, no. 2, pp. 245–257, 2025, doi: 10.1016/j.icte.2025.02.008.
- [32] M. S. Tahosin, M. A. Sheakh, T. Islam, R. J. Lima, and M. Begum, "Optimizing brain tumor classification through feature selection and hyperparameter tuning in machine learning models," *Informatics Med. Unlocked*, vol. 43, no. November, p. 101414, 2023, doi: 10.1016/j.imu.2023.101414.
- [33] P. Fatima, "Optimizing Algorithm Efficiency through Advanced Data Structures in C++: A Comparative Analysis of Performance, Scalability, and Complexity," *Int. J. Comput. Inf. Manuf.*, vol. 3, no. 2, pp. 66–72, 2023, [Online]. Available: <https://journals.gaftim.com/index.php/ijcim/article/view/256>
- [34] V. S. T., "A Study on Hyperparameter Tuning in Support Vector Machines and its Impact on Model Accuracy," *Glob. J. Eng. Innov. Interdiscip. Res.*, vol. 5, no. 1, pp. 1–6, 2025, doi: 10.33425/3066-1226.1063.
- [35] L. W. Rizkallah, "Optimizing SVM hyperparameters for satellite imagery classification using metaheuristic and statistical techniques," *Int. J. Data Sci. Anal.*, 2025, doi: 10.1007/s41060-025-00762-7.
- [36] Y. Sherry and N. C. Thompson, "How Fast Do Algorithms Improve? [Point of View]," *Proc. IEEE*, vol. 109, no. 11, pp. 1768–1777, 2021, doi: 10.1109/JPROC.2021.3107219.
- [37] A. M. Eltamaly, Z. A. Almutairi, and M. A. Abdelhamid, "Modern Optimization Algorithm for Improved Performance of Maximum Power Point Tracker of Partially Shaded PV Systems," *Energies*, vol. 16, no. 13, 2023, doi: 10.3390/en16135228.
- [38] B. Gaye, D. Zhang, and A. Wulamu, "Improvement of Support Vector Machine Algorithm in Big Data Background," *Math. Probl. Eng.*, vol. 2021, 2021, doi: 10.1155/2021/5594899.

- [39] I. S. Al-Mejibli, J. K. Alwan, and D. H. Abd, "The effect of gamma value on support vector machine performance with different kernels," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 5, pp. 5497–5506, 2020, doi: 10.11591/IJECE.V10I5.PP5497-5506.
- [40] Z. Liu and M. J. Zuo, "An Analytical Approach to Fast Parameter Selection of Gaussian RBF Kernel for Support Vector Machine An Analytical Approach to Fast Parameter Selection of Gaussian RBF Kernel for Support Vector Machine *," no. September, 2016.
- [41] A. A. Ewees, A. A. Hemedan, A. E. Hassanien, and A. T. Sahlol, "Optimized support vector machines for unveiling mortality incidence in Tilapia fish," *Ain Shams Eng. J.*, vol. 12, no. 3, pp. 3081–3090, 2021, doi: 10.1016/j.asej.2021.01.014.
- [42] C. Comito and C. Pizzuti, "Artificial intelligence for forecasting and diagnosing COVID-19 pandemic: A focused review," *Artif. Intell. Med.*, vol. 128, no. March, p. 102286, 2022, doi: 10.1016/j.artmed.2022.102286.
- [43] Y. Hahn, T. Langer, R. Meyes, and T. Meisen, "Time Series Dataset Survey for Forecasting with Deep Learning," *Forecasting*, vol. 5, no. 1, pp. 315–335, 2023, doi: 10.3390/forecast5010017.
- [44] H. J. Park, Y. S. Koo, H. Y. Yang, Y. S. Han, and C. S. Nam, "Study on Data Preprocessing for Machine Learning Based on Semiconductor Manufacturing Processes," *Sensors*, vol. 24, no. 17, pp. 1–14, 2024, doi: 10.3390/s24175461.
- [45] A. Azlan, Y. Yusof, and M. F. M. Mohsin, "Determining the impact of window length on time series forecasting using deep learning," *Int. J. Adv. Comput. Res.*, vol. 9, no. 44, pp. 260–267, 2019, doi: 10.19101/ijacr.pid77.
- [46] Purwanto, C. Eswaran, and R. Logeswaran, "Improved adaptive neuro-fuzzy inference system for HIV/AIDS time series prediction," *Commun. Comput. Inf. Sci.*, vol. 253 CCIS, no. PART 3, pp. 1–13, 2011, doi: 10.1007/978-3-642-25462-8_1.
- [47] Z. Zhan and S. K. Kim, "Versatile time-window sliding machine learning techniques for stock market forecasting," *Artif. Intell. Rev.*, vol. 57, no. 8, pp. 1–25, 2024, doi: 10.1007/s10462-024-10851-x.
- [48] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020, doi: 10.1016/j.neucom.2020.07.061.
- [49] R. Guido, M. C. Groccia, and D. Conforti, "A hyper-parameter tuning approach for cost-sensitive support vector machine classifiers," *Soft Comput.*, vol. 27, no. 18, pp. 12863–12881, 2023, doi: 10.1007/s00500-022-06768-8.
- [50] Y. J. Chang, Y. L. Lin, and P. F. Pai, "Support Vector Machines with Hyperparameter Optimization Frameworks for Classifying Mobile Phone Prices in Multi-Class," *Electron.*, vol. 14, no. 11, pp. 1–21, 2025, doi: 10.3390/electronics14112173.
- [51] M. Yalsavar, P. Karimaghaee, A. Sheikh-Akbari, M. H. Khooban, J. Dehmeshki, and S. Al-Majeed, "Kernel Parameter Optimization for Support Vector Machine Based on Sliding Mode Control," *IEEE Access*, vol. 10, pp. 17003–17017, 2022, doi: 10.1109/ACCESS.2022.3150001.
- [52] A. Yaman and A. Cengiz, "The Effects of Kernel Functions and Optimal Hyperparameter Selection on Support Vector Machines," *J. New Theory*, vol. 34, no. September, pp. 64–71, 2021, [Online]. Available: <https://dergipark.org.tr/en/pub/jnt>
- [53] S. Alrefaee, ... S. A. B.-I. J. of, and undefined 2022, "Hyperparameters optimization of support vector regression using black hole algorithm," *ijnaa.semnan.ac.irSD Alrefaee, SM Al Bakal, ZY AlgamalInternational J. Nonlinear Anal. Appl.* 2022•ijnaa.semnan.ac.ir, vol. 13, no. January, pp. 2008–6822, 2022, doi: 10.22075/ijnaa.2022.6107.
- [54] N. Ortiz, O. Avilés, and R. Jiménez, "Epsilon Support Vector Regression based Controller," *Int. J. Appl. Eng. Res.*, vol. 13, no. 19, pp. 14112–14115, 2018, [Online]. Available: <http://www.ripublication.com>
- [55] A. Kocaoğlu, "Efficient Optimization of a Support Vector Regression Model with Natural Logarithm of the Hyperbolic Cosine Loss Function for Broader Noise Distribution," *Appl. Sci.*, vol. 14, no. 9, 2024, doi: 10.3390/app14093641.

- [56] D. C. E. Saputra, A. Ma'arif, and K. Sunat, "Optimizing Predictive Performance: Hyperparameter Tuning in Stacked Multi-Kernel Support Vector Machine Random Forest Models for Diabetes Identification," *J. Robot. Control*, vol. 4, no. 6, pp. 896–904, 2023, doi: 10.18196/jrc.v4i6.20898.
- [57] J. Suárez Ferreira, M. Slavkovik, and J. Casillas, "General procedure to measure fairness in regression problems," *Int. J. Data Sci. Anal.*, 2025, doi: 10.1007/s41060-025-00721-2.
- [58] M. Ogunsanya, J. Isichei, and S. Desai, "Grid search hyperparameter tuning in additive manufacturing processes," *Manuf. Lett.*, vol. 35, pp. 1031–1042, 2023, doi: 10.1016/j.mfglet.2023.08.056.
- [59] Y. Zhao, W. Zhang, and X. Liu, "Grid search with a weighted error function: Hyper-parameter optimization for financial time series forecasting," *Appl. Soft Comput.*, vol. 154, no. January, 2024, doi: 10.1016/j.asoc.2024.111362.
- [60] C. Arnold, "The role of hyperparameters in machine learning models and how to tune them," pp. 841–848, 2024, doi: 10.1017/psrm.2023.61.
- [61] B. Li, "Random Search Plus: A more effective random search for machine learning hyperparameters optimization," pp. 1–95, 2020, [Online]. Available: https://trace.tennessee.edu/utk_gradthes/5849
- [62] P. Cihan, "Bayesian Hyperparameter Optimization of Machine Learning Models for Predicting Biomass Gasification Gases," *Appl. Sci.*, vol. 15, no. 3, 2025, doi: 10.3390/app15031018.
- [63] R. Marco, S. S. S. Ahmad, and S. Ahmad, "Bayesian Hyperparameter Optimization and Ensemble Learning for Machine Learning Models on Software Effort Estimation," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 3, pp. 419–429, 2022, doi: 10.14569/IJACSA.2022.0130351.
- [64] J. Wu, X. C. Hao, Z. L. Xiong, and H. Lei, "Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization," *J. Electron. Sci. Technol.*, vol. 17, no. 1, pp. 26–40, 2019, doi: 10.11989/JEST.1674-862X.80904120.