



Evaluating the Accuracy of a Hybrid Neural Model with RBF-Polynomial Kernel for Rainfall Prediction: A Comparative Analysis of Trainlm and Trainrp Functions

Syahrudin¹, Abdillah², Alfiana Sahraini³, Lilis Suriani⁴

^{1,2,3,4}Department of Mathematics Education, Universitas Muhammadiyah Mataram, Mataram, Indonesia

¹syahrudin.ntb@gmail.com, ²abdillahahmad24041983@gmail.com, ³alfianasahraini@gmail.com,

⁴lilissurianililis28@gmail.com

Abstract

Accurate rainfall prediction is crucial for effective water management and disaster mitigation. This study introduces a novel hybrid neural model that employs a fourth-degree polynomial kernel and provides the first empirical comparison of the trainlm and trainrp functions to enhance forecasting accuracy. This study explored the application of a neural network algorithm with RBF-Polynomial (degree 4) kernel for training and testing data in rainfall forecasting. This study focused on monthly rainfall data collected from Mataram City, Indonesia. We developed a hybrid BP-RVM algorithm as the main algorithm that offers a predictive approach to compare the trainlm and trainrp functions. We conducted 20 trials with combinations of learning, momentum, and gamma-RBF at internal values of 0.01-0.9. The training results from trainrp with more than 118 iterations yielded the best performance with learning rate 0.8 and momentum 0.2; MSE value of 2,236.25 and RMSE of 47.29. These results indicate a relatively low error rate for the proposed method. In contrast, the trainlm method, which only requires 18 iterations with a learning rate of 0.6 and momentum of 0.4, produces an MSE of 2,689.25 and RMSE of 51.86, showing its efficiency in reducing the computation time but with a slightly higher error rate than trainrp. Overall, the trainrp method was more accurate in capturing actual rainfall patterns with lower error rates, whereas the trainlm method exhibited good stability but greater sensitivity to parameter variations. This comparative analysis highlights the potential of trainrp to achieve more precise rainfall predictions within the study area.

Keywords: backpropagation; neural network; RBF-polynomial; relevance vector machine; rainfall

How to Cite: S. Syahrudin, A. Abdillah, Alfiana Sahraini, and Lilis Suriani, "Evaluating the Accuracy of a Hybrid Neural Model with RBF-Polynomial Kernel for Rainfall Prediction: A Comparative Analysis of Trainlm and Trainrp Functions", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 9, no. 5, pp. 1038 - 1046, Oct. 2025.

Permalink/DOI: <https://doi.org/10.29207/resti.v9i5.6388>

Received: February, 12, 2025

Accepted: September 20, 2025

Available Online: October 11, 2025

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

Published by Ikatan Ahli Informatika Indonesia

1. Introduction

Rainfall is a continuous time phenomenon. However, continuous rainfall data is generally not available and where available, is usually limited both temporally and spatially. In contrast, accumulated rainfall over successive time periods is much more universally available where accumulation periods are typically in days, but higher resolution (hours and minutes) is often available for more recent times [1]. The analysis of precipitation data (also known as rainfall) has a long history and has been the focus of a significant amount of research and development over its existence. It is a significant and actively pursued area of research and development with the aim of producing suitable models. The analysis of rainfall data can be undertaken

through multiple methodological frameworks and technological paradigms, encompassing statistical inference, mathematical modeling, and advanced computational approaches such as machine learning [2]. In contemporary research, rainfall forecasting has increasingly relied on computer-based simulation models to enhance predictive capability. Nevertheless, despite the integration of sophisticated computational techniques, the precision of rainfall prediction remains imperfect, with inherent uncertainties persisting. Accurate rainfall estimation, however, is indispensable across various domains, notably in agriculture, water resource management, and aviation operations [3].

The backpropagation (BP) has good computational properties especially when the data presented is large-

scale [4]. BP has been widely and successfully applied in various applications, such as pattern recognition, site selection, and performance evaluation. A conventional Backpropagation (BP) network architecture typically comprises three distinct layers: an input layer responsible for receiving the input data, a hidden layer that performs internal processing, and an output layer that produces the final response corresponding to the provided input [5]. The BP learning process involves the forward propagation of training data through the network, computation of the resulting error, backward propagation of that error, and subsequent adjustment of synaptic weights to minimize the prediction discrepancy [6].

Relevance Vector Machine (RVM) is a combination of a covariance function with a Gaussian process. The advantage of RVM is probabilistic prediction, which does not require regularization of parameter settings and uses arbitrary kernel functions [7]. It also provides inference at a low computational cost [8]. The Relevance Vector Machine (RVM) applies Bayesian inference as an optimization mechanism to achieve reliable results in both regression and classification tasks. The RVM represents a Bayesian-based model that eliminates the need for multiple free parameters commonly required in Support Vector Machines (SVM). Conventional SVM models depend on cross-validation procedures to perform post-training optimization. The RVM method employs several kernel functions in its simulation process, including the polynomial function, the binary sigmoid (logsig) function, and the radial basis function (RBF) [9], [10].

Sofian et al. [11] found the RBF neural network with an average square error (MSE) of 0.00092 during testing, showing its effectiveness compared to the backpropagation model. Furthermore, research by Syaharuddin et al. [12] showed that the combination of the logsig activation function and the trainrp training algorithm in each layer produced the best performance for training, testing, and prediction, achieving an accuracy rate of 99.71%. These results were obtained by configuring several parameters, including 1000 training epochs, a learning rate of 0.7, a goal error of 0.0001, and one training step. Then the research of Zurkarnain et al. [13] shows the implementation on rainfall data shows that the MSE value of the training results is 0.0395384228, and the average deviation value is 3.75382. Meanwhile, the forecasting results for the next three months namely January, February, and March 2018 are 6.1451, 8.5459, and 7.7391, respectively. Mandailina et al. [14] shows that the BP method can be utilized to formulate alternative policy recommendations for addressing and mitigating extreme climate change in the future, particularly during international events held in the Mandalika International Circuit area. Furthermore, Syaharuddin et al. [15] demonstrated that rainfall forecasting using a hybrid Backpropagation–Relevance Vector Machine (BP–RVM) algorithm with a Radial Basis Function

(RBF) kernel achieved significant improvements in prediction accuracy. The BP–RVM model, especially when combined with the RBF–trainrp kernel function, has shown lower error rates in rainfall prediction compared to other configurations, achieving high accuracy mean square error (MSE) and mean absolute percentage error (MAPE) values of (<10%). This hybrid approach effectively integrates the strengths of back propagation and relevance vector machines, enabling better modeling of complex hydro-climatological data.

Saleh et al. [16] have demonstrated the importance of hyper-parameter tuning and the use of extensive historical datasets to improve model performance, as seen in various machine learning applications across different geographical contexts. Overall, the BP–RVM algorithm represents a promising direction for future rainfall forecasting efforts, especially in the context of climate change and its associated challenges. Hussain et al. [17] showed that ensemble classification and regression outperformed traditional models, with an accuracy rate of 83.41% for predicting rainfall events. Kuan-Chenget al. [18] and Surana et al. [19] explained that neural networks algorithm exemplifies the potential of hybrid models to address climate variability challenges, further underscoring the importance of integrating historical weather data with advanced machine learning methods to improve predictive capabilities. Furthermore, Alve et al. [20] integrated machine learning techniques, such as Random Forest and XGBoost and demonstrated promising results in rainfall prediction, with the Random Forest algorithm achieving a high accuracy rate of 91% in Bangladesh.

RBF neural networks combined with a density parameter-based K-means algorithm have been used to improve training speed and accuracy in monthly rainfall forecasting, showing that this approach is superior to the standard Backpropagation model. Furthermore, a hybrid model approach combining Multilayer Perceptron (MLP) and RBF has been applied in weather forecasting, with the aim of improving prediction accuracy and overcoming the limitations of each method [1]. On the other hand, Prahutama and Yasin [21] applied SVR with quadratic loss function and various kernel functions to predict weekly rainfall in Semarang, achieving an R-square of 71.46% on the test data, highlighting the importance of proper kernel function selection in improving prediction accuracy.

Despite the proven effectiveness of BP and RVM in various machine learning applications, limited studies have explored their hybridization for rainfall prediction using polynomial-based kernel combinations. Previous research has predominantly employed standard kernels such as RBF or sigmoid, which may inadequately capture complex, high-order nonlinear relationships inherent in rainfall data. This reliance limits the generalizability and accuracy of the resulting models, particularly in dynamic climatic contexts. Most existing models rely solely on standard kernels such as RBF or

sigmoid, without evaluating the impact of integrating polynomial functions of higher degrees. This study addresses that gap by evaluating the performance of a hybrid BP-RVM model using a combined Radial Basis Function (RBF) and fourth-degree Polynomial kernel. This combination is hypothesized to improve prediction accuracy by capturing both localized and global nonlinear patterns in rainfall data. We hypothesize that integrating a higher-degree polynomial kernel with RBF in a hybrid BP-RVM model significantly enhances rainfall prediction accuracy by capturing both localized and global nonlinear patterns. Furthermore, a comparative analysis of the trainlm and trainrp functions is conducted to determine the optimal training strategy. The novelty of this research lies in (1) the use of RBF-Polynomial hybrid kernel in BP-RVM architecture and (2) the empirical evaluation of two distinct training functions, contributing to more robust and generalizable rainfall forecasting models, especially under conditions of climate uncertainty.

2. Methods

2.1 Data Selection

This research is an experiment by training-testing data using a hybrid backpropagation algorithm and relevance vector machine. The study focuses on monthly rainfall data from Mataram City, Indonesia, collected between January 1, 2014, and December 31, 2024, at coordinates latitude -8.56667 and longitude 116.09611 . The rainfall input data for the Mataram City area were obtained on a monthly basis for each year from 2014 to 2024. The total amount of rainfall recorded for each month is presented in Table 1.

2.2 Research procedure

The kernel function utilized in this study is a combination of the RBF and polynomial functions. Forecasting accuracy is evaluated using MSE and RMSE, with computational support from the MATLAB Graphical User Interface (GUI).

Table 1. Input Rainfall Data of Mataram City from 2014 to 2024

Month	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024
January	442,97	416,6	210,94	195,12	316,41	432,42	305,86	195,12	427,15	329,37	254
February	300,59	184,57	152,93	374,41	337,5	179,3	163,48	189,84	311,13	207,54	454
March	131,84	131,84	221,48	158,2	237,3	189,84	316,41	237,3	195,12	194,07	264
April	121,29	116,02	179,3	63,28	121,29	15,82	126,56	79,1	87,45	118,85	114
May	126,56	5,27	36,91	58,01	21,09	0	5,27	131,84	23,36	74,44	8
June	131,84	0	0	116,02	52,73	10,55	10,55	10,55	105,67	81,09	4
July	21,09	31,64	0	52,73	5,27	5,27	0	0	10,81	23,2	69
August	0	0	0	5,27	0	15,82	0	0	20,7	21,53	0
September	0	0	0	189,84	0	0	0	5,27	68,74	39,29	1
October	15,82	0	0	116,02	94,92	0	0	58,01	106,86	254,71	31
November	126,56	147,66	21,09	216,21	321,68	174,02	10,55	195,12	336,28	237,26	137
December	363,87	305,86	216,21	342,77	311,13	121,29	147,66	321,68	281,59	266,72	303

(Source: NASA website, <https://power.larc.nasa.gov/data-access-viewer/>)

The RBF-Polynomial architecture design within the hybrid BP-RVM algorithm integrates advanced methodologies to enhance predictive accuracy. This architecture leverages the strengths of both RBF and polynomial neural networks, optimizing performance through various design strategies. Generally, an RBF network consists of three main layers: (1) the input layer, which receives the input data; (2) the hidden layer, which contains several nodes that apply radial basis functions (commonly Gaussian) to transform the

input into a higher-dimensional space where the number of nodes must be sufficiently large to ensure effective interpolation; and (3) the output layer, which produces the final output based on the transformations occurring in the hidden layer [22]. This structure forms the foundation for RBF network design, facilitating effective interpolation and function approximation. Additionally, this study adopts a structured yet essential research methodology, as illustrated in Figure 1.

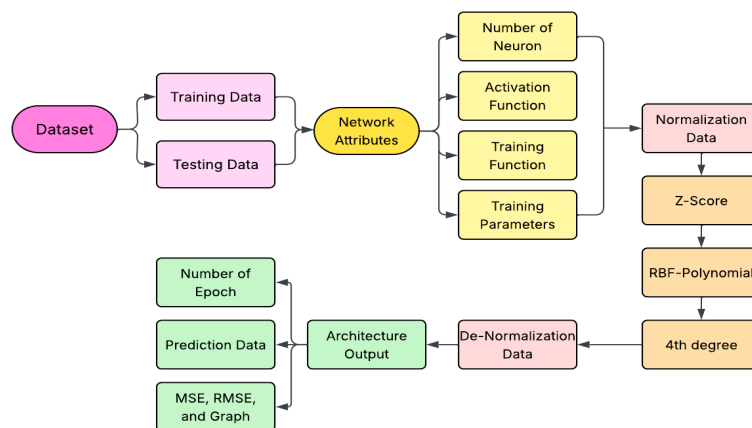


Figure 1. Flowchart of research

Based on Figure 1, there is a research flow that will be carried out in this study. The initial stage begins with a dataset that is the source of data for the model training and testing process. This dataset contains monthly rainfall data in the period 2014 to 2024 obtained from the NASA website. The dataset is divided into two parts: the training data, which is used to train the model to recognize patterns within the data, and the testing data, which is used to evaluate the model's performance after training is completed. Furthermore, the network attributes include the number of neurons, representing the number of hidden layers, and the activation function, which determines the output of each neuron. Training Function is the training algorithm used to update the network weights, training parameters are additional parameters such as, learning rate, momentum and others. Data analysis is in several stages, namely data normalization, determining the degree 4 polynomial equation, conducting training-testing using the BP-RVM algorithm with two trials, namely the trainlm and trainrp functions.

Data normalization uses Z-Score to simplify the network training process (Formula 1). Furthermore, the RBF-Polynomial method uses the Radial Basis Function (RBF) function to calculate the output based on the distance of the input data to a certain center point, while the Polynomial function is used as a data transformation method so that the patterns in the data become more structured (see Formulas 2 and 3). In this research, degree 4 is used (see Formula 8). The choice of the fourth polynomial degree is based on the consideration of the balance between the complexity of the model and the ability to capture the non-linearity of the data. Lower degrees (e.g. 2 or 3) are considered less capable of representing complex patterns in rainfall data, while higher degrees have the potential to cause overfitting, especially in small to medium scale datasets. Therefore, degree 4 was chosen as a suitable compromise to produce a sufficiently complex feature space transformation while still maintaining good model generalization. Next, data normalization is interpreting the prediction results and changing the relevant output model. The last stage of output architecture obtained from the MATLAB R2011a GUI includes several parts, namely, number of epoch full iterations on the training dataset, prediction data produces predictions based on input data, Mean Square Error (MSE), Root Mean Squared Error (RMSE) and Graph (visualization of training and testing results).

$$Z = \frac{x_i - \bar{x}_i}{\sigma} \quad (1)$$

with x_i is data, $\bar{x}_i$ is average, and the standard deviation of all values in the dataset. Then, radial basis function (RBF) kernel (Formula 2) and polynomial kernel (Formula 3):

$$K_{RBF}(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \quad (2)$$

$$K_{Poly}(x, y) = (x^T y + c)^d \quad (3)$$

Previous studies have shown that 2 and 3-degree polynomial kernels have provided fairly competitive performance in prediction and classification. Sofian et al. [23] demonstrated that the use of a second-degree polynomial kernel in a Relevance Vector Machine (RVM) for EEG classification resulted in high accuracy with an efficient model structure. Additionally, in the context of physical systems, a third-degree polynomial kernel exhibited superior predictive performance with lower error rates compared to other kernels [24]. Therefore, in this study, a 4-degree polynomial kernel is used as an effort to enhance the model's capacity to capture more complex nonlinear patterns in rainfall data, while also exploring its effectiveness compared to previous degrees. In this study, a hybrid kernel function was constructed by linearly combining the RBF and polynomial kernels to exploit the local sensitivity of RBF and the global generalization capability of polynomial kernels. The hybrid kernel is defined as:

$$K_{RBF-Poly}(x, y) = \alpha \cdot \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) + \beta \cdot (x^T y + c)^d \quad (4)$$

with $\alpha = 0.5$, $\beta = 0.5$, σ as the spread parameter, c as the offset, and $d = 4$ as the degree of the polynomial kernel. This configuration was selected based on empirical tuning to optimize prediction accuracy. The BP-RVM architecture parameters used are:

Architecture	: 12-25-13-7-1
Max. Epoch	: 1000
Error (Goal)	: 0.001
Learning Rate (LR)	: [0.01-0.9]
Momentum	: [0.1-0.9]
Show Step	: 50
Gamma RBF	: 0.01
Activ. Function	: radbas-logsig-logsig-purelin
Training Function	: trainrp and trainlm

We set the parameters for the BP-RVM architecture at the initial stage, prior to the training and testing phases. A binary sigmoid activation function (logsig) is applied to each hidden layer, as shown in Formula 5 [25],

$$f(x) = \frac{1}{1+e^{-x}} \quad (5)$$

The number of epochs generated, Mean Square Error (MSE) and Root Mean Squared Error (RMSE) are used in testing the accuracy of the developed network architecture. The MAPE category uses the scale developed by Lewis et al. (1962). The MSE and RMSE formulas are shown in Formulas 6 and 7.

$$MSE = \frac{1}{n} \sum_{i=1}^n (A_i - F_i)^2 \quad (6)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n}} \quad (7)$$

with A_i representing actual data, F_i representing forecasting data, and n representing the amount of data. In the data training and testing stages, the number of epochs, MSE and MAPE generated in each simulation are tabulated to facilitate the interpretation process. These parameters are used to find the best architecture for the prediction stage. The architecture uses the number of epochs to determine the duration of the training process. It uses the MSE value to evaluate the average error in recognizing actual data. It also applies the MAPE value to assess whether the forecasting results are acceptable or should be rejected.

3. Results and Discussions

3.1 Initial Analysis Results

From the data in Table 1, the average monthly rainfall values from 2014 to 2024 covering the period January to December were generated. The data shows significant variations in the average value of rainfall each month. January was 320.5418 mm, February was 259.5718 mm, March was 207.0364 mm, April was 103.9055 mm, May was 44.61364 mm, June was 47.54545 mm, July was 19.91 mm, August was 5.756364 mm, September was 27.64909 mm, October was 61, 57636 mm, November was 174.8573 mm, December was 71.0709 mm. The highest rainfall value was recorded in January from 2014 to 2024 with an average of 320.5418 mm, indicating the peak rainfall intensity in 10 years in January. In contrast, the lowest rainfall value was in August from 2014 to 2024 with an average of 5.756364 mm, reflecting the dry season period with very low rainfall intensity. The data fluctuations reflect typical seasonal patterns, where high rainfall occurs at the beginning and end of the year, such as January at 320.5418 mm, February at 259.5718 mm, and December at 271.0709 mm. Meanwhile, the mid-year period, especially from June to August, shows low rainfall with an average of below 50 mm per month. This analysis suggests a strong seasonal pattern that is likely influenced by regional climate phenomena such as the monsoon or other global wind patterns. Understanding these patterns is important to support water resources management and hydro-meteorological disaster mitigation.

3.2 Data Training-Testing Process

Generally, there are two stages in the prediction process using the RBF-Polynomial approach. First, the data in Table 1 is processed by calculating the average value for each month to generate a fourth-degree polynomial equation (see Figure 2 and Formula 8). This polynomial equation is then embedded into the BP-RVM algorithm, particularly in the first hidden layer, and is combined with a radial basis function (RBF) to enhance non-linear learning. Second, the model undergoes training and testing stages. The dataset is divided into 80% for training and 20% for testing to ensure appropriate validation and generalization. The prediction is performed after the training phase is completed. The

architecture applied in this study consists of 12-25-13-7-1, with 12 neurons in the input layer, 25 neurons in the first hidden layer, 13 neurons in the second hidden layer, 7 neurons in the third hidden layer, and 1 neuron in the output layer.

Figure 2 shows the 4th degree polynomial graph approximation with the average actual data determined from the average monthly data from January-December, thus Formula 8 is obtained.

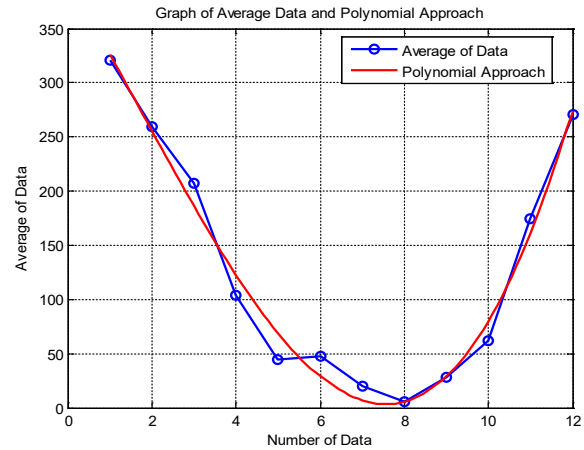


Figure 2. Approximation graph of actual data and polynomial equation

$$y = -0.010475x^4 + 0.85697x^3 - 4.3488x^2 - 62.968x + 392.0368 \quad (8)$$

Formula 8 was derived from the monthly average rainfall data and serves as the foundational input for the hybrid BP-RVM model. As seen in the graph, the polynomial approximation closely follows the general trend of the actual data, capturing both the downward and upward curvature effectively, particularly in the months with low and high rainfall. Although minor deviations are observed in some intervals particularly where the rainfall data fluctuates more sharply the polynomial function provides a smooth and continuous curve that represents the underlying pattern of rainfall distribution. This trend fitting is crucial, as it enhances the initial feature representation fed into the neural network, thereby improving the model's ability to generalize and predict future rainfall patterns. Furthermore, data training and testing is carried out using Formula 6 which has been obtained from the rainfall data in Table 1, the results of data training and testing data trainrp and trainlm are presented in Table 2. Table 2 presents the performance evaluation of Resilient Propagation (trainrp) and Levenberg-Marquardt (trainlm). Each method is tested with varying combinations of learning rate and momentum, which are critical hyper-parameters that influence both convergence speed and prediction accuracy. Additionally, number of iterations, MSE, and RMSE are reported as performance metrics to assess how well the network learns from the training data.

In the trainrp method, learning rates ranging from 0.01 to 0.9 are tested with momentum values from 0.1 to 0.9. The iteration count fluctuates between 97 and 233, reflecting the adaptive behavior of the algorithm under different settings. The lowest error was achieved with a learning rate of 0.8 and momentum of 0.2, producing an MSE of 2236.25 and an RMSE of 47.289 within 118

iterations. This result indicates that a relatively high learning rate combined with low momentum improves convergence and reduces prediction error in this context. Conversely, higher momentum values or extremely low learning rates resulted in significantly higher error rates, suggesting potential instability or slow learning.

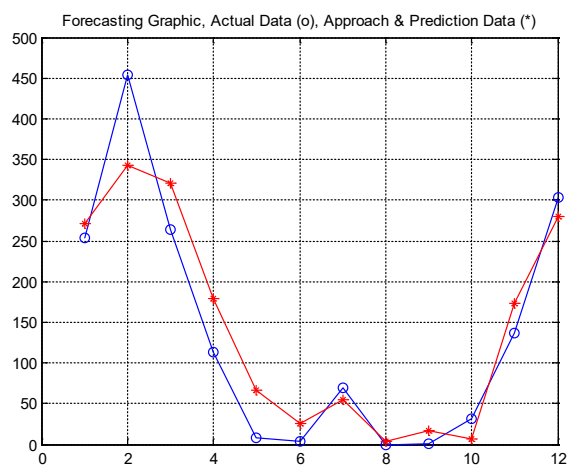
Table 2. Results of Training and Testing Data

Training	Learning Rate	Momentum	Iterations	MSE	RMSE
trainrp	0.01	0.9	123	8379.48	91.5395
	0.1	0.9	233	10331.2	101.642
	0.2	0.8	120	5743.91	75.7886
	0.3	0.7	113	4123.44	64.214
	0.4	0.6	97	4373.22	66.1304
	0.5	0.5	99	7881.86	88.7798
	0.6	0.4	124	8069.27	89.8291
	0.7	0.3	125	10131.5	100.655
	0.8	0.2	118	2236.25	47.289
trainlm	0.9	0.1	112	8458.62	91.9708
	0.01	0.9	15	24623.4	156.918
	0.1	0.9	12	6249.14	79.0515
	0.2	0.8	8	10459.9	102.274
	0.3	0.7	12	16154.2	127.099
	0.4	0.6	14	24354.1	156.058
	0.5	0.5	14	9196.73	95.8996
	0.6	0.4	18	2689.25	51.858
	0.7	0.3	17	14848.7	121.855
	0.8	0.2	10	16037.7	126.64
	0.9	0.1	9	15798.4	125.692

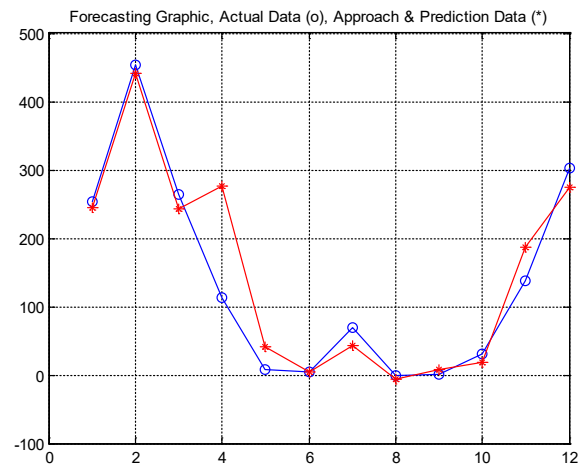
For the trainlm method, the number of iterations required to converge is noticeably lower, ranging from 8 to 18. Despite this efficiency, trainlm exhibits greater variability in performance. The most optimal result for this method was achieved at a learning rate of 0.6 and momentum of 0.4, with MSE of 2689.25 and RMSE of 51.858 after 18 iterations. While this is slightly less accurate than the best result of trainrp, trainlm demonstrates greater computational efficiency and faster convergence due to its second-order optimization strategy. However, its sensitivity to parameter selection remains a critical consideration, as several

configurations yielded RMSE values exceeding 120.

The trainrp algorithm provides higher accuracy but requires more iterations, while trainlm offers faster convergence with moderate accuracy. These findings suggest that trainrp is more suitable for tasks demanding high precision, whereas trainlm is advantageous in scenarios with limited training time or computational resources. The training and testing visualizations of both methods are further illustrated in Figure 3.



(a) Simulation result graph of trainrp



(b) Simulation result graph of trainlm

Figure 3. Data Simulation Results

3.2 Forecasting Results

Based on the analysis of Table 2 the effect of Learning

Rate shows that large values often require more iterations such as, at a Learning Rate of 0.8 on *trainrp* which requires 118 iterations. But in *trainlm*, a small Learning Rate results in a relatively large MSE value. In addition, the Momentum value also affects the training results. This parameter affects the number of iterations required as well as the evaluation values in the form of MSE and RMSE. In the *trainrp* method, the number of iterations varies between 97 to 233 iterations with the lowest MSE value of 2236.25 and RMSE of 47,289 achieved in the combination of learning rate 0.8 and Momentum 0.2. In contrast, the *trainlm* method has a smaller number of iterations, between 8 to 18 iterations, but produces higher MSE and RMSE values such as, in the combination of Learning Rate 0.01 and Momentum 0.9 with MSE of 24623.4 and RMSE of 156,918. However, between 8 to 18 iterations the lowest MSE value of 2689.25 and RMSE of 51,858 is achieved in the combination of Learning Rate 0.6 and Momentum 0.4.

Generally, the *trainrp* method shows better performance in reducing MSE and RMSE values than *trainlm*, despite requiring more training iterations. In contrast, the *trainlm* function is more suitable for scenarios where shorter training time is prioritized, even though it results in higher error values. The superior performance of *trainrp* can be attributed to its stability in gradient updates, as it employs the resilient backpropagation algorithm, which adjusts weight updates based solely on the sign of the gradient. This approach reduces sensitivity to gradient magnitude and helps avoid oscillations on complex error surfaces. These findings indicate that selecting the appropriate combination of training parameters plays a critical role in optimizing model performance.

The performance differences between the training algorithms were evaluated using the Wilcoxon signed-rank test, based on MSE and RMSE values obtained from the ten experimental settings in Table 2. The test results show that the *trainrp* function consistently produced lower error values compared to *trainlm*, with average MSE and RMSE values of 7,102.91 and 79.21, respectively. In contrast, *trainlm* yielded higher average values of 13,921.43 for MSE and 119.73 for RMSE. The Wilcoxon test revealed a statistically significant difference between the two methods, with p-values of 0.005 for MSE and 0.003 for RMSE, indicating that the superior performance of *trainrp* is unlikely to be due to random variation. These findings confirm that *trainrp* more stable gradient update mechanism based on the sign of the gradient rather than its magnitude provides better resistance to oscillations and local minima, particularly in highly nonlinear rainfall data. As a result, *trainrp* can be considered a more robust and reliable training strategy for hybrid neural models in rainfall prediction tasks.

Velasco et al. [26] found that rainfall prediction using various machine learning techniques has produced significant results in terms of MSE and RMSE. For

example, the support vector regression machine (SVRM) model showed an MSE of 3.46, demonstrating its effectiveness in predicting rainfall in tropical regions. Furthermore, research by Setiawan et al. [27] revealed an MSE value of 0.02769 with a well-tuned learning and iteration rate, indicating effective parameter optimization. Thus, Syaharuddin et al. [28] concluded that the Z-score technique performs effectively in normalizing rainfall data, with 10 epochs, a MAE of 0.051, and an MSE of 0.004. For air humidity data, both the mean-MAD and Z-score techniques are recommended. The mean-MAD technique achieved 8 epochs, a MAE of 0.013, and an MSE of 0.0004, while the Z-score technique reached 7 epochs, a MAE of 0.018, and an MSE of 0.0006. Furthermore, Ihsan et al. [29] conducted rainfall prediction in Makassar City using the Backpropagation method with a 12-32-16-8-1 network architecture. The study applied 350 epochs and learning rates of 0.01, 0.001, 0.0001, and 0.00001. The results demonstrated that the Backpropagation approach achieved superior performance, producing an RMSE of 137.32 or 0.11, which indicates high accuracy in forecasting trend variations and seasonal patterns.

Research by Usman et al. [30] in Semarang used a neural network algorithm that produced an RMSE of 13.055 and MAE of 6.621, showing superior performance compared to other algorithms. In addition, Prem Kumar et al. [31] also conducted research to evaluate the performance of several machine learning models. Among them, the Lasso Regression model achieved the best performance, with an R^2 score of 99.21%, MAE of 13.68, MSE of 6432.41, and RMSE of 80.20 when using 80% of the data for training and 20% for testing, showing superior prediction performance. These findings confirm that proper selection and parameter settings in machine learning algorithms are critical to improving rainfall prediction accuracy.

Generally, the *trainrp* method demonstrates better performance in reducing MSE and RMSE values compared to the *trainlm* method. However, this advantage comes at the cost of a higher number of iterations, making this method less efficient for implementation in systems with time or computational resource constraints. Conversely, the *trainlm* method is superior in terms of training time efficiency as it requires fewer iterations, although it produces a higher error rate. The training results of both methods are greatly influenced by the variation in the combination of Learning Rate and Momentum parameters, indicating that the parameter tuning process plays a crucial role in optimizing model performance.

These findings have important implications in practical applications. In real-time prediction systems such as rainfall prediction, agricultural planning, or early warning systems, the choice of training method must consider the trade-off between accuracy and computational efficiency. The *trainlm* method can be chosen for systems requiring rapid response, even with

tolerance for increased error values. Conversely, the trainrp method is more suitable for historical data analysis or offline applications emphasizing high accuracy, without pressure on training duration.

Furthermore, these results underscore the need for the application of automatic parameter optimization strategies, such as grid search or Bayesian optimization, especially for nonlinear and dynamic data such as climate and weather data. For future research, it is recommended that model development focus on integrating adaptive training techniques, applying dynamic regularization, and conducting more extensive cross-validation. Additionally, testing on complex and varied field data is needed to enhance the model's stability and generalization capabilities. As a result, the resulting model will be more relevant in supporting decision-making in strategic sectors dependent on climate predictions, such as agriculture, disaster mitigation, and water resource management.

4. Conclusions

Based on the experimental results, this study concludes that the trainrp method demonstrates superior performance in minimizing MSE and RMSE values, although it requires a greater number of iterations. The optimal parameter setting for trainrp was found at a Learning Rate of 0.8 and Momentum of 0.2, achieving an MSE of 2236.25 and RMSE of 47.29 with 233 iterations. In contrast, the trainlm method exhibited advantages in training time efficiency, requiring only 8 to 18 iterations, with its best performance obtained at a Learning Rate of 0.6 and Momentum of 0.4, resulting in an MSE of 2689.25 and RMSE of 51.86 in just 18 iterations. Therefore, trainrp is more appropriate for applications where prediction accuracy is prioritized, while trainlm is better suited for real-time or computationally limited environments where faster convergence is more critical, despite a slightly higher error rate. The scientific contribution of this study lies in providing a comparative analysis of two commonly used training algorithms, trainrp and trainlm, under varying learning rates and momentum parameters, applied within a novel hybrid RBF-polynomial kernel neural network architecture for rainfall prediction. This specific model variant, to the best of our knowledge, represents the first empirical implementation combining a fourth-degree polynomial kernel with RBF in a BP-based hybrid network, offering enhanced capacity to capture both localized and global nonlinear patterns in meteorological data.

The results provide empirical evidence that careful tuning of learning parameters significantly influences model performance and can inform the selection of training algorithms in related neural network-based forecasting tasks. This study is limited by the use of a small, domain-specific dataset and the comparison of only two training algorithms, which may restrict the

generalizability of the results. Key hyper-parameters such as RBF gamma and weight initialization were not explored. Future research should include additional training methods, diverse datasets, and advanced techniques like adaptive learning rates, regularization, and kernel tuning. Real-time application in dynamic fields such as climate forecasting or early warning systems is also recommended to evaluate practical feasibility.

Acknowledgment

The author would like to express sincere gratitude to the Research and Development Institute (LPP) of the Central Board of Muhammadiyah for the financial support provided through the Muhammadiyah Research Grant Program (RisetMu) under contract number 0258.551/I.3/D/2025. This support has significantly contributed to the implementation and publication of this research.

References

- [1] G. Giani, L. Tarasova, R. A. Woods, and M. A. Rico-Ramirez, "An Objective Time-Series-Analysis Method for Rainfall-Runoff Event Identification," *Water Resour. Res.*, vol. 58, no. 2, pp. 1–18, 2022, doi: 10.1029/2021WR031283.
- [2] A. Al Mutairi, "A new alpha logarithmic-generated class to model precipitation data with theory and inference," *Heliyon*, vol. 9, no. 9, p. e19561, 2023, doi: 10.1016/j.heliyon.2023.e19561.
- [3] E. G. Wahyuni, L. M. F. Fauzan, F. Abriyani, N. F. Muchlis, and M. Ulfa, "Rainfall prediction with backpropagation method," in *Journal of Physics: Conference Series*, 2018, doi: 10.1088/1742-6596/983/1/012059.
- [4] L. G. Wright et al., "Deep physical neural networks trained with backpropagation," *Nature*, vol. 601, no. 7894, pp. 549–555, 2022, doi: 10.1038/s41586-021-04223-6.
- [5] D. Huang and Z. Wu, "Forecasting outpatient visits using empirical mode decomposition coupled with backpropagation artificial neural networks optimized by particle swarm optimization," *PLoS One*, vol. 12, no. 2, pp. 1–17, 2017, doi: 10.1371/journal.pone.0172539.
- [6] J. Tarigan, Nadia, R. Diedan, and Y. Suryana, "Plate Recognition Using Backpropagation Neural Network and Genetic Algorithm," in *Procedia Computer Science*, 2017, pp. 365–372, doi: 10.1016/j.procs.2017.10.068.
- [7] S. M. Acosta, A. L. Amoroso, Á. M. O. Sant'Anna, and O. C. Junior, "Predictive modeling in a steelmaking process using optimized relevance vector regression and support vector regression," *Ann. Oper. Res.*, vol. 316, no. 4, pp. 905–926, 2022, doi: <https://doi.org/10.1007/s10479-021-04053-9>.
- [8] A. Bharathi and K. Anandakumar, "Cancer classification using relevance vector machine learning approach," *J. Med. Imaging Heal. Informatics*, vol. 5, no. 3, pp. 630–634, 2015, doi: 10.1166/jmih.2015.1412.
- [9] R. K. Agrawal, F. Muchahary, and M. M. Tripathi, "Ensemble of relevance vector machines and boosted trees for electricity price forecasting," *Appl. Energy*, vol. 250, pp. 540–548, 2019, doi: 10.1016/j.apenergy.2019.05.062.
- [10] Q. Bing, B. Gong, Z. Yang, Q. Shang, and X. Zhou, "Short-Term Traffic Flow Local Prediction Based on Combined Kernel Function Relevance Vector Machine Model," *Math. Probl. Eng.*, vol. 2015, 2015, doi: 10.1155/2015/154703.
- [11] I. M. Sofian, A. K. Affandi, I. Iskandar, and Y. Apriani, "Monthly rainfall prediction based on artificial neural networks with backpropagation and radial basis function," *Int. J. Adv. Intell. Informatics*, vol. 4, no. 2, pp. 154–166, 2018, doi: 10.26555/ijain.v4i2.208.
- [12] Syaharuddin, D. Pramita, T. Nusantara, Subanji, and H. R. P.

- Negara, "Analysis of accuracy parameters of ANN backpropagation algorithm through training and testing of hydro-climatology data based on GUI MATLAB," in *IOP Conference Series: Earth and Environmental Science*, Institute of Physics Publishing, Jan. 2020. doi: 10.1088/1755-1315/413/1/012008.
- [13] D. Z. Zurkarnain, A. Damayanti, and E. Winarko, "Hybrid Jaringan Saraf Tiruan Backpropagation dengan Firefly Algorithm dan Simulated Annealing untuk Peramalan Curah Hujan di Surabaya," *Contemp. Math. Appl.*, vol. 3, no. 1, p. 56, 2021, doi: 10.20473/conmatha.v3i1.26942.
- [14] V. Mandailina, A. Nurhalimah, S. Mehmood, Syaharuddin, and Ibrahim, "Study of Climate Change in the Mandalika International Circuit Area Using Neural Network Backpropagation," *Rev. d'Intelligence Artif.*, vol. 36, no. 6, pp. 847–853, 2022, doi: 10.18280/ria.360604.
- [15] Syaharuddin, Fatmawati, H. Suprajitno, and Ibrahim, "Hybrid Algorithm of Backpropagation and Relevance Vector Machine with Radial Basis Function Kernel for Hydro-Climatological Data Prediction," *Math. Model. Eng. Probl.*, vol. 10, no. 5, pp. 1706–1716, 2023, doi: 10.18280/mmep.100521.
- [16] M. A. Saleh, H. M. Rasel, and B. Ray, "A comprehensive review towards resilient rainfall forecasting models using artificial intelligence techniques," *Green Technol. Sustain.*, vol. 2, no. 3, p. 100104, 2024, doi: 10.1016/j.grets.2024.100104.
- [17] A. Hussain, A. Aslam, S. Tripura, V. Dhanawat, and V. Shinde, "Weather Forecasting Using Machine Learning Techniques: Rainfall and Temperature Analysis," *J. Adv. Inf. Technol.*, vol. 15, no. 12, pp. 1329–1338, 2024, doi: 10.12720/jait.15.12.1329-1338.
- [18] K.-C. Lin, W.-T. Chen, P.-L. Chang, Z.-Y. Ye, and C.-C. Tsai, "Enhancing the Rainfall Forecasting Accuracy of Ensemble Numerical Prediction Systems via Convolutional Neural Networks," *Artif. Intell. Earth Syst.*, vol. 3, no. 4, pp. 1–22, 2024, doi: <https://doi.org/10.1175/AIES-D-23-0105.1>.
- [19] A. Surana, S. E. Pawar, S. Raha, N. Mali, and T. Mukherjee, "Ensemble Fine Tuned Multi Layer Perceptron For Predictive Analysis Of Weather Patterns and Rainfall Forecasting From Satellite Data," *ICTACT J. Soft Comput.*, vol. 15, no. 2, pp. 3491–3496, 2024, doi: <https://www.i-scholar.in/index.php/IJSC/article/view/226424>.
- [20] S. R. Alve, A. J. Pial, M. Z. Mahmud, and S. Islam, "Harnessing Historical Weather Data with Machine Learning for Rainfall Forecasting in Bangladesh: A Time Series Analysis," *Cogniz. J. Multidiscip. Stud.*, vol. 4, no. 8, pp. 274–289, 2024, doi: 10.47760/cognizance.2024.v04i08.017.
- [21] A. Prahutama and H. Yasin, "Prediction of Weekly Rainfall in Semarang City Use Support Vector Regression (SVR) with Quadratic Loss Function," *Int. J. Sci. Eng.*, vol. 9, no. 1, pp. 13–16, 2015, doi: <https://doi.org/10.12777/ijse.0.0>.
- [22] L. Zhang, C. Chen, Y. Xia, Q. Song, and J. Cao, "Prediction of Blade Tip Timing Sensor Waveforms Based on Radial Basis Function Neural Network," *Appl. Sci.*, vol. 13, no. 17, pp. 1–19, 2023, doi: 10.3390/app13179838.
- [23] E. Dong, G. Zhu, C. Chen, J. Tong, Y. Jiao, and S. Du, "Introducing chaos behavior to kernel relevance vector machine (RVM) for four-class EEG classification," *PLoS One*, vol. 13, no. 6, pp. 1–19, 2018, doi: 10.1371/journal.pone.0198786.
- [24] C. McCall et al., "Distinguishing radiation mechanisms and particle populations in blazar jets through long-term multiband monitoring with RINGO3 and Fermi," *Mon. Not. R. Astron. Soc.*, vol. 532, no. 2, pp. 2788–2819, 2024, doi: 10.1093/mnras/stae1693.
- [25] N. Fariq, Y. Visuvanathan, S. H. Syam, A. Jamil, and M. S. Mohd, "Log-sigmoid Activation Function based MLP Network for Aggregate Classification," *J. Kejuruter.*, vol. 37, no. 1, pp. 471–475, 2025, doi: [https://doi.org/10.17576/jukm-2025-37\(1\)-34](https://doi.org/10.17576/jukm-2025-37(1)-34).
- [26] L. C. Velasco, J. M. Aca-ac, J. J. Cajes, N. J. Lactuan, and S. C. Chit, "Rainfall Forecasting using Support Vector Regression Machines," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 3, pp. 231–237, 2022, doi: 10.14569/IJACSA.2022.0130329.
- [27] W. Setiawan, A. Barokah, and Mula'ab, "Rainfall Prediction Using Backpropagation with Parameter Tuning," *MATEC Web Conf.*, vol. 372, p. 07003, 2022, doi: 10.1051/mateconf/202237207003.
- [28] Syaharuddin, Fatmawati, and H. Suprajitno, "Long-term forecasting for climate classification using hybrid backpropagation relevance vector machine algorithm," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 1441, no. 1, pp. 1–6, 2025, doi: 10.1088/1755-1315/1441/1/012010.
- [29] H. Ihsan, A. Illa, and E. Nensi, "Implementation Of Backpropagation And Hybrid Arima-Nn Methods In Predicting Accuracy Levels Of Rainfall In Makassar City," *BAREKENG J. Ilmu Mat. dan Terap.*, vol. 18, no. 4, pp. 2435–2448, 2024, doi: <https://doi.org/10.30598/barekengvol18iss4pp2435-2448>.
- [30] C. D. Usman, A. P. Widodo, K. Adi, and R. Gernowo, "Rainfall prediction model in Semarang City using machine learning," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 30, no. 2, pp. 1224–1231, 2023, doi: 10.11591/ijeecs.v30.i2.pp1224-1231.
- [31] P. Kumar, R. Lakshmi, and B. Behera, "Performance Analysis and Evaluation of Machine Learning Algorithms in Rainfall Prediction," *Int. J. Adv. Sci. Technol.*, vol. 29, no. 5, pp. 5727–5741, 2020, doi: <https://sersc.org/journals/index.php/IJAST/article/view/15057>