PERFORMANCE OF MACHINE LEARNING FOR IMPUTING MISSING DAILY RAINFALL DATA IN EAST JAVA UNDER MULTIPLE SATELLITE DATA MODELS

Lilis SRIWAHYUNI [1], ***Sri NURDIATI*** [1*], ***Endar Hasafah NUGRAHANI*** [1], ***Mohamad Khoirun NAJIB*** [1]

DOI: 10.21163/GT_2025.201.23

ABSTRACT

The estimation and monitoring of rainfall patterns are crucial for hydrological system modelling. However, station data in Indonesia often need to be completed, making analysis using these data problematic. One way to address incomplete data is through imputation or filling in missing data by utilizing other available information. This study aims to identify the most accurate machine learning method and satellite dataset for imputing missing daily rainfall data at BMKG stations in East Java. The four machine learning methods used are Multiple Linear Regression (MLR), Convolutional Neural Network (CNN), Multiple Layer Perceptron (MLP), and Support Vector Regression (SVR). The satellite datasets used are ERA5, ERA5 Land, CMORPH CRT, CMORPH BLD, and CHIRPS. The data were divided into training and testing sets with varying ratios of 95:5%, 90:10%, 80:20%, 70:30%, and 50:50%. Based on the analysis of data proportions, scenarios with a more significant proportion of training data, such as 95:5%, yield better performance. MLP demonstrates the most significant potential for further analysis regarding machine learning methods due to its lowest loss value, measured using Mean Absolute Error (MAE). Regarding satellite data, ERA5 is more stable and reliable for rainfall imputation modelling. Combining the MLP method and ERA5 satellite data delivers the best results with the lowest loss value and minimizes the risk of overfitting. This study significantly improves the quality of rainfall data and supports more accurate meteorological analyses in Indonesia.

Key-words: *Imputation missing data; Machine Learning; Mean Absolute Error; Satellite Data; Station Data*

1. INTRODUCTION

Water plays a crucial role in sustaining the lives of humans, animals, plants, and the entire natural ecosystem. One natural water source is rainfall, which significantly impacts the water resources on Earth. The amount of rainfall that falls to the Earth's surface over a specific period is called precipitation. Rainfall in Indonesia is one of the atmospheric parameters that is difficult to predict over a relatively long period due to its spatial and temporal variability (Septiawan et al., 2017). Accurate estimation and monitoring of rainfall patterns are essential for hydrological system modelling as well as for the planning and management of water resources in various sectors of society (Soares et al., 2016), such as hydroelectric power generation, human water supply, agriculture, land use planning, and drought management and mitigation (Vicente et al., 2019).

Rainfall data is obtained through rain gauges, which directly measure rainfall at specific locations and are considered reference sources for rainfall observation (Falck et al., 2015). However, rain gauges may not effectively capture spatial variations in rainfall across large areas or regions with complex topography (Falck et al., 2015; Nóbrega et al., 2008; Wagner et al., 2012). Additionally, technical or human mistakes often affect the recording process, leading to incomplete data. Consequently, limitations hinder accurate explanations of the stochastic processes of rainfall at the observed locations (de Oliveira et al., 2008). If not carefully addressed, missing data can introduce significant analytical bias in addition to reducing statistical power, jeopardizing the trustworthiness of the research and any interpretive findings that follow (Magyari-Sáska et al., 2025). Observational data from stations is crucial in climatological data analysis (Desmonda, 2018; Rafhida, 2024), as it is

[1]*Department of Mathematics, Faculty of Mathematics and Natural Sciences, IPB University, Bogor, Indonesia. (LS) lilissriwahyuni@apps.ipb.ac.id,* [*]*Corresponding author: (SN) nurdiati@apps.ipb.ac.id,(EHN) e_nugrahani@apps.ipb.ac.id, (MKN) mohknajib@gmail.com.*

directly collected from the location and represents actual data. However, due to the incomplete station data in Indonesia, analyzing rainfall data using station data becomes problematic. As a result, many researchers have turned to satellite data for their analyses (Nurdiati et al., 2021; Li, 2020; Najib et al., 2022). However, satellite data, which a specific model typically generates, inevitably contains errors. Consequently, the analysis results will also contain errors. One method researcher used to reduce the mistakes in satellite data is bias correction (Partarini et al., 2021; Nurdiati et al., 2021), but this approach only helps to reduce errors, particularly systematic errors (bias). Therefore, satellite data analysis is only sometimes better than station data analysis. As a result, one way to handle incomplete station data for accurate analysis is through missing data imputation. Imputation is estimating missing values by replacing them with values calculated precisely (Silva-Ramírez et al., 2011). One approach that can be used for missing data imputation is machine learning (Wangwongchai et al., 2023). Jerez et al. (2010) conducted a study comparing missing data imputation using machine learning-based methods and statistical methods. The results showed that machine learning methods were more accurate in imputing missing data than statistical methods.

A similar study was conducted by Duarte et al. (2022), which compared direct imputation using satellite data from IMERG-GPM (Global Precipitation Measurement method) and imputation using station data with the Simple Linear Regression (SLR) and Multiple Linear Regression (MLR) methods for rainfall station data in Brazil. The results showed that the MLR method had better accuracy at some stations, while at other stations, IMERG-GPM outperformed in terms of accuracy. Therefore, the MLR method was combined with several satellite datasets for imputing missing rainfall data at BMKG stations in East Java. Yang et al. (2022) investigated missing data imputation using Beidou satellite data in China with Linear Interpolation, Fourier Transform, and Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) methods. The results showed that the CNN-LSTM method performed best, with an MAE value of 21.957. Additionally, the study compared CNN and LSTM separately, revealing that CNN was more accurate than LSTM. Based on these findings, CNN was chosen for this study because the data used is spatial, making CNN sufficient for building the model without the need for CNN-LSTM, which has a more complex architecture.

Silva-Ramírez et al. (2011) conducted a study comparing the Multiple Layer Perceptron (MLP) method with several classical methods, such as mean/mode imputation, regression models, and hot-deck imputation for missing data imputation. The results showed that MLP achieved higher accuracy than the other classical methods. Additionally, MLP was considered suitable for satellite data, making it a viable method for this study. Wang et al. (2006) conducted a study comparing the Support Vector Regression (SVR) method with K-Nearest Neighbor and Bayesian Principle Component methods for imputing missing data in gene expression data. The results showed that SVR achieved better accuracy than the other methods. Additionally, SVR was considered suitable for non-linear data due to its ability to utilize kernels, making it the chosen method for this study. Based on the previous studies, it is proposed to use the MLR, CNN, MLP, and SVR methods for imputing missing data, with satellite data as additional information due to its relatively more comprehensive coverage.

This study aims to determine the most suitable machine learning method and satellite data for filling in station data in East Java. Four machine learning methods are used: MLR, CNN, MLP, and SVR. Meanwhile, the satellite data includes ERA5, ERA5 Land, CMORPH CRT, CMORPH BLD, and CHIRPS. We limit the scope by performing imputation on rainfall data at stations in East Java, Indonesia. This research contributes to improving data quality by filling in incomplete data. Consequently, the data can be used to predict rainfall more accurately, benefiting various fields such as agriculture, water resource management, and disaster mitigation.

2. STUDY AREA AND DATASETS

2.1. East Java Province

This study focuses on the East Java region, which is geographically located between 5.37° – 8.48° South Latitude and 111.0° – 114.4° East Longitude, with an area of approximately 47.800 km^2. This region has a diverse topography, ranging from lowlands to mountains. East Java is known for its tropical climate, which has two main seasons: rainy and dry. This seasonal pattern is influenced

by the monsoon system, where the Asian monsoon brings moisture from the Indian Ocean, resulting in high rainfall during the rainy season, while the Australian monsoon brings dry winds during the dry season. A map of the East Java observation area is shown in **figure 1**.

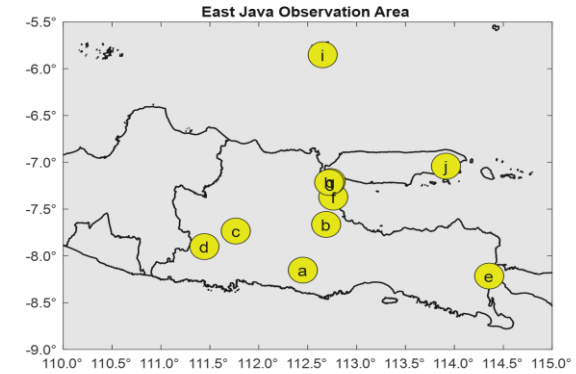


Fig. 1. East Java observation area.

The labels in **figure 1** are adjusted to match the station names in **Table 1**. The variations in climate and rainfall patterns in East Java significantly impact various sectors such as agriculture, water resources, and disaster mitigation. However, the rainfall data collected by the BMKG observation stations in East Java often need more data due to various causes, such as technical disruptions or human errors in the recording or processing stages. To address this issue, a data imputation process needs to be carried out by utilizing satellite data as an additional source of information. Satellite data provides a broader view of the atmospheric conditions in areas not covered by the observation station network. Therefore, the results of the analysis are more comprehensive and reliable.

2.2. Sources and Types of Datasets

The data used in this study are secondary in the form of daily rainfall (RR) data from January 2010 to December 2023. The data used are of two types: station data and satellite data.

2.2.1. Station Data

The station data used were obtained from the Badan Meteorologi, Klimatologi, and Geofisika (BMKG) websites of East Java. Rainfall is measured in millimetres (mm). Rainfall of 1 mm corresponds to 1 litre of rainwater falling on a surface area of 1 m². BMKG classifies rainfall based on its intensity as follows: light rain occurs when the daily rainfall is between 5-20 mm, moderate rain ranges from 20-50 mm/day, heavy rain occurs between 50-100 mm/day, and very heavy rain is classified when rainfall exceeds 1000 mm (BMKG, 2008). Of the 12 stations available, the researcher used data from 10 stations, as the other two were newly established and did not have rainfall data for 2010.

Table 1.

Missing data at BMKG stations in East Java.

Station	Station Name	Missing data
a	Geofisika Malang	34.18%
b	Geofisika Nganjuk	49.53%
c	Geofisika Pasuruan	31.59%
d	Klimatologi Jawa Timur	7.18%
e	Meteorologi Banyuwangi	34.04%
f	Meteorologi Juanda	11.15%
g	Meteorologi Maritim Tanjung Perak	10.44%
h	Meteorologi Perak 1	15.94%
i	Meteorologi Sangkapura	19.00%
j	Meteorologi Trunojoyo	15.32%

Table 1 shows data about ten weather stations in East Java that operate at different times. Each station has a different percentage of missing data. The East Java Climatology Station has the lowest percentage of missing data, which is 7.18%, while the Nganjuk Geophysical Station has the highest percentage of missing data, 49.53%. The average missing data across the ten BMKG stations in East Java is approximately 22.87%.

2.2.2 Satellite Data

This study uses five types of satellite data: ERA5, ERA5-Land, CHIRPS, CMORPH CRT, and CMORPH BLD. These five types of satellite data are stored in the Network Common Data File (NetCDF) format. The satellite data has four main variables: longitude, latitude, time, and total precipitation. The longitude variable indicates the longitude coordinates of the observation location, while the latitude variable covers the latitude position of the observation. The time variable represents the time of daily rainfall observation, and the total precipitation variable shows the total rainfall, with dimensions of $\text{longitude} \times \text{latitude} \times \text{time}$. The resolution, format, and source of each satellite data type can be seen in **Table 2.**

Table 2.

Specification of satellite data.

Types of Satellite Data	Spatial resolution	Temporal resolution	Format	Sumber
ERA5	$0{,}25° \times 0{,}25°$	*m/hourly*	NetCDF	*Website* ECMWF
ERA5 Land	$0{,}1° \times 0{,}1°$	*m/hourly*	NetCDF	*Website* ECMWF
CMORPH CRT	$0{,}1° \times 0{,}1°$	*mm/daily*	NetCDF	*Website* NOAA
CMORPH BLD	$0{,}25° \times 0{,}25°$	*mm/daily*	NetCDF	*Website* NOAA
CHIRPS	$0{,}05° \times 0{,}05°$	*mm/daily*	NetCDF	*Website* NOAA

The satellite data used in this study has varying spatial resolution. ERA5 and CMORPH BLD have the lowest spatial resolution, while ERA5 Land and CMORPH CRT offer higher resolution. On the other hand, CHIRPS has the highest spatial resolution, providing more explicit details for analysis. In addition to spatial resolution, the satellite data also has different temporal resolutions. The temporal resolution used in this study is set in mm/day. Therefore, ERA5 and ERA5 Land data, which initially have a temporal resolution of m/hourly, will be converted into mm/day to ensure consistency and ease of analysis. Satellite data is not used directly for imputing missing data. Instead, it is used as an input variable in the development of a machine learning-based model designed to estimate rainfall values at locations with missing data. This approach enables more accurate imputation by utilizing the more comprehensive information provided by satellite data. The developed model is then validated using complete station data, where a portion of the data is randomly removed to assess the accuracy of the model.

3. METHODS

3.1. Multiple Linear Regression (MLR)

MLR is one of the machine learning methods that can impute missing data. Tabony (1983) proposed that MLR fills in missing data using data stations from neighbouring locations as predictor variables in a regression equation. To apply the MLR method, at least two predictor variables must be used. Therefore, the following equation connects the amount of rainfall to be imputed at the station data (response variable) with N, representing the amount of rainfall in the satellite data (as predictor variables):

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_N x_N$$

where Y is the dependent variable; $\beta_1, \beta_2, \ldots, \beta_N$ are regression coefficient; β_0 intercept; and $x_1, x_2, \ldots, x_N$ are value rainfall. One method that can be applied to estimate regression parameters is Ordinary Least Square (OLS), which is given in the following equation.

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

where y_i represents the actual data value and $\hat{y}_i$ represents the predicted result (Han et al. 2023).

The performance of MLR deteriorates when multicollinearity or linear relationships among predictor variables increase (Vieira et al., 2020). This is one of the drawbacks of MLR using the OLS method. Therefore, linear regression models with regularization, such as ridge regression and the Least Absolute Shrinkage and Selection Operator (LASSO), have been developed. These models include penalty terms to address multicollinearity.

Ridge regression was first introduced by A.E. Hoerl in 1970 to address multicollinearity issues (Afham et al., 2017). Multicollinearity arises when two or more predictor variables are correlated. Ridge regression adds a penalty to the model to constrain the coefficient values of the linear regression model. The following equation represents Ridge regression.

$$\widehat{\boldsymbol{\beta}} = \underset{\widehat{\boldsymbol{\beta}}}{\operatorname{argmin}} (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_2^2$$

where Y is a dependent variable, X is independent variable, $\boldsymbol{\beta}$ are regression coefficient, λ is a shrinkage parameter whose value is always positive, and $\|\boldsymbol{\beta}\|_2^2$ is the ridge penalty, equal to $\sum_{j=1}^{k} {\beta_j}^2$ (Saleh et al., 2019). If $\lambda \to 0$, then the regression coefficient values will increase as in linear regression, but if $\lambda \to \infty$, the regression coefficient values will approach zero.

In addition to ridge regression, the LASSO method was first introduced by Tibshirani (1996), which addresses multicollinearity issues (Andana et al., 2017). LASSO addresses multicollinearity by shrinking regression coefficients to precisely zero, unlike ridge regression, which can only shrink coefficients close to zero. Therefore, LASSO is more suitable for high-dimensional data as it can reduce the number of predictor variables used. The following equation represents LASSO.

$$\widehat{\boldsymbol{\beta}} = \underset{\widehat{\boldsymbol{\beta}}}{\operatorname{argmin}} (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_1$$

where $\|\boldsymbol{\beta}\|_1$ is the LASSO penalty, which is equal to $\sum_{j=1}^{k} |\beta_j|$ (Saleh et al. 2019).

3.2. Convolutional Neural Network (CNN)

Compared to other networks with FC layers, the Convolutional Neural Network (CNN), a type of Artificial Neural Network (ANN), has a deep feed-forward architecture, a fantastic generalizing ability, and the ability to learn highly abstracted features of objects, particularly spatial data, and identify them more effectively. A deep CNN model comprises a limited number of processing layers capable of learning several levels of abstraction for different aspects of input data, such as images. High-level features are learned and extracted by the initiatory layers (with lesser abstraction), whereas low-level characteristics (with more abstraction) are learned and extracted by the deeper levels. The conceptual model of CNN is shown in **figure 2**.

A CNN model typically consists of two stages: feature learning and classification. Feature learning includes convolutional and sub-sampling layers, while classification consists of fully connected layers (Nurdiati et al., 2022). Feature learning is the first step in the CNN model and generates a feature map output. During this stage, the convolution layer process includes four key components: padding, stride, kernel, and activation function. Padding is a technique used to add rows or columns around the edges of the input image. Padding aims to preserve the data at the borders from being lost. Stride is the length of the shift step, and the longer the stride, the lower the density. Stride is used to control the density of convolution. The kernel in a 2D CNN is an $n \times n$ matrix used to perform the convolution operation on the input data.

After convolution, the feature map consists of several features susceptible to overfitting (Hawkins, 2004). Therefore, pooling layers (max-pooling and average pooling) are proposed to avoid redundancy (Fukushima, 1980). The pooling layer functions to reduce the amount of data by eliminating trivial features (Li et al., 2021). After the convolution and sub-sampling processes are performed, the final feature map is used as input data for the next stage, which is the fully connected layer.

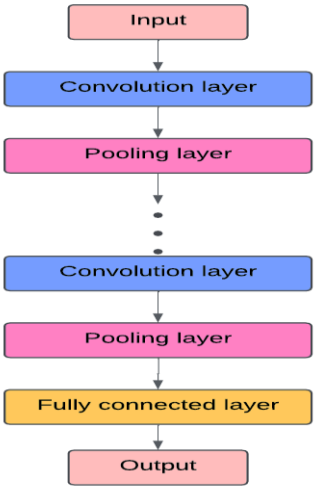


Fig. 2. Conceptual model of CNN (*Sultana et al., 2018)*.

3.3. Multiple Layer Perceptron (MLP)

MLP (Multilayer Perceptron) is an artificial neural network consisting of several layers of computational units connected sequentially in a feed-forward pattern (Bishop, 2006; Chiu, 2001; Mitchell, 1997). This structure enables MLP to process data hierarchically, where each hidden layer functions to extract increasingly complex features from the input data. MLP has been widely used in various applications, particularly for prediction problems, due to its flexibility in handling data with complex and non-linear relationships. This advantage is achieved through non-linear activation functions, which effectively help the network model the relationship between input and output, making it suitable for complex problems like imputing missing rainfall data.

Silva-Ramírez et al. (2011) used an MLP consisting of three processing layers. In the first layer, weights connect the input variables to H hidden units (neurons), while in the second layer, weights connect these hidden neurons to the output units. The hyperbolic tangent activation function is used in the hidden layer, and the identity function is used in the output layer. The following formula gives the hyperbolic tangent function.

$$g(u) = \frac{(e^{u} - e^{-u})}{(e^{u} + e^{-u})}$$

In this case, H represents the number of neurons in the hidden layer, while $v_{ih}, i = 0,1,2, \dots, p, h = 1,2, \dots, H$ are the synaptic weights that connect the input of size p to the hidden layer. Additionally, $w_{hj}, h = 1,2, \dots, H, j = 0,1,2, \dots, q$ are the synaptic weights that connect the hidden layer to the output layer of size q. The neural network output o_j for p input $x_1, \dots, x_p$ is formulated as follows.

$$o_j = w_{0j} + \sum_{h=1}^{H} w_{hj} g\left(v_{0h} + \sum_{i=1}^{p} v_{ih} x_i\right), j = 1,2, \dots, q.$$

In the context of missing data imputation, each categorical variable must be converted into a vector of dummy variables with values 0–1 for each class. As a result, the number of inputs p is generally more significant than the number of variables in the data file. The number of outputs in the MLP is adjusted to match the number of inputs. Thus, the network structure becomes (p, H, p). The imputation process for categorical variables is carried out by selecting the predicted dummy variable with the highest value as the category representing the imputed result. The learning process in the network is performed by adjusting the synaptic weights using a supervised learning approach. Training data in input-output pairs is repeatedly fed to the network to allow weight adjustments to improve the alignment between the network's predicted results and output values.

3.4. Support Vector Regression (SVR)

SVR is an effective machine learning method for missing data imputation. The SVR method approach provides better results than SVM in missing data imputation cases. Honghai et al. (2005) used SVR to perform missing data imputation. The research procedure involved setting the response variable as the predictor variable and the predictor variable as the response variable, and then SVR predicted the value of the predictor variable.

SVR uses regularization techniques to control the complexity of the model and avoid overfitting.

$$\varphi(\boldsymbol{w}, b, \xi) = \frac{1}{2}(\boldsymbol{w}\boldsymbol{w}') + C \sum_{i \in SV} \xi_i$$

C the regularization parameter controls the trade-off between the maximum margin and prediction error (Brereton and Lloyd, 2009). A significant value of C means the model will try to classify all training data correctly, leading to overfitting. In contrast, a small value of C will result in a more significant margin, allowing some classification errors, which can help prevent overfitting.

SVR is suitable for cases with non-linear data because it utilizes the principle of working in a high-dimensional space using kernels to model the complex relationship between available data and missing values. SVR has several popular kernels, such as the polynomial kernel and the Radial Basis Function (RBF) kernel. The polynomial kernel is a function used to measure the similarity between two vectors in feature space. This kernel function can expand the model's capacity by mapping input data to a higher-dimensional space. The polynomial kernel function is defined as:

$$\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{z}) = (1 + \boldsymbol{x}^T \boldsymbol{z})^d$$

where $x\ and\ z$ are vector input, 1 is constanta which controls the bias, and d is the degree of the polynomial. We can capture more complex data features using the polynomial kernel than a simple linear model cannot. The higher the degree d, the more complex the resulting model becomes.

The RBF kernel, also known as the Gaussian kernel, measures the similarity between two points in a way that is sensitive to the distance between those points. The RBF kernel function is defined as:

$$\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{z}) = e^{\frac{-(\boldsymbol{x}, \boldsymbol{z})^2}{\sigma^2}}$$

where σ is a parameter that controls the width of the Gaussian function. The RBF kernel maps the input data to an infinite-dimensional feature space, allowing it to capture complex patterns. The smaller the value of σ, the more sensitive the kernel is to changes in the distance between data points, which affects how the model learns from the data

3.5. Optimizer

An optimizer is an algorithm or method used to iteratively adjust the model's bias and weights during training to minimize the error function. Optimizers are crucial for improving the model's performance and accelerating the convergence process. Parameter updates during training are based on the learning rate, which determines the step size taken when updating parameters. Meanwhile, a training epoch refers to one complete cycle in which the entire training data is used. Since the learning rate is a very important hyperparameter, its value must be selected carefully to ensure that the learning process runs optimally without issues. This research uses three top optimizers based on the study by (Nurdiati et al., 2022): Adaptive Moment Estimation (Adam), Nesterov Adaptive Moment Estimation (NAdam), and Adaptive Moment Estimation with Weight Decay (AdamW). Adam (Kingma, 2014) is a popular optimization algorithm in deep learning model training. This algorithm combines two main techniques, Momentum and RMSProp, enabling faster convergence and improving training stability. Adam uses the exponential moving averages of the first and second gradients to adjust the learning rate for each model parameter adaptively. Before calculating the parameter updates for Adam, NAdam, and AdamW, the first step is to compute the gradients, the first-moment estimate (mean), and the second-moment estimate (variance) sequentially using the given equations.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t,$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2,$$

$$\widehat{m_t} = \frac{m_t}{(1-\beta_1^t)}, \widehat{v_t} = \frac{v_t}{(1-\beta_2^t)}$$

where β_1 and β_2 is the exponential decay rate. The recommended default values for β_1 is 0.9, β_2 is 0.999 and ϵ is 10^{-8}. NAdam (Dozat, 2016) is a modified algorithm of the Adam optimizer that combines the advantages of Adam with Nesterov Momentum to improve the speed and stability of convergence. Using Nesterov momentum, NAdam considers the gradient at the anticipated position, resulting in a more optimal direction for parameter updates. Like Adam, NAdam also uses the first (momentum) and second (RMSProp) gradient averages to adjust the learning rate for each model parameter adaptively. This algorithm is suitable for various deep-learning applications because it can optimize the loss function more efficiently. AdamW is another modified algorithm of the Adam optimizer that addresses regularization issues in deep learning models, particularly related to weight decay. Unlike Adam, which integrates ℓ_2 regularization after parameter updates, AdamW applies weight decay directly to the parameters before the update is performed, resulting in more consistent parameter updates and more effectively reducing overfitting. By separating the regularization process from the gradient-based update, AdamW provides better control over regularization. The updating formula of Adam, NAdam, and AdamW is given in **Table 3**.

Table 3.

Updating the formula of Optimizer.

Optimizer	Updating formula
Adam	$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\widehat{v_t}} + \epsilon}\widehat{m_t}$
NAdam	$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\widehat{v_t}} + \epsilon}\left(\beta_1\widehat{m_t} + \frac{1-\beta_1}{1-\beta_1^t}g_t\right)$
AdamW	$\theta_{t+1} = \theta_t - \zeta_t\left(\frac{\eta}{\sqrt{\widehat{v_t}} + \epsilon}\widehat{m_t} + \lambda\theta_t\right)$

where λ is the weight decay value, AdamW uses a scaling factor ζ_t to adjust the scheduling of the learning rate η and weight decay λ, which can be controlled through the SetScheduleMultiplier(t) procedure (Loshchilov & Hutter, 2019). At each iteration, the value of ζ_t gradually decreases following the cosine annealing method, where the learning rate gradually decreases with each batch during the training process (Loshchilov & Hutter, 2016)

3.6. Metric Evaluate

To evaluate the accuracy of the imputation model used in this study, the Mean Absolute Error (MAE) metric is applied as the primary measure. This study uses only the MAE metric to evaluate the model's accuracy because it aligns with the loss function used, which is MAE. MAE is a straightforward metric because it measures the average of the absolute differences between the predictions and the actual values, making it easy to interpret. Additionally, MAE is robust to outliers. MAE is a measure that describes the variability of estimation errors. A more minor error variance indicates that the model used for estimation is performing well. MAE can be calculated using the formula provided by (Junninen et al., 2004).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|O_i - P_i|$$

where n is data, O_i is the original or actual value, and P_i is the predicted value.

4. RESULTS AND DISCUSSION

4.1. Pre-processing of Datasets

Data pre-processing cleans the collected data (Hicham et al., 2020). Data pre-processing is the leading and most critical step in the Knowledge Discovery in Databases (KDD) process (Benhar et al., 2020; Gunadi, 2022). This technique is necessary to remove noise from raw data in order to extract

meaningful information from it. Data pre-processing aims to obtain data ready to be used in the data processing stage.

4.1.1. Station Data

All the station data that has been downloaded is combined into a single tabular file to facilitate data analysis. Based on the missing data for each station, the average missing data is approximately 22.87%. The data is then pre-processed using Matlab software to convert the index "8888," which indicates no measurement was taken, and to change empty cells into the Not a Number (NaN) format.

4.1.2. Satellite Data

The satellite data is pre-processed using Matlab software to adjust the variables of longitude, latitude, time, and total precipitation (tp). The ERA5 and ERA5 Land data have tp units in m/hourly, so they must be converted to mm/day for data consistency. The size of the satellite data matrix after pre-processing is summarized in **Table 4**.

Table 4.

The size of the satellite data matrix after pre-processing.

No.	Jenis Data Satelit	Longitude	Latitude	Time(day)	TP (mm)
1	ERA5	17×1	13×1	5113×1	$17 \times 13 \times 5113$
2	ERA5 Land	46×1	30×1	5113×1	$46 \times 30 \times 5113$
3	CHIRPS	71×1	66×1	5113×1	$71 \times 66 \times 5113$
4	CMORPH CRT	49×1	40×1	5113×1	$49 \times 40 \times 5113$
5	CMORPH BLD	19×1	15×1	5113×1	$19 \times 15 \times 5113$

Table 4 shows the matrix size of each type of satellite data used in this study. The matrix dimensions consist of longitude, latitude, and time coordinates. The longitude and latitude coordinates reflect the observation area, while the time dimension covers the number of days from January 1, 2010, to December 31, 2023. ERA5 has the smallest matrix size compared to the other satellite data due to its relatively low spatial resolution. In contrast, CHIRPS has the largest matrix size due to its highest spatial resolution among the other datasets. Each type of satellite data covers different observation areas. ERA5-Land, CHIRPS, and CMORPH BLD are specifically designed to measure precipitation over land areas. On the other hand, ERA5 and CMORPH CRT cover broader regions and can measure precipitation over land and oceans. A visualization of the observation areas for each satellite dataset is shown in **figure 3**.

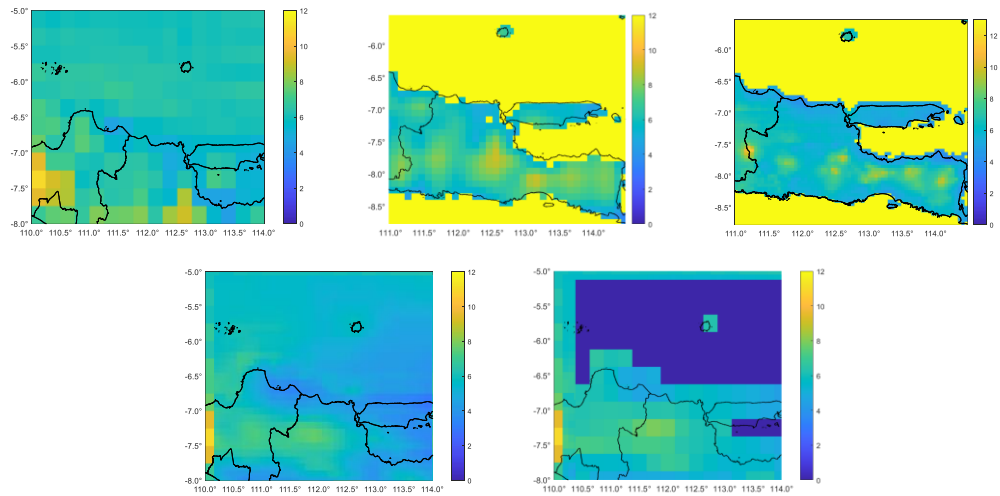


Fig. 3. Visualization of satellite data. (a) ERA5. (b) ERA5 Land. (c) CHIRPS. (d) CMORPH CRT. (e) CMORPH BLD.

Figure 3 shows the visualization of satellite data from various sources after the pre-processing stage. All data have been adjusted to cover the East Java region. The blue colour indicates low rainfall. Meanwhile, the yellow colour represents the highest rainfall on the map. Each data source has a different spatial resolution, affecting the detail level in the precipitation visualization. ERA5 and CMORPH BLD have the lowest spatial resolution, $(0.25° \times 0.25°)$ providing the coarsest visualization detail. CMORPH CRT and ERA5 Land have higher spatial resolutions $(0.1° \times 0.1°)$, resulting in smoother and more detailed visualizations. CHIRPS has the highest spatial resolution, $(0.05° \times 0.05°)$, providing the finest detail in the visualization compared to the other datasets. All satellite data have undergone pre-processing and are now ready for the next phase.

4.2. Construction Model

4.2.1. Setup Hyperparameter

Before training model, setup hyperparameters are configured for model construction, including selecting the optimizer and learning rate. This step aims to identify the parameters that provide the best performance before applying the model more broadly. At this stage, the three best optimizers used, as suggested by Nurdiati et al. (2022), are Adam, NAdam, and AdamW. Additionally, three learning rate values are tested: 0.1, 0.01, and 0.001. The training process was conducted for up to 300 epochs (ε) at the Geofisika Station in Malang. The experimental results are presented in **Table 5**.

Table 5.

Optimizer and learning rate testing.

Optimizer - Learning rate	Loss train	Loss test	Epoch
Adam - 0.001	7.661	7.555	127
Adam - 0.01	7.641	7.594	40
Adam - 0.1	7.717	7.306	24
NAdam - 0.001	7.605	7.473	283
NAdam - 0.01	7.656	7.337	63
NAdam - 0.1	7.762	7.322	28
AdamW - 0.001	7.967	7.673	70
AdamW - 0.01	7.929	7.631	30
AdamW - 0.1	7.944	7.245	13

Table 5 presents the results of testing the combination of three optimizers with three different learning rates. At a learning rate 0.001, NAdam performed better than the other optimizers, with the lowest train loss and test loss. This indicates that NAdam can update weights stably and efficiently at a small learning rate. At a learning rate of 0.01, Adam achieved a lower train loss. However, NAdam provided a minor test loss, indicating that the model trained with NAdam had better generalization ability on unseen data.

On the other hand, at a learning rate of 0.1, Adam performed the best, with the lowest train and test loss values, suggesting that at a more significant learning rate, Adam could learn data patterns quickly and efficiently. Therefore, this combination is selected to train models at other stations as it balances accuracy, learning stability, and generalization ability. The results of this trial will be applied to the four machine learning methods.

The constructed model is tailored to the machine learning methods and the size of each satellite dataset. This study employs four machine learning methods: Multiple Linear Regression (MLR), Convolutional Neural Network (CNN), Multilayer Perceptron (MLP), and Support Vector Regression (SVR), along with five types of satellite data: ERA5, ERA5 Land, CMORPH CRT, CMORPH BLD, and CHIRPS. As a result, 20 models were built using Julia programming language.

4.2.2 Multiple Linear Regression (MLR)

MLR is one of the machine learning methods used for missing data imputation. Before constructing MLR model, an important step that must be carried out is setup hyperparameter as shown in **Table 6**.

Table 6.

Hyperparameter for MLR model.

Hyperparameter	Value	Information	Reference
Regularization	L1 (LASSO), L2 (Ridge)	Prevents overfitting. L1 encourages sparsity; L2 shrinks coefficients	LASSO: (Tibshirani, 1996) Ridge: (Hoerl, 1962)
Optimizer	Adam, Nadam, AdamW	By Experiment	Nurdiati et al., 2022
Learning Rate	0.1, 0.01, 0.001	By Experiment	Nurdiati et al., 2022
Maximum Epoch	1000	By Experiment	Nasution & Andayani, 2017
Activation Function	Softplus	By Experiment	Rasywir et al., 2022

After setup hyperparameter, the next step is to build the MLR model. The first step in construct MLR model is inputing data in the form of a matrix size from satellite data (independent variables), as shown in **figure 4**. The second step is flatten matrix 3D into a vector 1D. After flattening, the number of neurons in the dense layer matches the total dimensions of the input data. The number of neurons in the dense layer varies depending on the grid size of the satellite data. ERA5 uses 221 neurons, ERA5 Land requires 1380 neurons, CMORPH CRT involves 1960 neurons, CMORPH BLD uses 285 neurons, and CHIRPS employs 4686 neurons for the largest dimension.

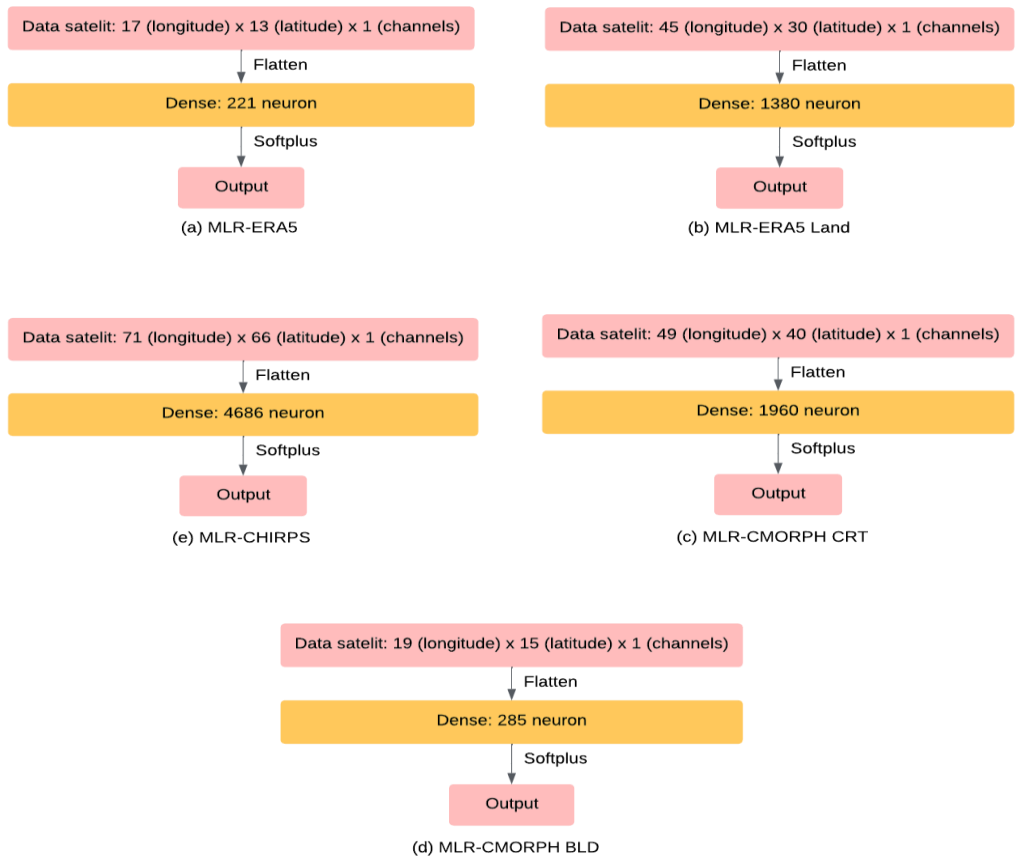


Fig. 4. Architecture of MLR model.

Subsequently, a softplus activation function is applied in the dense layer to ensure the output values are positive. The model is designed to accommodate the unique characteristics of each type of satellite data with varying spatial resolutions. This figure provides a clear visual representation of the differences in size and architecture needed to process data from each satellite type. Last step is predicted missing rainfall data used satellite data.

4.2.3. Convolutional Neural Network (CNN)

The CNN model uses a modified LeNet architecture (LeCun, 1998). Modifications to the LeNet architecture were made to adapt the model to the characteristics of rainfall and satellite data. LeNet was chosen for its simple architecture, which is suitable for the relatively small size of the satellite data. LeNet is effective in capturing spatial patterns in satellite rainfall data. Before construct CNN model, the hyperparameters are set first as shown in **Table 7**.

Table 7.

Hyperparameter for CNN model.

Hyperparameter	Value	Information	Reference
Convolution layers	4 layers	By LeNet Architecture	LeCun, 1998
Fully connected layers	3 layers	By LeNet Architecture	LeCun, 1998
Activation Function	ReLu, Sigmoid, Softplus	By Experiment	Rasywir et al., 2022
Pooling Type	Max Pooling	By Experiment	Alwanda et al., 2020
Optimizer	Adam, Nadam, AdamW	By Experiment	Nurdiati et al., 2022
Learning Rate	0.1, 0.01, 0.001	By Experiment	Nurdiati et al., 2022
Maximum Epoch	1000	By Experiment	Nasution & Andayani,2017

After setup hyperparameter, the next step is to build CNN model. The CNN model is presented in **figure 5**. The architecture consists of four convolutional layers that are responsible for extracting features from satellite image data with spatial information. Each layer uses different kernel operations to capture various features from the satellite data. The activation function used is ReLU (Rectified Linear Unit). ReLU is the most commonly used activation function in CNNs (Nair and Hinton, 2010). ReLU is used to transform all input values into positive numbers using the following mathematical equation.

$$f(x)_{ReLU} = \max(0, x).$$

After the convolutional layers, the processed data has a 3D matrix size. The flattening process converts the data into a 1D vector so that it can be used by the fully connected layers. Each neuron in a layer is connected to every neuron in the previous layer, making the network highly effective in making predictions. In the fully connected layers, the activation functions used are sigmoid and softplus. Sigmoid takes real numbers as input and bounds the output within the interval [0,1]. Small values of x approach 0, while large values of x approach 1. The mathematical representation of sigmoid is:

$$f(x)_{sigm} = \frac{1}{1 + e^{-x}}$$

Softplus is a smoother version of ReLU that always produces positive values, making it suitable for predicting rainfall values. The mathematical equation for softplus can be seen in the equation below.

$$f(x)_{softplus} = ln(1 + e^{x})$$

where x is input neuron and e is exponential number.

Dropout of 0.5 is applied to several layers to prevent overfitting. Dropout is a regularization technique that works by randomly ignoring some neurons during training. This forces the model not to rely on specific neurons and helps improve the model's generalization ability. The main differences in the figure lie in the kernel size, padding, and stride used in each layer. All these parameters are adjusted based on the size of the satellite data used.

4.2.4. Multiple Layer Perceptron (MLP)

MLP is an artificial neural network architecture that models non-linear input and output data relationships. The network structure is designed by determining the number of hidden layers and the number of neurons per layer. Before construct CNN model, the hyperparameters are set first as shown in **Table 8**.

After setup hyperparameter, the next step is to build CNN model. Each type of satellite data has a different number of neurons, as shown in **figure 6**.

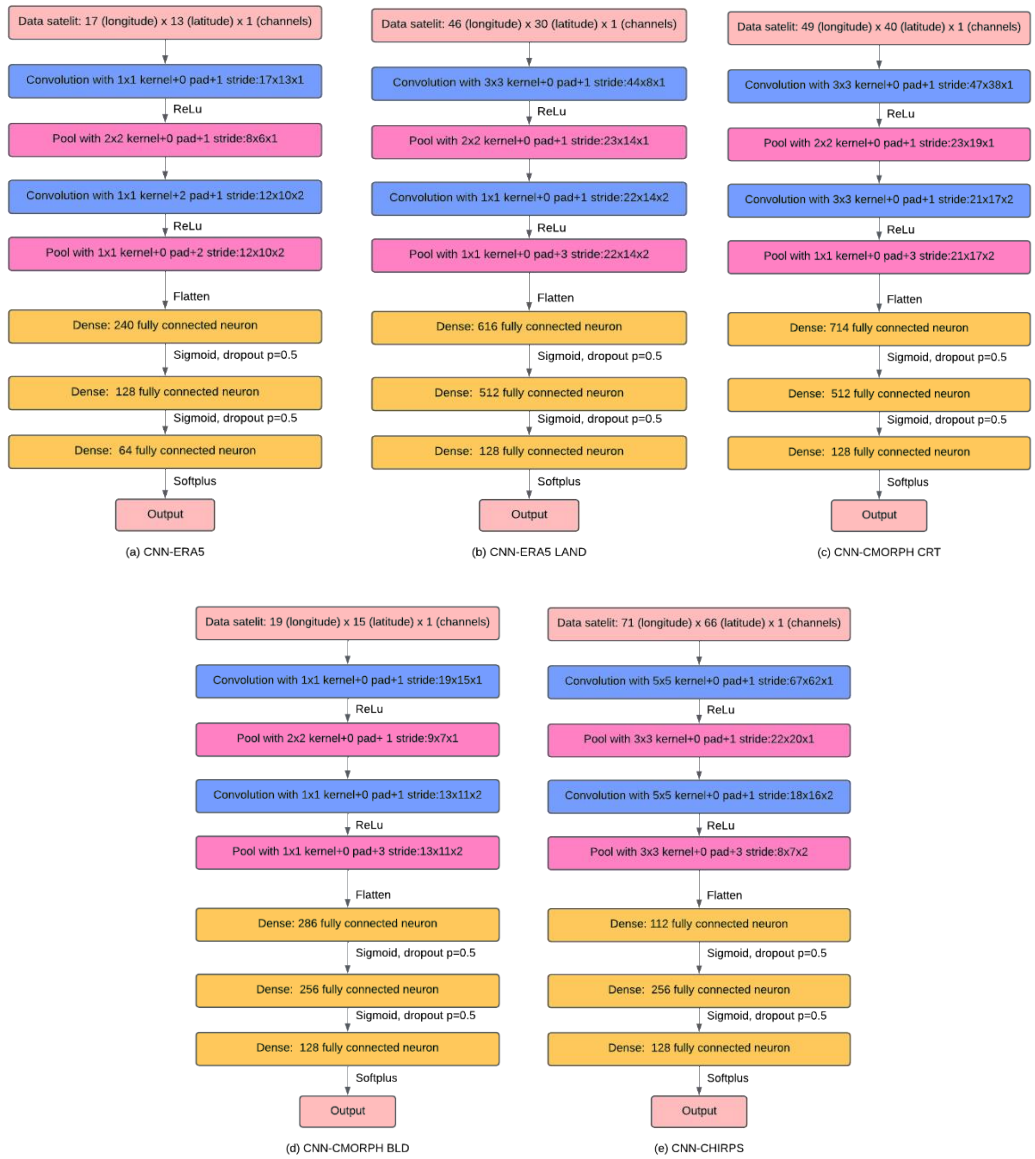


Fig. 5. Modified LeNet architecture.

Table 8.

Hyperparameter for MLP model.

Hyperparameter	Value	Information	Reference
Hidden layers	3 layers	By Literature	Silva-Ramírez et al., 2011
Activation Function	Sigmoid, Softplus	By Experiment	Rasywir et al., 2022
Optimizer	Adam, Nadam, AdamW	By Experiment	Nurdiati et al., 2022
Learning Rate	0.1, 0.01, 0.001	By Experiment	Nurdiati et al., 2022
Maximum Epoch	1000	By Experiment	Nasution & Andayani, 2017

Figure 6 illustrates the MLP architecture designed to process satellite data to extract spatial features from satellite images. This architecture uses three dense (fully connected) layers to learn complex relationships between satellite data as input and station data as output. The satellite data

varies in size and is flattened into a one-dimensional (1D) vector. This process aims to convert the image data into a format that the dense layers can process. The first dense layer is the result of flattening the satellite data. The neurons in this layer serve as the initial representation of the satellite data, where basic patterns from the image data begin to be identified. The second and third dense layers are designed with varying numbers of neurons, adjusted to the characteristics of each satellite dataset. These layers are responsible for learning more profound and complex relationships in the data, with the second layer typically having more neurons than the third to capture richer feature patterns.

Each dense layer uses the sigmoid activation function, which allows the model to learn non-linear relationships between the data features. Then, a dropout with a value of 0.5 is applied to prevent overfitting by randomly removing some neurons during training, helping the model become more resilient to relying on specific patterns in the training data. The softplus activation function follows the final dense layer, which is suitable for prediction tasks requiring positive results.

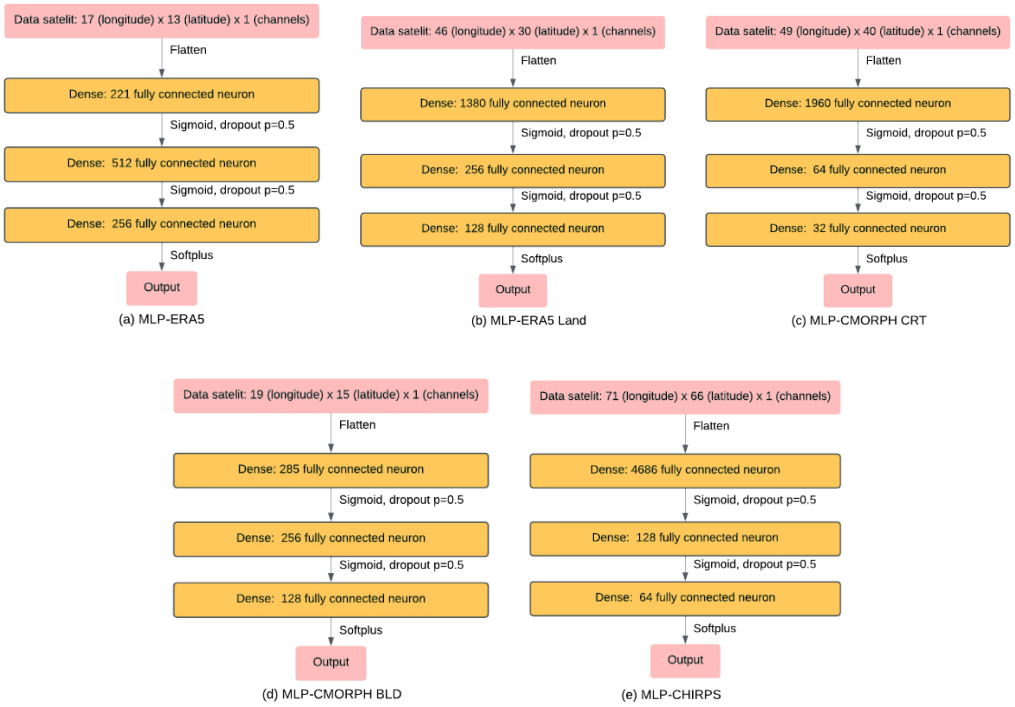


Fig. 6. MLP model architecture.

4.2.5. Support Vector Regression (SVR)

SVR is a kernel-based machine learning method that models the relationship between input and output variables while considering a specific margin of error (ϵ). Unlike other methods such as MLR, CNN, or MLP, SVR employs a unique approach using a kernel function to map data into higher-dimensional space. This study uses the Radial Basis Function (RBF) kernel to build the SVR model. The RBF kernel was chosen for its ability to capture complex non-linear patterns, enabling data not linearly separable in the original space to be separated in a higher-dimensional space.

SVR optimizes the cost and gamma parameters to enhance the model's performance for each dataset. Referring to the study by He (2018), which tested various combinations to determine the optimal cost and gamma, the search range for parameters was narrowed to 2^{-6} for cost and 0.001-0.01 for gamma. This study tested combinations of cost and gamma using this approach, with each test repeated 10 times to ensure consistency of results and avoid computational bias. The trials were conducted at the Malang Geophysical Station, with results presented in **Table 10**.

Table 9.

Hyperparameter for SVR model.

Hyperparameter	Value	Information	Reference
Kernel	RBF	By Literature	Smola & Schölkopf, 2004
Cost	2.0, 3.0, 4.0, 5.0, 6.0	By Experiment	He, 2018
Gamma	0.001, 0.01	By Experiment	He, 2018
Epsilon	0.001	By LIBSVM package	-
Optimizer	Adam, Nadam, AdamW	By Experiment	Nurdiati et al., 2022
Learning Rate	0.1, 0.01, 0.001	By Experiment	Nurdiati et al., 2022
Maximum Epoch	1000	By Experiment	Nasution & Andayani, 2017

Table 10.

Experiment to determine the best cost and gamma parameters. The yellow colour indicates the best combination.

cost	gamma	Loss train	Loss test
2.0	0.001	7.963	7.835
3.0	0.001	7.947	7.815
4.0	0.001	7.942	7.762
5.0	0.001	7.982	7.485
6.0	0.001	8.119	7.648
2.0	0.01	8.018	7.799
3.0	0.01	8.042	7.814
4.0	0.01	8.093	7.777
5.0	0.01	8.166	7.761
6.0	0.01	8.266	7.766

Based on the cost and gamma parameter combination test results, **Table 8** shows the average values of the loss train and loss test for each parameter pair. The best performance is observed at a gamma value of 0.001 when the cost is 5.0, with a loss train of 7.982 and a loss test of 7.485. The lower loss test value compared to other parameter pairs indicates that the model achieves the best generalization for the test data. When the cost increases to 6.0, loss train and loss test tend to grow. At gamma 0.01, loss train and loss test tend to rise with increasing cost. This indicates that increasing the cost value at this gamma does not positively impact the model's performance. The gamma value of 0.001 results in a lower loss test compared to gamma 0.01 at the same cost combination. This suggests that gamma 0.001 is more suitable for the dataset used in this study. The combination of cost 5.0 and gamma 0.001 results in the best accuracy and thus will be used to train the model on all data.

The SVR model is built using the Julia programming language with the LIBSVM library. This model uses Epsilon-SVR, a regression method based on a Support Vector Machine (SVM) aimed at predicting continuous values while allowing for a specific deviation (epsilon) between predicted and actual values. The cost parameter controls the penalty level for prediction errors in the model. This study uses a cost value of 5.0. Gamma determines the influence of a data point. A small gamma value (such as 0.001) makes the model more general, while a considerable gamma value makes the model more focused on specific data points, which can lead to overfitting. The Radial Basis Function (RBF) kernel is used because it is highly effective in mapping data from the input space to a higher-dimensional space, allowing the model to capture non-linear patterns effectively.

4.3. Training and Evaluate Model

This section discusses model training, which aims to optimize parameters so that the model can effectively recognize patterns from the training data, while model testing is used to evaluate the accuracy of the developed model using the MAE metric. In this process, the data used to train the model comes from complete satellite data. The evaluation is conducted using complete station data as the primary reference. To simulate missing data conditions, a portion of the station data is deliberately removed at random. The modified data is then used as input for the model, and the predicted results are compared with the original values from the station data that were not removed. The training process is conducted with various data split ratios of 95:5%, 90:10%, 80:20%, 70:30%, and 50:50%. Additionally, the training process is limited to a maximum of 1000 epochs (ε).To ensure

consistent results and avoid computational bias, the testing procedure is repeated 10 times. The results are presented in **Table 11**.

The CMORPH CRT dataset has the lowest training loss across all scenarios compared to other datasets. However, this dataset also consistently experiences overfitting in all data-splitting scenarios, indicated by higher test loss than training loss. This condition suggests that although training performance appears very good, the model's ability to generalize to test data is still limited. On the other hand, the CMORPH BLD, CHIRPS, and ERA5 Land datasets do not show signs of overfitting in any data-splitting scenarios. However, the training loss and test loss for these three datasets tend to be higher than others. This indicates that while the model can avoid overfitting, its overall performance is still suboptimal in training and prediction.

Table 11.

The training model uses MLP. The grey colour indicates overfitting.

Data	95%		90%		80%		70%		50%	
	MAE train	MAE test	MAE train	MAE test	MAE train	MAE test	MAE train	MAE test	MAE train	MAE test
ERA5	6.996	6.588	6.975	6.874	7.084	7.124	7.013	7.231	6.947	7.304
ERA5 LAND	8.000	7.711	7.9985	7.813	8.0123	7.837	8.003	7.913	8.009	7.943
CMORPH CRT	6.922	7.139	6.822	7.338	6.787	7.370	6.773	7.402	6.671	7.493
CMORPH BLD	7.996	7.732	7.995	6.636	7.949	7.876	8.012	7.892	7.987	7.965
CHIRPS	8.003	7.673	8.000	7.783	8.009	7.834	8.000	7.920	8.107	7.861

The ERA5 dataset shows varying performance across data-splitting scenarios. In the 95:5% scenario, this dataset provides the most optimal results with relatively low and balanced training and test loss. However, in scenarios with a more significant proportion of test data, such as 80:20%, 70:30%, and 50:50%, the model experiences overfitting. Overall, although the MLP method is more resistant to overfitting than previous methods, the performance of each dataset still depends on its characteristics, affecting the model's ability to learn patterns and make predictions.

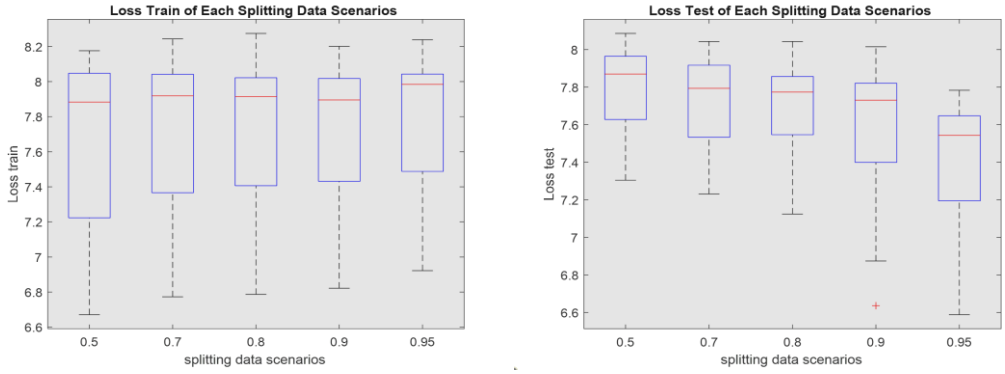
4.4. Performance Loss Training and Testing

This section explains the performance of loss training and testing, which is visualized using a boxplot. The choice of boxplot as a visualization method is based on its ability to provide a more comprehensive interpretation, such as displaying the median, minimum value, and maximum value, and detecting outliers that may affect model performance. Additionally, boxplots facilitate comparative analysis by presenting the loss distribution of each model side by side, allowing for a clearer evaluation of the variation and stability of model performance. The loss performance is assessed based on three aspects: data splits, machine learning methods, and satellite data.

4.4.1. Loss Performance Across Data Splits

This section examines the performance of loss during the training and testing processes under various data split scenarios. The data is divided into two parts: training and testing data, with five different splitting scenarios: 95:5%, 90:10%, 80:20%, 70:30%, and 50:50%. The loss values for training (training loss) and testing (testing loss) are calculated for each scenario, and the results are presented in boxplots shown in **Figure 7(a)** for the training loss values and **Figure 7(b)** for the testing loss values. The median value for the training data (train loss) tends to remain stable at around 7.8 across all data split scenarios. This indicates that the model's performance in learning the training data patterns is relatively unaffected by variations in the data divided proportions. The distribution of loss values in the training data also appears consistent across scenarios, without any noticeable outliers, as shown in **Figure 7(a)**. This stability suggests that the model can perform training effectively across different data configurations. In contrast, for the testing data (test loss), outliers were observed in the 90:10% data split scenario, as shown in **Figure 7(b)**. These outliers indicate that in some tests, the model was able to produce a test loss much lower than the average or median value. The median test

loss tends to be slightly higher in scenarios with a more significant proportion of test data, such as the 50:50% split. This suggests that a more significant proportion of test data may impact the model's performance, although the distribution of test loss remains narrower compared to the training data. Despite some variations, the narrower distribution indicates the model's performance consistency on the test data.



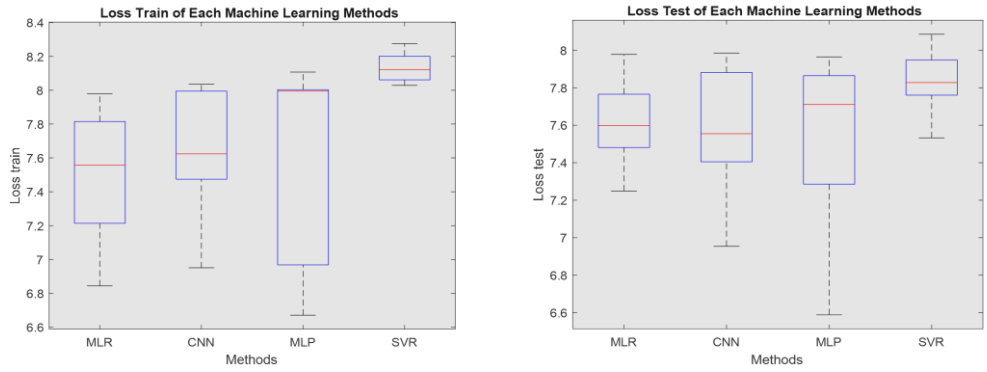
(a) (b)

Fig. 7. Loss performance across data splits. (a) loss train; (b) loss test.

Scenarios with a more significant proportion of training data, such as the 95:5% split, result in better and more consistent performance. This is evident from the slightly higher train loss than the test loss, indicating that more training data helps the model learn the patterns more deeply while improving its generalization ability. This condition reduces the risk of overfitting and ensures that the model performs well on unseen data. In contrast, scenarios with a more significant proportion of test data, such as the 50:50% split, tend to result in less stable performance. The more minor training data in this scenario can limit the model's ability to learn the patterns optimally, ultimately increasing the risk of overfitting or the model's inability to generalize effectively. Therefore, a more significant proportion of training data, such as in the 95:5% scenario, is recommended to balance training and testing performance.

4.4.2. Loss Performance by Machine Learning Methods

This section examines loss performance during the training and testing processes using four different machine learning methods. The model is trained and tested using Multiple Linear Regression (MLR), Convolutional Neural Network (CNN), Multiple Layer Perceptron (MLP), and Support Vector Regression (SVR). The loss values for training (training loss) and testing (testing loss) are calculated for each method, and the results are presented in the form of boxplots displayed in **figure 8(a)** for training loss and **figure 8(b)** for testing loss.



(a) (b)

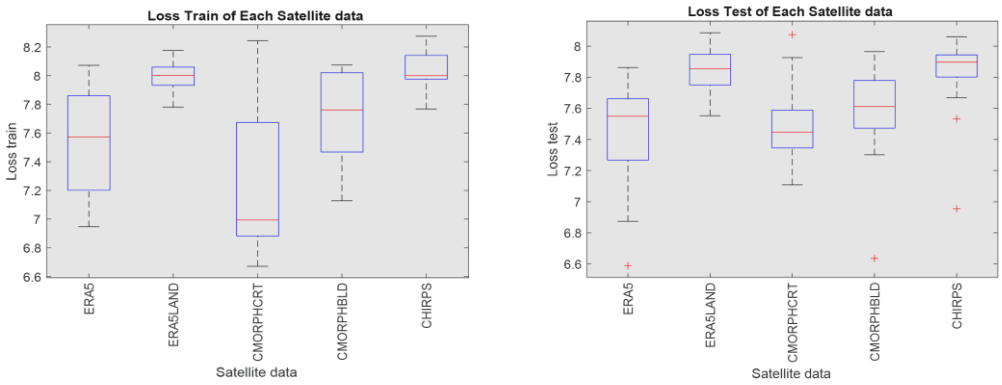
Fig. 8. Loss performance by machine learning methods. (a) loss train; (b) loss test.

Based on the analysis of the distribution of loss values and the median values, the MLR and CNN methods show the best distribution. However, upon further examination, both methods have higher loss values on the test data than the training data, indicating potential overfitting. This overfitting suggests that the model is too focused on patterns in the training data, making it less capable of making accurate predictions on new data. On the other hand, the SVR method has the highest median loss, both on the training and testing data. This indicates that SVR struggles to learn data patterns. However, this method is consistent, as shown by the narrowest loss distribution. This consistency indicates that SVR's performance is stable across different conditions, although it is less optimal overall. The MLP method has the widest loss distribution, especially on the training data. The high variability in training performance suggests that the MLP model is sensitive to certain training conditions. Additionally, outliers in the MLP loss distribution indicate that in some scenarios, the model can achieve loss values much lower than the median. This suggests that MLP has the potential to perform very well under certain conditions, although it generally shows considerable variability.

CNN performs very well, with a low median loss on the training and testing data. This indicates that CNN can learn data patterns effectively without losing generalization. However, when compared further, the MLP method remains the top choice because it has the lowest overall loss value. Therefore, MLP should be considered for further analysis, especially if it can be optimized to minimize performance variability and address potential outliers.

4.4.3. Loss Performance Based on Satellite Data

This section examines the loss performance during the training and testing processes using five different types of satellite data: ERA5, ERA5 Land, CMORPH CRT, CMORPH BLD, and CHIRPS. The model is trained and tested using each type of satellite data. The training loss and testing loss are calculated for each type of satellite data, and the results are presented in the form of boxplots shown in **figure 9(a)** for the training loss and **figure 9(b)** for the testing loss.



(a) (b)

Fig. 9. Loss performance based on satellite data. (a) loss train; (b) loss test.

The performance analysis of various datasets shows that the CMORPH CRT dataset has a lower median value than other satellite data. This indicates that the model can learn the data patterns from CMORPH CRT well during training. However, the high variability in the loss suggests inconsistent data quality. Additionally, the higher testing loss compared to the training loss indicates a risk of overfitting. This overfitting occurs because the model is too focused on the training data, making it less capable of generalizing to the testing data. On the other hand, the ERA5 Land and CHIRPS datasets show relatively higher median values but with narrower distributions. This distribution reflects a more consistent model performance, although their absolute performance is not as good as CMORPH CRT. This stability suggests that these datasets have more uniform data quality, supporting the model in learning data patterns more evenly. The ERA5 and CMORPH BLD datasets have higher median values than CMORPH CRT but remain relatively low overall. These two datasets also have

narrower distributions, indicating more stable training performance. This stability suggests that data from ERA5 and CMORPH BLD strongly support learning data patterns without excessive fluctuations. **Figure 9(a)**, no significant outliers are visible in the loss distribution. However, in **figure 9(b)**, several outliers can be seen in the ERA5, CMORPH CRT, and CHIRPS data. The presence of these outliers indicates that the model may produce loss values that differ significantly from the median under certain conditions. This is particularly evident in the CMORPH CRT dataset, which, despite providing low training loss values, shows a more significant generalization gap. This further strengthens the indication of overfitting risk. The ERA5 and CMORPH BLD datasets demonstrate the most consistent performance in training and testing data. This is reflected in their low median values and narrow distribution variations. With high-performance stability, these two datasets are more reliable for rainfall imputation modelling. Conversely, while CMORPH CRT offers a low median value, its high risk of overfitting and variability make it less ideal for use without further processing.

4.5. Discussion

This study provides valuable insights, but several limitations should be considered. First, the study only utilizes data from BMKG stations in East Java, meaning its generalization to other regions requires further investigation. Second, while MLP shows the best results in this study, further optimization of the model architecture and hyperparameters could enhance its performance. Additionally, this study has not explored the impact of more complex temporal variability, such as seasonal effects and extreme climate events. For future research, it is recommended to expand the study area and test hybrid methods that combine multiple machine learning models. With these improvements, the accuracy and reliability of rainfall imputation models can be further enhanced.

4.5.1. Imputation Missing Data

This section explains how the results of missing rainfall data imputation are visualized to compare the original values with the values after imputation. Previously, the best model was selected based on the Mean Absolute Error (MAE) metric, which was the Multi-Layer Perceptron (MLP) method using ERA5 satellite data. This model was chosen because it provided the highest accuracy compared to other models, as indicated by the lowest MAE value. MLP, as a type of artificial neural network, is capable of handling data complexity effectively, making it well-suited for imputing missing rainfall data in BMKG stations across East Java. The ERA5 satellite data, which provides weather and climate information with spatial resolution, was used as input to train the model. The selected model was then applied to fill in the missing rainfall data at BMKG stations in East Java. Once the imputation process was completed, visualization was conducted to compare the original data with the imputed data. This visualization aims to observe how well the model captures the variability patterns of rainfall and to assess the accuracy and consistency of the imputation results compared to the available data.

This visualization is based on several sample stations from different years. The x-axis represents the days from the first to the last day of the observation period, while the y-axis represents the rainfall values (mm). The imputed missing rainfall data is visualized using blue bar charts, while the original rainfall data is displayed as a red line chart. The dashed red line indicates the presence of missing data, whereas the blue bars represent the predicted values that replace the missing data. Overall, this visualization demonstrates that the selected method effectively captures rainfall patterns, particularly in filling in missing data without causing significant differences from the original observations. This confirms that the applied model achieves a high level of accuracy in imputing missing rainfall data.

4.5.2 Solution to Overfitting Problem

Overfitting occurs when a machine learning model produces accurate results on a training dataset but is ill-suited to a new dataset, producing inaccurate results. When the model encompasses every point in the training data set, it is said to be overfit (Anil & Singh, 2023). Noisy data and data overload, which occur when the data collection contains noisy and irrelevant information, are the primary causes of overfitting (Li & Spratling, 2023). The overfitting issue, which is explained below, is highlighted in **figure 11**.

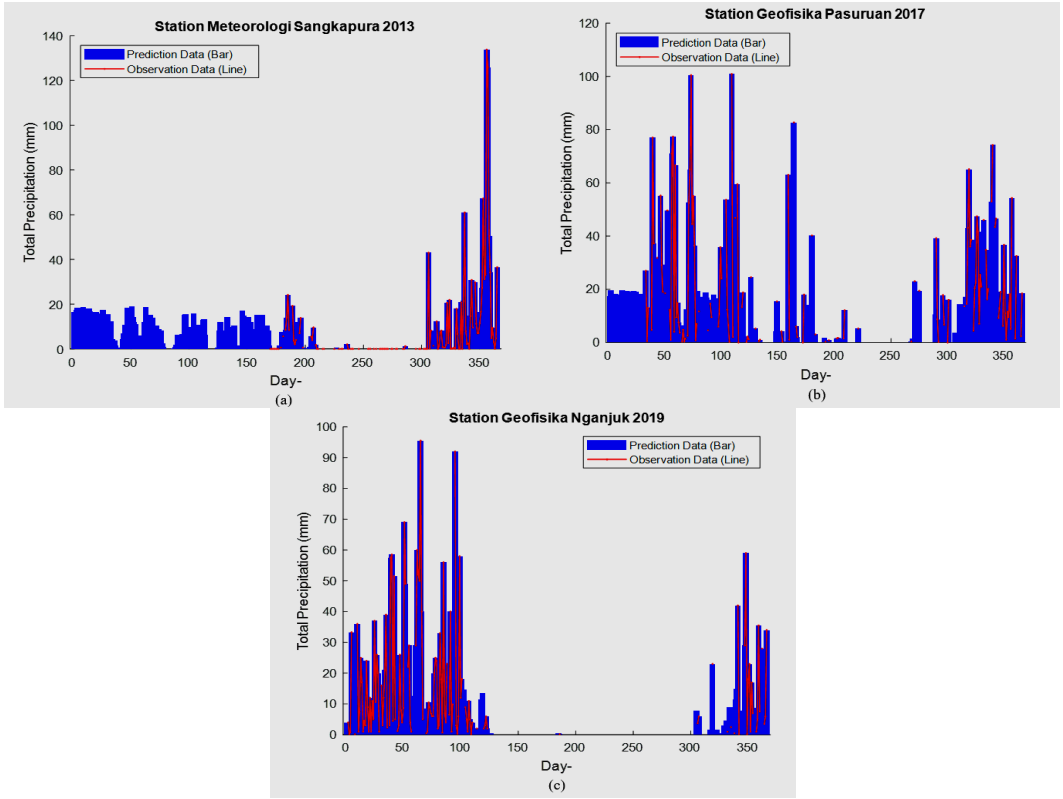


Fig. 10. Compared observation data and prediction data.

Fig. 11. Overfitting Model (*Anil & Singh, 2023*).

The overfitting issue arises when the machine learning model uses fewer and/or unprocessed data; the second issue arises when a more complex model is applied to a straightforward problem. The model will get extremely complex and overfitting will become an issue when the layer is increased. Accordingly, an overfitting model results from using a lot of layers in neural networks (Gupta & Sharma, 2022). To avoid overfitting, we take the steps as follows (Anil & Singh, 2023).

1. Model selection: It is necessary to comprehend the dataset, its type, and the model that will yield the best outcomes for the dataset at hand. CNN and MLP for spatial datasets, MLR for linear datasets, SVR for nonlinear datasets, or other models perform better.
2. Dataset filtering and feature extraction: The dataset and features are used to train the machine learning model. As a result, trustworthy features and filtered datasets solve the overfitting issue.
3. Reducing Layers/Complexity: A neural network that has too many internal, dense layers is too complex. The mode had a low bias when the model had more layers or complexity, which also led to an overfitting issue. As a result, neural networks' internal layers must be minimized or reduced.
4. Early Stopping: The model trains itself during the training session by repeatedly learning the data. Consequently, overfitting results from the model overtraining the training data. After the model

has been trained on the training data, it is required to limit the number of training repetitions and/or to terminate the model early, a process known as early stopping.

5. Employ Dropouts: A neural network has several layers, with several neurons in each layer. The term "dropout" refers to the random dropping of neurons to avoid overfitting certain of them. The neural network benefits more from this method.

5. CONCLUSIONS

This study aims to determine the most suitable machine learning method and satellite data for imputing rainfall data at BMKG stations in East Java. Based on the analysis of data proportions, scenarios with a more significant proportion of training data, such as 95:5%, yield better performance. In contrast, the 50:50% scenario shows less stable results due to the limited training data. Regarding machine learning methods, CNN demonstrates excellent performance with a low average loss value and good generalization ability. However, MLP shows the most significant potential for further analysis due to its lowest loss value. Regarding satellite data, ERA5 and CMORPH BLD are more stable and reliable for rainfall imputation modelling. Combining the MLP method and ERA5 satellite data delivers the best results with the lowest loss value.

ACKNOWLEDGEMENT

We thank the BIMA Kemendikbudristek program, which has provided funding support for this research through the scheme "Penelitian Tesis Magister (PTM)" with contract number 027/E5/PG.02.00.PL/2024 and subcontract number 22306/IT3.D10/PT.01.03/P/B/2024. In addition, we also thank the Department of Mathematics, IPB University, for the support provided in this research.

The codes are available from the corresponding/first author upon reasonable request.

R E F E R E N C E S

Afham, M., Nur, I. M., & Utami, T. W. (2017). Pemodelan Regresi Ridge pada Kasus Curah Hujan di Kota Semarang. In *PROSIDING SEMINAR NASIONAL & INTERNASIONAL*.

Alwanda, M. R., Ramadhan, R. P. K., & Alamsyah, D. (2020). Implementasi metode convolutional neural network menggunakan arsitektur LeNet-5 untuk pengenalan doodle. *Jurnal Algoritme*, *1*(1), 45-56.

Andana, A. P., Safitri, D., & Rusgiyono, A. (2017). Model Regresi Menggunakan Least Absolute Shrinkage and Selection Operator (Lasso) Pada Data Banyaknya Gizi Buruk Kabupaten/Kota Di Jawa Tengah. *Jurnal Gaussian*, *6*(1), 21-30.

Anil, A. K. P., & Singh, U. K. (2023). An Optimal Solution to the Overfitting and Underfitting Problem of Healthcare Machine Learning Models. *Journal of Systems Engineering and Information Technology (JOSEIT)*, *2*(2), 77-84.

Benhar, H., Idri, A., & Fernández-Alemán, J. L. (2020). Data pre-processing for heart disease classification: A systematic literature review. *Computer Methods and Programs in Biomedicine*, *195*, 105635.

Bishop, C. M. (2006). Pattern recognition and machine learning. Springer Google Scholar, 2, 1122–1128.

Brereton RG, Lloyd GR. 2010. Support vector machines for classification and regression. *Analyst*, 135(2), 230-267. DOI:10.1039/b918972f

Chiu, D. K. (2001). Book review:" Pattern classification," RO Duda, PE Hart, and DG Stork. *International Journal of Computational Intelligence and Applications*, 1(03), 335–339.

Desmonda, D., Tursina, T., & Irwansyah, M. A. (2018). Prediksi besaran curah hujan menggunakan metode fuzzy time series. *JUSTIN (Jurnal Sistem dan Teknologi Informasi)*, 6(4), 145-149.

de Oliveira, L. F., Antonini, J. C. D. A., Fioreze, A. P., & da Silva, M. A. (2008). Maximum rainfall estimation methods for Goiás. *Revista Brasileira de Engenharia Agrícola e Ambiental*, *12*, 620-625.

Dozat, T. (2016). *Incorporating nesterov momentum into adam*.

Duarte, L. V., Formiga, K. T. M., & Costa, V. A. F. (2022). Comparison of methods for filling daily and monthly rainfall missing data: statistical models or imputation of satellite retrievals? *Water*, *14*(19), 3144.

Falck, A. S., Maggioni, V., Tomasella, J., Vila, D. A., & Diniz, F. L. (2015). Propagation of satellite precipitation uncertainties through a distributed hydrologic model: A case study in Brazil's Tocantins–Araguaia basin.

Journal of Hydrology, *527*, 943-957.

Fukushima K. (1980). Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biological cybernetics*, 36(4), 193–202. DOI: 10.1007/BF00344251.

Gunadi, G. (2022). Penerapan Algoritma K-Means Clustering untuk Menganalisa Transaksi Penjualan Jasa Cetak pada Unit Print on Demand (POD) Percetakan Gramedia. *Infotech*: *Journal of Technology Information*, 8(2), 117–126. DOI: 10.37365/jti.v8i2.148.

Gupta, G. K., & Sharma, D. K. (2022, March). A review of overfitting solutions in smart depression detection models. In *2022 9th International conference on computing for sustainable global development (INDIACom)* (pp. 145-151). IEEE.

Hawkins, D. M. (2004). The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1), 1–12.

He, X. J. (2018). Crude oil prices forecasting: time series vs. SVR models. *Journal of International Technology and Information Management*, 27(2), 25-42.

Hicham A., Jeghal A., Sabri A., & Tairi H. (2020). A Survey on Educational Data Mining [2014-2019]. In 2020 International Conference on Intelligent Systems and Computer Vision (ISCV) (pp. 1-6). *IEEE*. DOI: 10.1109/ISCV49265.2020.9204013.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1), 55-67.

Honghai F., Guoshun C., Cheng Y., Bingru Y., Yumei C. (2005). A SVM Regression Based Approach to Filling in Missing Values. In International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (pp. 581-587). Berlin, Heidelberg: *Springer Berlin Heidelberg*. DOI: 10.1007/11553939_83.

Jerez, J. M., Molina, I., García-Laencina, P. J., Alba, E., Ribelles, N., Martín, M., & Franco, L. (2010). Missing data imputation using statistical and machine learning methods in an actual breast cancer problem. *Artificial intelligence in medicine*, 50(2), 105-115.

Junninen, H., Niska, H., Tuppurainen, K., Ruuskanen, J., & Kolehmainen, M. (2004). Methods for imputation of missing values in air quality data sets. *Atmospheric environment*, *38*(18), 2895-2907.

Kingma, D. P. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv*:1412.6980.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.s

Li L. (2020). A Robust Deep Learning Approach for Spatiotemporal Estimation of Satellite AOD and PM2. 5. *Remote Sensing*, 12(2), 264. DOI:10.3390/rs12020264.

Li Z, Liu F, Yang W, Peng S, Zhou J. 2021. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12), 6999-7019. DOI: 10.1109/TNNLS.2021.3084827.

Li, L., & Spratling, M. (2023). Understanding and combating robust overfitting via input loss landscape analysis and regularization. *Pattern Recognition*, *136*, 109229.

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. *International conference on learning representations.*

Loshchilov, I., & Hutter, F. (2016). Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv*:1608.03983.

Magyari-Sáska, Z., Haidu, I., & Magyari-Sáska, A. (2025). Experimental Comparative Study on Self-Imputation Methods and Their Quality Assessment for Monthly River Flow Data with Gaps: Case Study to Mures River. *Applied Sciences*, *15*(3), 1242. https://doi.org/10.3390/app15031242

Mitchell, T. M., & Mitchell, T. M. (1997). *Machine learning* (Vol. 1, No. 9). New York: McGraw-Hill.

Nóbrega R.S., Souza E.P., Galvíncio J.D. (2008). Análise da Estimativa de Precipitação do TRMM Em Uma Sub-Bacia da Amazônia Ocidental. *Revista de Geografia*, 25(1), 6-20.

Najib, M. K., Nurdiati, S., & Sopaheluwakan, A. (2022). Copula-based joint distribution analysis of the ENSO effect on the drought indicators over Borneo fire-prone areas. *Modeling Earth Systems and Environment*, 1-10.

Nasution, T. H., & Andayani, U. (2017, March). Recognition of roasted coffee bean levels using image processing and neural network. In *IOP Conference Series: Materials Science and Engineering* (Vol. 180, No. 1, p. 012059). IOP Publishing.

Nurdiati, S., Khatizah, E., Najib, M. K., & Hidayah, R. R. (2021, February). Analysis of rainfall patterns in Kalimantan using fast fourier transform (FFT) and empirical orthogonal function (EOF). In *Journal of Physics: Conference Series* (Vol. 1796, No. 1, p. 012053). IOP Publishing.

Nurdiati S., Najib M.K., Bukhari F., Revina R., Salsabila F.N. (2022). Performance Comparison of Gradient-Based Convolutional Neural Network Optimizers for Facial Expression Recognition. Barekeng: *Jurnal Ilmu Matematika dan Terapan*, 16(3), 927-938. DOI: 10.30598/barekengvol16iss3pp927-938.

Nurdiati, S., Sopaheluwakan, A., Pratama, Y. A., & Najib, M. K. (2021). Statistical bias correction on the climate model for el nino index prediction. *Al-Jabar: Jurnal Pendidikan Matematika*, *12*(2), 273-282.

Partarini N.M.C., Sujono J., Pratiwi E.P.A. (2021). Koreksi dan Validasi Data Curah Hujan Satelit GPM-IMERG dan CHIRPS di DAS Selorejo, Kabupaten Malang. *Civil Engineering, Environmental, Disaster & Risk Management Symposium (CEEDRiMS) Proceeding 2021*.

Rafhida, S. A., Nurdiati, S., Budiarti, R., & Najib, M. K. (2024). Bias Correction of Lake Toba Rainfall Data Using Quantile Delta Mapping. *CAUCHY: Jurnal Matematika Murni dan Aplikasi*, 9(2), 297-309.

Rasywir, E., Pratama, Y., & Fachruddin, F. (2022). Eksperimen Pengujian Optimizer dan Fungsi Aktivasi Pada Code Clone Detection dengan Pemanfaatan Deep Neural Network (DNN). *Building of Informatics, Technology and Science (BITS)*, *4*(2), 405-412.

Saleh A.M.E., Arashi M., Kibria B.G. (2019). *Theory of Ridge Regression Estimation with Applications*. John Wiley & Sons.

Septiawan P., Nurdiati S., Sopaheluwakan A. (2019). Numerical Analysis Using Empirical Orthogonal Function Based on Multivariate Singular Value Decomposition on Indonesian Forest Fire Signal. *In IOP Conference Series: Earth and Environmental Science* (Vol. 303, No. 1, p. 012053). IOP Publishing. DOI: 10.1088/1755-1315/303/1/012053.

Silva-Ramírez, E. L., Pino-Mejías, R., López-Coello, M., & Cubiles-de-la-Vega, M. D. (2011). Missing value imputation on missing completely at random data using multilayer perceptrons. *Neural Networks*, 24(1), 121-129.

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, *14*, 199-222.

Soares A.S.D., Paz A.R.D., Piccilli D.G.A. (2016). Avaliação Das Estimativas de Chuva do Satélite TRMM no Estado da Paraíba. *RBRH*, 21, 288-299. DOI: 10.21168/rbrh.v21n2.p288-299.

Sultana, F., Sufian, A., & Dutta, P. (2018, November). Advancements in image classification using convolutional neural network. In *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)* (pp. 122-129). IEEE.

Tabony R.C. (1983). The Estimation of Missing Climatological Data. *Journal of Climatology*, 3(3), 297–314. DOI: 10.1002/joc.3370030308.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1), 267-288.

Vicente-Serrano S.M., Beguería S., López-Moreno J.I., García-Vera M.A., Stepanek P. (2010). A Complete Daily Precipitation Database for Northeast Spain: Reconstruction, Quality Control, and Homogeneity. *International Journal of Climatology*, 30(8), 1146-1163. DOI: 10.1002/joc.1850.

Vieira S., Gong Q.Y., Pinaya W.H., Scarpazza C., Tognin S., Crespo-Facorro B., Mechelli A. (2020). Using Machine Learning and Structural Neuroimaging to Detect First Episode Psychosis: Reconsidering the Evidence. *Schizophrenia Bulletin*, 46(1), 17-26. DOI: 10.1093/schbul/sby189.

Wagner P.D., Fiener P., Wilken F., Kumar S., Schneider K. (2012). Comparison and Evaluation of Spatial Interpolation Schemes for Daily Rainfall in Data Scarce Regions. *Journal of Hydrology*, 464, 388-400. DOI: 10.1016/j.jhydrol.2012.07.026.

Wangwongchai A., Waqas M., Dechpichai P., Hlaing P.T., Ahmad S., Humphries U.W. (2023). Imputation of Missing Daily Rainfall Data; A Comparison Between Artificial Intelligence and Statistical Techniques. *Methods*X, 11, 102459. DOI:10.1016/j.mex.2023.102459.

Wang, X., Li, A., Jiang, Z., & Feng, H. (2006). Missing value estimation for DNA microarray gene expression data by Support Vector Regression imputation and orthogonal coding scheme. *BMC Bioinformatics*, 7, 1-10.

Yang X., Cui R., Tian C., Hu S., Jiang J., Xu P. (2022). Linear Spline and CNN-LSTM for Missing Values Imputation of Beidou Satellite Radiation Dose Data. *Chinese Journal of Space Science*, 42(1), 163. DOI:10.11728/cjss2022.01.201116100.