Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 4, August 2025, Page. 2811-2824
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5044

Comparison of Accuracy and Computation Time for Predicting Earthquake Magnitude in Java Island

Abdul Hakim Prima Yuniarto*[1], Taqwa Hariguna[2], Devi Astri Nawangnugraeni[3]

[1]Electrical Engineering, Sekolah Tinggi Teknik Wiworotomo Purwokerto, Indonesia
[2]Master of Computer Science, Universitas Amikom Purwokerto, Indonesia
[3]Informatics, Universitas Jenderal Soedirman, Indonesia

Email: [1]a.hakim.py@gmail.com

Received : Jul 6, 2025; Revised : Aug 28, 2025; Accepted : Aug 29, 2025; Published : Sep 2, 2025

Abstract

Java Island has numerous active faults, making earthquake magnitude prediction a crucial component of disaster mitigation efforts. This study conducted a rigorous comparative analysis of four machine learning algorithms—Random Forest, Neural Network, Linear Regression, and Support Vector Machine—to determine their effectiveness in this specific task. The methodology employed involved systematic hyperparameter optimization for each model to ensure a fair and robust evaluation, with performance measured by Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and training time. The results showed that all three nonlinear models significantly outperformed Linear Regression. Random Forest achieved the highest accuracy (RMSE 0.5445), but Support Vector Machine and Neural Network demonstrated very competitive and nearly equal performance. The study concluded that while Random Forest has a slight advantage, several state-of-the-art models are highly capable of addressing this problem after appropriate optimization. This underscores the critical role of methodical tuning and implies that model selection in practical applications depends on a trade-off between modest improvements in accuracy and computational efficiency.

Keywords: *Earthquake, Linear Regression, Neural Network, Random Forest, Support Vector Machine*

1. INTRODUCTION

1.1. Background

Indonesia, a country situated at the intersection of several major tectonic plates, is inherently vulnerable to seismic activity. In particular, the island of Java, which is the center of the country's population, economy, and government, is under constant threat from potential earthquakes. This threat is not a hypothesis, but a well-documented geological reality. Based on data from the Geological Survey Center, several active faults on the island of Java can trigger earthquakes of significant strength, thereby drastically increasing the risk of natural disasters in the region [1]. These faults, as visually shown in Figure 1, are cracks in the Earth's crust where movement has occurred. The red lines on the map are not just geographic markers, but representations of latent sources of seismic energy that can be released at any time, causing severe shaking.

Earthquakes are one of the most destructive natural disasters, given the force of the shock that can flatten buildings and infrastructure, and cause significant loss of life [2]. The impact of earthquakes is not limited only to physical damage, but also extends to social and economic aspects, causing long-term psychological trauma, paralyzing local economic activities, and requiring enormous recovery costs.

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 4, August 2025, Page. 2811-2824
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5044

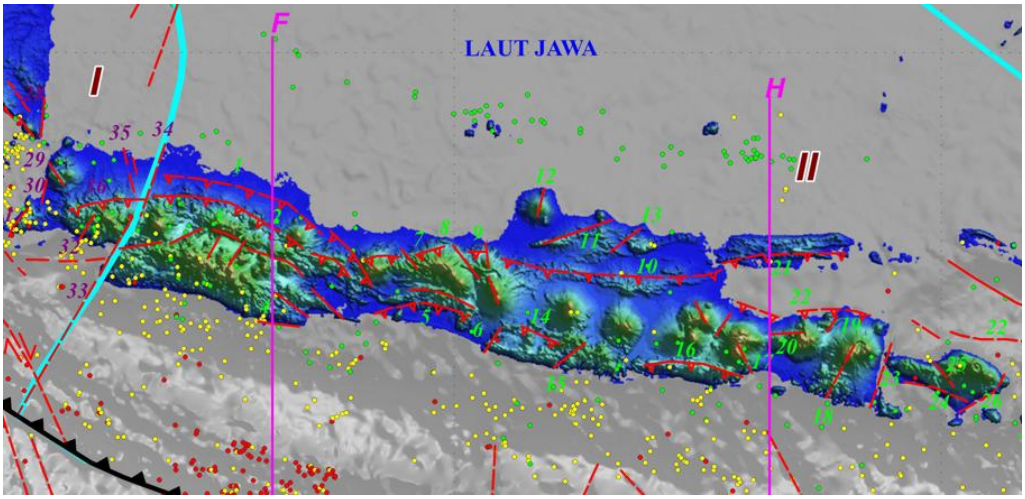

Figure 1. Map of Active Faults in Java Island

History records a series of tragic events on the island of Java that serve as a tangible reminder of the destructive power of earthquakes. Some examples of major earthquakes that have occurred on the island of Java include the 2006 earthquake in Yogyakarta, with a magnitude of 5.9 SR, which shook the Bantul, Sleman, and surrounding areas, resulting in 5,782 deaths and approximately 66,359 houses being severely damaged [3], [4]. This event highlights the vulnerability of densely populated communities when confronted with unexpected natural forces.

More recently, the 2022 Cianjur earthquake with a magnitude of 5.6 SR also recorded up to 600 fatalities and caused significant infrastructure damage in the area [5], [6]. Although the magnitude was moderate, the impact was catastrophic because it occurred on land with a shallow depth and in an area with a building density that did not meet earthquake-resistant standards. In addition, there was the Batang earthquake in 2024. The earthquake had a magnitude of 4.4 SR and shook the Batang, Pekalongan, and surrounding areas. Although it did not cause significant casualties, this earthquake damaged many homes and infrastructure in the area [7], [8], showing that even earthquakes with smaller magnitudes still have the potential for damage that cannot be ignored. These incidents highlight the importance of disaster preparedness, particularly in regions with a high earthquake potential, such as Java.

Given the significant potential of these active faults, predicting earthquake magnitude is crucial for practical mitigation efforts. Although predicting the exact time and location of an earthquake remains the biggest challenge in seismology, efforts to predict potential magnitudes based on historical data and geophysical parameters represent a crucial step forward. This magnitude prediction can provide valuable information for planning and preparation, particularly in early warning systems and disaster loss reduction. With an estimate of the earthquake strength, the government and related institutions can design safer spatial planning, strengthen building standards, and develop more effective and targeted evacuation plans.

This is where the role of modern computing technology becomes very relevant. One approach that can be used to predict earthquake magnitude is to apply machine learning methods, which enable the analysis of complex patterns and relationships in earthquake data [9]. Machine learning, with its ability to "learn" from large volumes of data, offers the hope of identifying hidden correlations between various precursor variables (such as location, depth, and time of previous earthquakes) and future earthquake magnitudes. This approach goes beyond conventional statistical analysis by addressing the non-linear relationships that are often characteristic of complex and chaotic geophysical systems.

Several previous studies have applied various machine learning algorithms to predict earthquake magnitude. For example, Ade Fauzan and Defri Ahmad have predicted the magnitude of an earthquake

Vol. 6, No. 4, August 2025, Page. 2811-2824
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5044

in Padang City using the Random Forest technique, yielding an RMSE value of 0.31758 and an MSE value of 0.10085 [10]. Then, Oman Somantri has predicted the strength of an earthquake in Indonesia using the Neural Network method optimized by a genetic algorithm, and the prediction results obtained an RMSE value of 0.708 [11]. Furthermore, Annisa Alifa et al. have predicted the strength of an earthquake using the Linear Regression algorithm, yielding prediction results with an RMSE value of 48.8352, a MAPE value of 1.2564, and a MAE value of 24.065 [12]. Then, Oman Somantri et al. predicted the strength of earthquakes using a Support Vector Machine model based on window parameters, obtaining a prediction result with an RMSE value of 0.712 [13].

However, although these studies have been conducted and demonstrate the potential of each algorithm, there has been no study that directly compares the four algorithms in terms of accuracy and computation time for predicting earthquake magnitude in the same geospatial context, namely Java Island. Each of these studies stands alone with possibly different datasets and scopes, making it difficult to draw valid conclusions about which algorithm is inherently superior for this task. Therefore, this study aims to fill this gap by comparing the Random Forest, Neural Network, Linear Regression, and Support Vector Machine algorithms in terms of two main aspects, namely prediction accuracy and computation time. Thus, the formulation of the problem addressed in this study is: Which algorithm has the best accuracy in predicting earthquake magnitude, and which algorithm has the best computational time for such predictions? This question has two equally important sides. High accuracy is the main goal to ensure the reliability of predictions. Still, fast computation time is also crucial if this model is to be implemented in an early warning system that requires the quickest possible response.

This research contributes to the understanding of the most effective machine learning algorithm for predicting earthquake magnitude, considering both accuracy and computation time factors simultaneously. The results of this study can provide significant contributions to the development of more efficient earthquake prediction models that can be implemented to improve disaster mitigation in Java Island. Understanding the trade-off between accuracy and speed will be a valuable guide for data scientists, engineers, and policymakers in selecting and developing effective decision-making tools for addressing seismic threats.

1.2. Earthquake Magnitude Prediction Algorithm

To answer the problem formulation, this study will implement and compare four popular machine learning algorithms that have different fundamental approaches to data modeling.

Random Forest

Random Forest (RF) is an ensemble learning algorithm for classification and regression tasks. Illustrated in Figure 2, this algorithm operates by constructing multiple decision trees on a randomly selected subset of the training data. This approach, known as bagging, aims to produce more stable predictions and prevent overfitting, where the final prediction for regression is determined by averaging the results from all trees [14].

A significant advantage of Random Forest (RF) is its ability to handle both categorical and continuous variables, as well as its tolerance for missing data [15]. In addition, its ability to process complex, non-linear, and high-dimensional data makes RF very suitable for various modelling needs, including in the context of earthquake prediction, which is non-linear [16], [17].

As an ensemble model, Random Forest does not have a single mathematical formulation like Linear Regression. Instead, it builds several K different decision trees during the training process.

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 4, August 2025, Page. 2811-2824
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5044

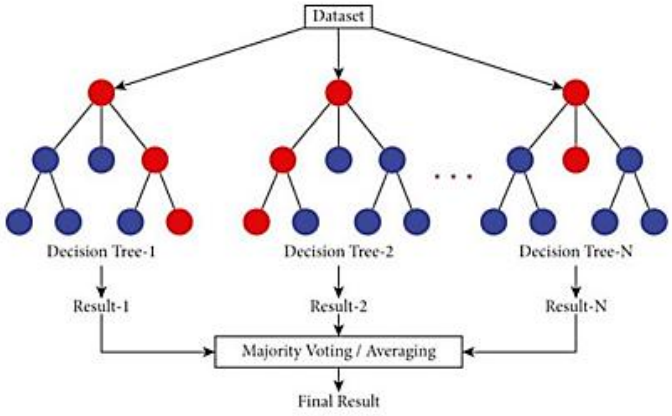

Figure 2. Random Forest

For regression tasks, the final prediction is the average of the predictions of all individual decision trees. The prediction formulation is as follows:

$$\hat{y}_{rf} = \frac{1}{K}\sum_{k=1}^{K} T_k(x) \quad (1)$$

Information:
y_{rf} = final prediction results from RF
K = total number of Decision trees
$T_k(x)$ = prediction from the k-th Decision tree input x

Neural Network

A Neural Network is an algorithm for identifying data patterns through a process inspired by the workings of the human brain [18]. Like the brain, this network consists of interconnected processing units (neurons) [19]. Each neuron functions to receive input, process it mathematically, and produce output in response.

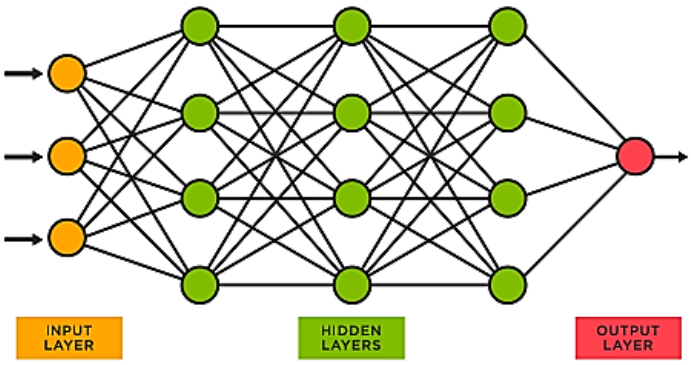

Figure 3. Neural Network

Illustrated in Figure 3, a Neural Network architecture generally consists of three types of layers: an input layer for receiving data, one or more hidden layers for computation, and an output layer for the final prediction [20]. Its ability to model highly complex and abstract relationships makes it a strong candidate for earthquake prediction problems where patterns are challenging to detect.

Artificial Neural Networks consist of processing units called neurons. Each neuron receives one or more inputs, processes them, and forwards the results to other neurons. The computational process in a neuron can be formulated as follows:

Vol. 6, No. 4, August 2025, Page. 2811-2824
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5044

$$y = f(\sum_{i=1}^{n}(w_i x_i) + b) \quad (2)$$

Information:
y = output of neurons
x_i = i-th input signal
w_i = weight of each input
b = bias to shift the activation function
f = activation function

Linear Regression

Linear regression is a fundamental forecasting method that models a linear relationship between variables for prediction [21]. This method assumes a linear relationship, as in Figure 4, to illustrate the relationship between these variables [22]. Simple, this model serves as a baseline for evaluating more complex models.

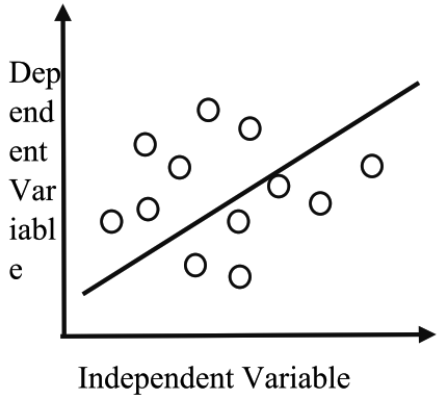

Figure 4. Linear Regression

The formula for obtaining the equation of a line in linear regression can be written as follows [23].

$$Y = a + bx \quad (3)$$

Information:
Y = dependent variable
α = intercept
b = regression coefficient
x = independent variable

Although its linearity assumption may be a limitation in modelling complex natural phenomena such as earthquakes, its computational speed and ease of interpretation make Linear Regression still relevant to be tested in this comparative study.

The linear regression model aims to model the relationship between a dependent variable (Y) and one or more independent variables (X) by fitting a linear equation. For cases with multiple features (multivariate) such as latitude, longitude, and depth, the equation can be written as follows:

$$\hat{Y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p \quad (4)$$

Information:
Y = predicted magnitude value
β_0 = intercept
β_1,β_2,β_p = regression coefficient

Vol. 6, No. 4, August 2025, Page. 2811-2824
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5044

Support Vector Machine

The Support Vector Machine (SVM) is a machine learning algorithm used for classification and prediction [24], with a variant, Support Vector Regression (SVR), used for regression tasks. The primary concept is to identify the optimal hyperplane that maximises the margin or separation distance between two data classes [25]. The working concept of SVM for classification is illustrated in Figure 5.

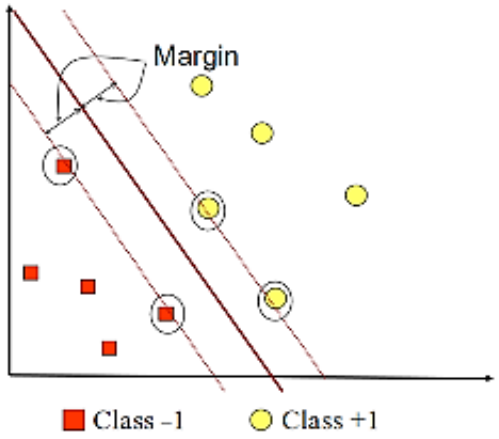

Figure 5. Support Vector Machine

SVM can handle both linear and non-linear data cases. For non-linear data, SVM employs the kernel trick concept, which maps the data to a higher-dimensional space to find a separating hyperplane. The goal of this approach is to maximise the margin between classes [26], so SVM is very powerful for data that cannot be separated linearly, such as seismic data.

In the context of regression, SVM operates under a principle known as Support Vector Regression (SVR). Unlike regular regression, which attempts to minimise error, SVR attempts to fit a regression function to the data while ensuring that the deviation (error) of most data points falls within a specified tolerance margin, called ϵ (epsilon). The optimisation goal is:

Minimize $$\frac{1}{2}\|\boldsymbol{w}\|^2 \quad (5)$$

With constrain $$|\boldsymbol{y_i} - (\boldsymbol{w}.\boldsymbol{x_i} + \boldsymbol{b})| \leq \boldsymbol{\epsilon} \quad (6)$$

Information:
w = weight vector
b = bias
ϵ = tolerance margin
x_i = feature vector of the i-th data
y_i = actual value

2. METHOD

2.1. Research Design

This research is a quantitative comparative study comparing four machine learning algorithms: Random Forest, Neural Network, Linear Regression, and Support Vector Machine. Four algorithms were chosen because they represent diverse fundamental approaches. The algorithms were evaluated based on their accuracy and computational time for the task of predicting earthquake magnitudes in

Vol. 6, No. 4, August 2025, Page. 2811-2824
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5044

Java. The evaluation was conducted objectively by testing all algorithms on the same dataset and using the same computing environment.

The entire experimental process, from data preprocessing to model evaluation, was implemented using the Python programming language within the Google Colaboratory environment. The scientific libraries that served as the foundation for this research included Pandas for data manipulation, Scikit-learn for machine learning model implementation and evaluation, and Matplotlib and Seaborn for data visualisation. The use of these open-source tools ensures transparency and ease of replication of the research.

2.2. Research Procedures

To ensure valid and replicable results, this study followed a series of structured and logical steps. This study went through several research stages. These stages must be carried out sequentially to achieve optimal results, as the output from one stage becomes the input for the next. These stages are shown in Figure 6 below.

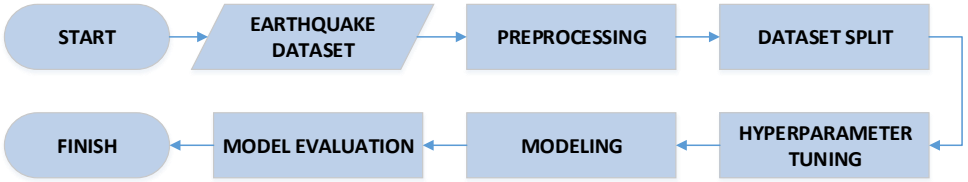

Figure 6. Research Flowchart

Based on Figure 6, the research flow begins with preprocessing the earthquake dataset. Next, the dataset is divided into training and test data. The training data is used to create predictive models for the four tested algorithms, followed by a hyperparameter tuning process to find the best model configuration. Finally, the optimized models are evaluated using the test data to measure and compare performance based on accuracy and computation time.

2.3. Dataset

The data source used in this study is the Indonesian earthquake catalogue dataset obtained from Kaggle, a leading public data repository platform. This dataset was selected due to its comprehensiveness and long temporal span (2008–2023), encompassing 92,887 earthquake event records. This large data volume and long period provide a rich and sufficient historical basis for training robust machine learning models. Parameters such as event time, location (latitude and longitude), depth, and magnitude are fundamental in seismological analysis, ensuring the data's relevance for this study.

2.4. Preprocessing

Raw data is rarely directly usable for modelling. Preprocessing is crucial for cleaning, filtering, and selecting the most relevant data to ensure the quality of model input.

- **Data Filtering Based on Location**
  The first step was to filter the data to include only earthquake events located around Java Island. The justification for this step was the research objective of building a regionally specific prediction model. Seismotectonic characteristics, such as the type and behaviour of active faults, can vary significantly across regions in Indonesia. By focusing the dataset on a single, homogeneous geographic region, the model was expected to learn more relevant local patterns and produce more accurate predictions for that region. After this process, the dataset used for modelling consisted of 9,395 data points.

Vol. 6, No. 4, August 2025, Page. 2811-2824
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5044

- **Attribute Selection**
  The next step was attribute selection. Of the 13 available attributes, only lat (latitude), lon (longitude), depth (depth), and mag (magnitude) were retained. Latitude, longitude, and depth were selected as predictor variables (features) because they are fundamental geospatial coordinates that define the earthquake hypocenter, which is theoretically the most influential factor in determining the energy released (magnitude). The mag attribute was designated as the target variable (label) to be predicted. Other attributes were excluded because initial exploratory data analysis indicated a very high percentage of missing values. Including these attributes would require complex imputation techniques that could potentially introduce bias into the model or drastically reduce the sample size if rows with missing values were removed. Therefore, to maintain data integrity and volume, it was decided to focus on the primary predictor attributes for which the data were most complete and relevant.

2.5. Split Dataset

Separating a dataset into training and test data is a fundamental procedure for obtaining an objective and unbiased evaluation of model performance. This practice is crucial for preventing overfitting, a condition in which a model memorises too much of the training data and is unable to generalise to new data. An overfitting model will perform very well on the training data but poorly on the test data, making it useless in real-world applications [27].

2.6. Algorithm Model Selection

After data preparation and establishing a validation strategy, the next step is to select a model. This study strategically selected four models Random Forest, Neural Network, Linear Regression, and Support Vector Machine to cover a broad spectrum of complexity and modelling approaches.

- **Linear Regression** was chosen as the base model. Due to its simplicity, this model serves as a fundamental benchmark for comparison. Its performance will indicate the extent to which a simple linear relationship can explain variations in earthquake magnitude. If more complex models fail to significantly outperform Linear Regression, it may indicate that the data lacks strong nonlinear patterns or that the features used are insufficiently informative.
- **Neural networks** were chosen because of their capacity as universal approximators. Theoretically, with exemplary architecture, neural networks can model highly complex, nonlinear functions. This makes them an up-and-coming candidate for geophysical phenomena, such as earthquakes, where the interactions between variables can be complex and non-intuitive.
- **Random Forest** was chosen because of its reputation as one of the best off-the-shelf algorithms and its robustness. Its ensemble nature makes it less susceptible to overfitting than a single decision tree. Its ability to implicitly handle feature interactions and its solid performance across a wide range of problem domains make it a prime contender for this comparative study.
- **Support Vector Machines** were chosen for their unique geometric approach to finding optimal decision boundaries. Using the kernel trick, SVMs can efficiently model nonlinear boundaries, making them a powerful alternative to Neural Networks and Random Forests in handling data complexity.

2.7. Model Performance Evaluation

To quantitatively compare the performance of the four models, clear and relevant evaluation metrics are needed. This study uses two main criteria: predictive accuracy and computational efficiency.

- **Root Mean Squared Error (RMSE)**

Vol. 6, No. 4, August 2025, Page. 2811-2824
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5044

In regression problems, accuracy is measured by how close the predicted values are to the actual values, and the Root Mean Squared Error (RMSE) is a commonly used metric to measure it. RMSE is popular because it provides a significant penalty for large errors, and the results are easy to interpret because they have the same units as the target variable. The smaller the RMSE value, the better the forecasting accuracy [28]. The RMSE equation used is as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{l}(y_i - \bar{\bar{y}}_l)^2} \quad (7)$$

Information:
n = amount of data
i = whole of data
y_i = actual value
y_l = predicted value

- **Mean Absolute Error (MAE)**
  MAE measures the average of the absolute values of the errors. Unlike RMSE, MAE provides a more intuitive picture of the average magnitude of the prediction error because it does not square the error. This metric is less sensitive to outliers than RMSE. A lower MAE value also indicates better performance.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (8)$$

Information:

n = jumlah total dataamount of data
y_i = magnitude actual value
$\hat{y}_i$ = magnitude predicted value

- **Training Time**
  In addition to accuracy, model efficiency is a crucial practical factor. Time to Train measures the total duration (in seconds) required for an algorithm to process training data and build a predictive model until it is ready for use. This metric represents the computational burden of a model, with shorter times indicating higher efficiency. In this study, time was measured by recording the difference in system time before and after the model training (model fitting) execution in a Python environment.

2.8. Hyperparameter Optimization

To find the optimal hyperparameter combination (except for Linear Regression), this study uses the Grid Search with Cross-Validation (Grid Search CV) method. This process works as follows:

- Defines a "grid" containing all possible hyperparameter values.
- Systematically test each combination of hyperparameters in the grid.
- Evaluate each combination using Cross Validation to ensure stable performance.
- Select the combination with the best average cross-validation score as the final model configuration.

3. RESULT

3.1. Best parameters

To ensure optimal performance for each model, a hyperparameter optimization process was performed. This process is crucial because the correct parameters can significantly improve the model's

Vol. 6, No. 4, August 2025, Page. 2811-2824
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5044

ability to learn data patterns. Table 1 presents the optimal parameters identified for each algorithm following the tuning process.

Table 1. Best Model Parameters

No	Algorithm	Best Parameters	Best Score (CV_neg_MSE)
1	Random Forest	max_depth: 10, min_samples_leaf: 2, n_estimators: 200	-0.3253
2	Support Vector Machine	activation: relu, alpha: 0.001, hidden_layer_sizes: (50, 50)	-0.3517
3	Neural Network	C: 50, gamma: scale, kernel: rbf	-0.3525
4	Linear Regression	-	-

In Random Forest, using n_estimators of 200 means the model builds 200 decision trees, creating a robust and stable ensemble. Setting max_depth to 10 and min_samples_leaf to 2 helps prevent overfitting by limiting the complexity of each tree.

For Support Vector Machines, the RBF (Radial Basis Function) kernel is very effective for handling non-linear data relationships, mapping the data to a higher-dimensional space to find the optimal separating hyperplane. A C value of 50 balances classification error and margin width, while a gamma scale adjusts for the influence of each training sample.

In Neural Networks, an architecture with two hidden layers, each containing 50 neurons (hidden_layer_sizes=(50, 50)), provides sufficient capacity to learn complex patterns. The ReLU activation function and alpha regularization parameter of 0.001 help improve learning efficiency and prevent overfitting. This tuning process is crucial to finding an architecture that balances complexity and generalization ability.

3.2. Predictive Model Performance Analysis

After each model was optimised, a performance evaluation was conducted using the test data. Comparative results are presented in Table 2, which summarises the main evaluation metrics: Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) as indicators of accuracy, and Training Time as an indicator of efficiency.

In terms of accuracy, Random Forest emerged as the best-performing model, as indicated by the lowest RMSE and MAE values. However, it is worth noting that its superiority is not absolute; Support Vector Machine and Neural Network demonstrated very competitive performance with a tiny margin of error. This indicates that the three nonlinear models have nearly equal capabilities in modeling this data. Linear Regression lagged significantly in terms of accuracy, highlighting its limitations in nonlinear problems.

Table 2. Predictive Model Performance

No	Algorithm	RMSE	MAE	Training Time (s)
1	Random Forest	0.5445	0.4218	2,80
2	Support Vector Machine	0.5621	0.4283	4,58
3	Neural Network	0.5633	0.4438	2,90
4	Linear Regression	0.6295	0.4855	0,002

Vol. 6, No. 4, August 2025, Page. 2811-2824
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5044

In terms of efficiency, linear regression is the fastest due to its computational simplicity. Among the top three models, Random Forest and Neural Network show efficient and nearly identical training times (around 2.8-2.9 seconds), while Support Vector Machine takes slightly longer.

Overall, these quantitative metrics provide a clear performance ranking, but they do not explain the error behavior of each model. Therefore, further visual analysis is required.

3.3. Prediction Error Analysis

To gain deeper insights beyond aggregate metrics, a visual analysis of the prediction results was performed. Figure 7 presents a scatterplot comparing the actual magnitude values (x-axis) with the model-predicted magnitude values (y-axis) for the test data.

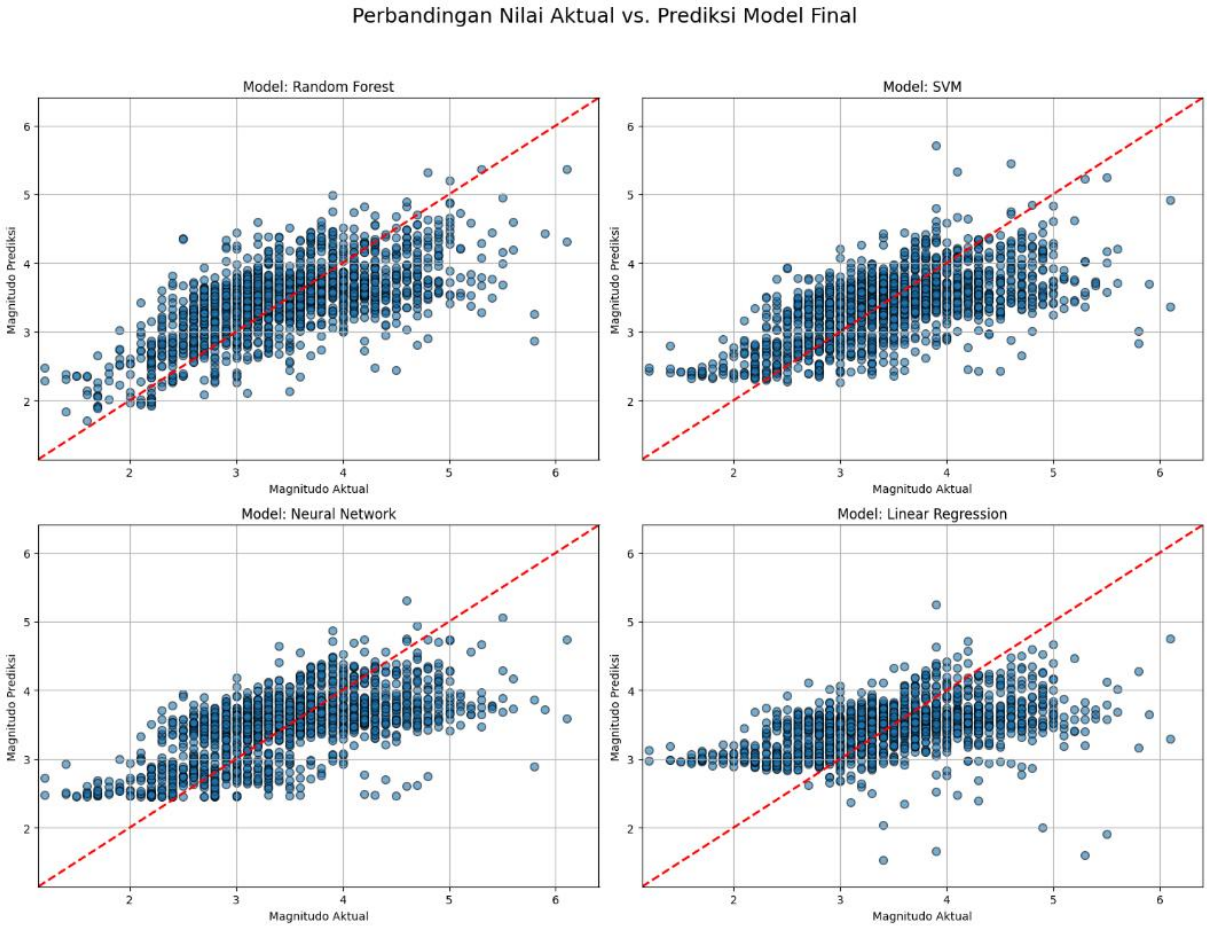

Figure 7. Comparison of Actual vs Predicted Data

Several visual interpretations can be drawn from Figure 9. In the Random Forest plot, the data points appear to be clustered much more closely around the diagonal line (the perfect prediction line). This visually confirms that the model's predictions have a high level of accuracy with low error variance.

In contrast, in plot of the linear regression, the data points are much more sparsely spaced and are spread further from the diagonal line. This indicates a greater degree of error in each prediction and visually explains why the RMSE and MAE values for this model are higher.

This visual analysis reinforces the findings from Table 2, showing that Random Forest's superiority is reflected not only in its lower average error value but also in the consistency of its predictions across a wide range of magnitudes.

Vol. 6, No. 4, August 2025, Page. 2811-2824
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5044

4. DISCUSSIONS

4.1. Model Performance Implications

The most significant finding is the substantial performance gap between the three non-linear models (Random Forest, SVM, and Neural Network) and the linear regression model. The significantly higher RMSE value of Linear Regression strongly indicates that the relationship between geospatial features (latitude, longitude, depth) and earthquake magnitude is highly non-linear. Seismic phenomena are inherently complex and chaotic geophysical processes; therefore, they cannot be accurately modelled simply by assuming a linear relationship. The ability of Random Forest, SVM with RBF kernel, and Neural Network to capture these complex interactions between features is key to their success.

4.2. Comparative Analysis of High-Performing Models

Although Random Forest achieved the highest accuracy score, its advantage over SVM and Neural Network was very slight. The RMSE difference of only about 0.02 points suggests that there is no single "magic model" for this problem. Instead, it demonstrates that several different nonlinear approaches are capable of achieving very competitive levels of performance after careful hyperparameter optimization. This means that selecting the best model in real-world applications will likely depend on trade-offs beyond pure accuracy, such as training time efficiency or ease of implementation, with Neural Network and Random Forest showing slightly better efficiency than SVM.

4.3. Comparison with Related Research

In this study, Random Forest produced an RMSE value of 0.5445. This result is slightly higher than the study by Fauzan and Ahmad [10], who obtained an RMSE of 0.31758 in Padang City. This difference is likely caused by several factors, including the distinct geospatial characteristics of the wider Java Island and the more specific city of Padang, as well as potential differences in the size and features of the datasets used. This indicates that earthquake prediction models can be susceptible to geographic location.

For Neural Network and Support Vector Machine, this study obtained significantly superior results (RMSE 0.5633 for NN and 0.5621 for SVM) compared to the study by Somantri et al. [11], [13] who reported RMSE 0.708 for NN and 0.712 for SVM in Indonesia. This significant performance improvement is likely due to the systematic and comprehensive implementation of a hyperparameter optimization strategy (Grid Search CV) in this study, which ensures each model is optimally configured for the dataset used.

The linear regression results in this study (RMSE = 0.6295) confirm the general finding that linear models are less suitable for this task. Although the RMSE value cannot be directly compared with the results of the study by Alifa et al. [12] (RMSE 48.8352) due to possible differences in data scale, the qualitative conclusion remains the same: linear models fail to capture the complexity of seismic data.

Overall, this comparison positions the performance of the models in this study in a highly competitive manner and underscores the importance of meticulous hyperparameter optimization to achieve high prediction accuracy.

4.4. Research Contribution

The primary contribution of this research is to provide a regionally specific performance benchmark for earthquake magnitude prediction in Java. An emphasis on systematic hyperparameter optimization methodologically strengthens this contribution. This research demonstrates that fair and valid model comparisons can only be achieved through careful tuning, ensuring the full potential of complex algorithms such as SVMs and Neural Networks can be accurately revealed in the geophysical domain.

Vol. 6, No. 4, August 2025, Page. 2811-2824
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5044

Furthermore, this study provides a new perspective on model selection. The discovery of performance convergence among leading nonlinear models shifts the discussion from searching for a single best model to a more diverse decision landscape. This suggests that practitioners can select the most appropriate model based not only on the highest accuracy but also on other trade-offs, such as training time efficiency, tailored to specific operational needs.

5. CONCLUSION

5.1. Conclusion

This study concludes that nonlinear models (Random Forest, SVM, and Neural Network) are significantly more accurate than Linear Regression for earthquake magnitude prediction in Java. Random Forest demonstrates the highest accuracy, but with a slight advantage over SVM and Neural Network, which perform very competitively. The main contribution of this study is the provision of a valid regional benchmark through a systematic hyperparameter optimization methodology, as well as a shift in model selection perspective towards considering the trade-off between accuracy and computational efficiency.

5.2. Recommendation

This study's limitations lie in its reliance on basic geospatial features. Therefore, further research is recommended to conduct feature engineering with more complex variables, such as distance to active faults, and to explore more modern ensemble algorithms such as XGBoost or LightGBM for potential accuracy improvements.

ACKNOWLEDGEMENT

Our thanks are extended to the Master of Computer Science Study Program at Universitas Amikom Purwokerto, for their support throughout the research process, which enabled the completion of this research in a timely and smooth manner.

REFERENCES

[1] A. Soehaimi, Y. Sopyan, Ma'mur, and F. Gustin, "Peta Patahan Aktif Indonesia," 2021.
[2] M. R. Purwanti, Z. K. Salsabila, and F. Liantoni, "Prediksi Gempa Bumi di Yogyakarta Berdasarkan Nilai Magnitudo, Kedalaman, dan Lokasi Gempa Menggunakan Naïve Bayes," *PETIR J. Pengkaj. dan Penerapan Tek. Inform.*, vol. 17, no. 1, pp. 122–132, 2024.
[3] N. Pramudito, "Mengenang Tragedi Gempa Jogja 2006 27 Mei, Diguncang Dahsyat Selama 57 Detik, Ribuan Orang Meninggal Dunia," *Radar Solo, Jawa Pos*, 2025.
[4] Y. C. A. Sanjaya and I. E. Pratiwi, "18 Tahun Silam Yogyakarta Diguncang Gempa M 5,9, Ribuan Orang Meninggal Dunia," *Kompas.com*, 2024.
[5] R. H. Permana, "Kilas Balik Duka Cianjur Diguncang Gempa Dashyat," *Detik News*, 2023.
[6] M. Rizky, "Gempa Dahsyat Cianjur, Jawa Barat Terbanyak Bencana 2022," *CNBC Indonesia*, 2022.
[7] T. Detikcom, "Gempa Hari Ini 7 Juli 2024 di Batang: Kekuatan, Jenis, dan Dampaknya," *detiknews*, 2024.
[8] D. K. Rizqi, "Penyebab Gempa M 4,6 di Batang, Ternyata Ada Aktivitas Sesar Aktif," *Radar Semarang*, 2024.
[9] I. Maulita and A. M. Wahid, "Prediksi Magnitudo Gempa Menggunakan Random Forest , Support Vector Regression , XGBoost , LightGBM , dan Multi-Layer Perceptron Berdasarkan Prediksi Magnitudo Gempa Menggunakan Random Forest , Support Vector Regression , XGBoost , LightGBM , dan Multi-La," *J. Pendidik. dan Teknol. Indones.*, vol. 4, no. 5, pp. 221–232, 2024.
[10] A. Fauzan and D. Ahmad, "Analisis Hasil Prediksi Magnitudo Gempa Di Wilayah Kota Padang

Vol. 6, No. 4, August 2025, Page. 2811-2824
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5044

Menggunakan Teknik Random Forest," *J. Lebesgue J. Ilm. Pendidik. Mat. Mat. dan Stat.*, vol. 4, no. 3, pp. 1569–1576, 2023.

[11] O. Somantri, "Prediksi Kekuatan Gempa Bumi Indonesia Berdasarkan Nilai Magnitudo Menggunakan Neural Network," in *Prosiding Seminar Nasional Informatika Bela Negara*, 2021, vol. 2, no. November, pp. 203–207.

[12] A. A. Nurhalizah, Y. Cahyana, and Rahmat, "Model Prediksi Kekuatan Gempa Dengan Menggunakan Algoritma Linear Regression Dan Support Vector Regression (Studi Kasus BMKG)," *Sci. Student J. Information, Technol. Sci.*, vol. V, no. 2, p. 41, 2024.

[13] O. Somantri, S. Purwaningrum, and Riyanto, "MODEL SUPPORT VECTOR MACHINE (SVM) BERDASARKAN PARAMETER WINDOWS UNTUK PREDIKSI KEKUATAN GEMPA BUMI," *JTT (Jurnal Teknol. Ter.*, vol. 8, no. 1, pp. 17–24, 2022.

[14] H. Tantyoko, D. K. Sari, and A. R. Wijaya, "Prediksi Potensial Gempa Bumi Indonesia Menggunakan Metode Random Forest Dan Feature Selection," *IDEALIS Indones. J. Inf. Syst.*, vol. 6, no. 2, pp. 83–89, 2023.

[15] F. E. Penalun, A. Hermawan, and D. Avianto, "Perbandingan Random Forest Regression dan Support Vector Regression Pada Prediksi Laju Penguapan," *J. Fasilkom*, vol. 13, no. 02, pp. 104–111, 2023.

[16] S. Chowdhury, A. K. Saha, and D. K. Das, "Hydroelectric Power Potentiality Analysis for the Future Aspect of Trends with R2 Score Estimation by XGBoost and Random Forest Regressor Time Series Models," *Procedia Comput. Sci.*, vol. 252, pp. 450–456, 2025.

[17] A. Edianto, G. Trencher, N. Manych, and K. Matsubae, "Forecasting coal power plant retirement ages and lock-in with random forest regression," *Patterns*, vol. 4, no. 7, 2023.

[18] M. Pangaribuan, J. J., And Lestari, "Perbandingan Metode Moving Average (Ma) Dan Neural Network Yang Berbasis Algoritma Backpropagation Dalam Prediksi Harga Saham," *J. Inf. Syst. Dev.*, vol. 5, no. Vol 5, No 1 (2020): Journal Information System Development (ISD), pp. 26–34, 2020.

[19] Y. P. Sugandhi, B. Warsito, and A. R. Hakim, "Prediksi Harga Saham Harian Menggunakan Cascade Forward Neural Network (CFNN) Dengan Particle Swarm Optimization (PSO)," *Stat. J. Theor. Stat. Its Appl.*, vol. 19, no. 2, pp. 71–82, 2019.

[20] D. Saputro and D. Swanjaya, "Analisa Prediksi Harga Saham Menggunakan Neural Network Dan Net Foreign Flow," *Gener. J.*, vol. 7, no. 2, pp. 96–104, 2023.

[21] P. Sulardi, T. Hendro, and F. R. Umbara, "PREDIKSI KEBUTUHAN OBAT MENGGUNAKAN REGRESI LINIER," in *Prosiding SNATIF*, 2017, vol. 4, pp. 57–62.

[22] J. M. Sangeetha and K. J. Alfia, "Financial stock market forecast using evaluated linear regression based machine learning technique," *Meas. Sensors*, vol. 31, no. December 2023, p. 100950, 2024.

[23] M. R. Athallah and A. F. Rozi, "Implementasi Data Mining Untuk Prediksi Peramalan Penjualan Produk Hj Karpet Menggunakan Metode Linear Regression," *J. Sains dan Teknol.*, vol. 2, no. 3, pp. 180–187, 2023.

[24] R. J. Kuo and T. H. Chiu, "Hybrid of jellyfish and particle swarm optimization algorithm-based support vector machine for stock market trend prediction," *Appl. Soft Comput.*, vol. 154, no. January, p. 111394, 2024.

[25] Nurul Salsabila Syam *et al.*, "Model Support Vector Machine untuk Prediksi pada Penggunaan Energi Listrik di Rumah Hemat Energi," *J. Inform.*, vol. 1, no. 2, pp. 56–59, 2022.

[26] D. Tomar and S. Agarwal, "Twin Support Vector Machine: A review from 2007 to 2014," *Egypt. Informatics J.*, vol. 16, no. 1, pp. 55–69, 2015.

[27] B. G. Marcot and A. M. Hanea, "What is an optimal value of k in k-fold cross-validation in discrete Bayesian network analysis?," *Comput. Stat.*, vol. 36, no. 3, pp. 2009–2031, 2021.

[28] N. Selle, N. Yudistira, and C. Dewi, "Perbandingan Prediksi Penggunaan Listrik dengan Menggunakan Metode Long Short Term Memory (LSTM) dan Recurrent Neural Network (RNN)," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 9, no. 1, pp. 155–162, 2022.