IPTEK The Journal of Technology and Science, 37(1), 2026 (e/pISSN:2088-2033, 0853-4098)

DOI: 10.12962/j20882033.v37i1.9266 | Received 25 Januari, 2025; Revised 17 November 2025; Accepted 17 November 2025

ORIGINAL RESEARCH

Comparison of Supervised Machine Learning Methods for Predicting Stunting Prevalence in West Java Province

Niken Riyanti* | Roy Rudolf Huizen | Evi Triandini

Departement of Magister Information System, Institut Teknologi dna Bisnis STIKOM Bali, Denpasar, 80234, Indonesia

Correspondence

*Email: : 232011016@stikom-bali.ac.id

Present Address

Jl. Raya Puputan No.86, Dangin Puri Klod, Kec. Denpasar Timur, Kota Denpasar, Bali 80234

Abstract

Stunting is a condition affecting children's growth and development due to chronic malnutrition and recurring infections, characterized by a height below -2 standard deviations on the WHO growth curve. It remains a major global nutritional issue, with 149.2 million children (22%) affected worldwide in 2020. In the same year, 276,069 children in West Java (24.5%) were classified as stunted. Addressing this issue can involve predictive approaches, such as supervised machine learning. The methods compared include Polynomial Regression (PR), Support Vector Regression (SVR), and Linear Regression (LR) in four treatments. The models analyzed include PR, SVR, LR, PR-XGB (Polynomial Regression with XGBoost), SVR-SGB (Support Vector Regression with Stochastic Gradient Boosting), LR-XGB (Linear Regression with XGBoost), PR-XGB2, SVR-SGB2, LR-XGB2 (the previous models with double boosting), PR-XGB2-Opt, SVR-SGB2-Opt, and LR-XGB2-Opt (double boosting with hyperparameter optimization). The novelty of this study lies in improving the performance of the models through a double-boosting technique using Extreme Gradient Boosting (XGBoost) with hyperparameter optimization via GridSearchCV. Among models without boosting, LR achieved the best performance with MSE 0.018217, MAE 0.130036, MAPE 0.314071; with single boosting, SVR-XGB performed best with MSE 0.031485, MAE 0.162925, MAPE 0.344510; with double boosting and hyperparameter optimization, both models LR-XGB2 and LR-XGB2-Opt maintained the best performance with the same value, MSE 0.016474, MAE 0.124677, MAPE 0.309293. These results suggest that double boosting with proper tuning significantly enhances model performance in predicting stunting prevalence.

KEYWORDS:

GridSearchCV; Linear Regression; Polynomial Regression; Stunting; Supervised Machine Learning; Support Vector Regression; XGBoost.

1 | INTRODUCTION

Stunting is a growth and development disorder in children resulting from chronic malnutrition and recurrent infections, characterized by a height below the standard. According to the WHO, a child is classified as stunted if their height is less than -2 standard deviations from the WHO growth curve[1] is also the most significant nutritional problem worldwide, with the prevalence of stunted children reaching 149.2 million (22%) in 2020, with Southeast Asia accounting for 15.3 million (27.4%)[2]. In Indonesia, the prevalence of stunting in 2022 was 21.6%, representing a 2.8% decrease from 2021. However, this figure still does not meet the WHO's threshold for stunting prevalence, which is 20% of the population[3].

West Java Province is one of the priority provinces for accelerating the reduction of stunting. This is due to the fact that West Java has the highest number of stunted children in Indonesia, with limited access to clean water being the primary contributing factor to stunting[4]. The incidence of stunting in West Java reached 276,069 children, or 24.5%, in 2020. Although there was a 4.3% decrease in 2022, stunting remains a significant challenge for the West Java provincial government[3]. The use of Machine Learning to predict stunting has been explored, but studies are still limited, leaving a significant research gap in applying Machine Learning approaches to stunting. For example, research using stunting prevalence data from East Java Province compared three Machine Learning models: SVR, RFR, and Linear Regression, with SVR identified as the best model for predicting stunting in East Java[5].

Another study compared various linear models from the scikit-learn library, including linear regression, Ridge Regression, Lasso Regression, Orthogonal Matching Pursuit, Tweedie Regressor, Polynomial Regression with and without a pipeline, SVR, K-Nearest Neighbors regressor, and MLPR. The evaluation revealed that Polynomial Regression had the lowest RMSE across three different variable schemes[6].

Previous studies have shown that linear regression models, although applied, have not been the most effective models. However, no research has yet incorporated boosting algorithms to enhance the performance of Linear Regression, particularly in West Java Province. Linear Regression can be utilized as a weak predictor, whose performance can be enhanced through boosting algorithms[7]. Additionally, combining methods can enhance a model's performance or even lead to the creation of new algorithms. In a previous study[8], Gradient Boosting and Nesterov's Accelerated Descent were combined to develop a new algorithm called AGB. By merging these two algorithms, AGB achieved performance comparable to Gradient Boosting while requiring significantly fewer components.

Boosting algorithms play a significant role in improving the performance of the models used. In research conducted to predict heart disease, several boosting algorithms, such as AdaBoost, GradientBoost, XGBoost, CatBoost, and Light GradientBoost, were compared with the proposed technique, Two-Level Boosting. The results indicated that the Two-Level Boosting technique achieved the highest accuracy, followed by XGBoost and GradientBoost[9]. However, applying double boosting introduces potential challenges, including the risk of overfitting and increased computational time. To address these issues, hyperparameter optimization is proposed using the GridSearchCV algorithm. GridSearchCV is capable of identifying the optimal set of hyperparameters by exhaustively testing various parameter combinations, which are then applied to XGBoost to obtain the most effective version of the model.

Based on the explanation above, this study focuses on optimizing the performance of linear regression in predicting stunting prevalence in West Java by comparing three models under four treatments: Polynomial Regression, Support Vector Regression, and linear regression with single boosting, double boosting, and double boosting with hyperparameter optimization. The evaluation results will reveal which model demonstrates the best performance based on evaluation metrics such as MAE, MSE, MAPE, and the computational time required for modeling each model, including the proposed model.

2 | PREVIOUS RESEARCHES

Several previous studies have applied Machine Learning approaches to predict stunting prevalence in specific regions. In addition to Machine Learning, other approaches have also been explored, such as Decision Support Systems and application development, to address stunting-related issues.

Stunting case predictions in South Kalimantan were conducted by comparing linear models from the scikit-learn library in the Python programming language[6]. The models used included linear regression, Ridge Lasso, Orthogonal Matching Pursuit, Tweedie Regressor, Polynomial Regression with a pipeline, Polynomial Regression without a pipeline, Support Vector Regression, K-nearest neighbors regressor, and MLP Regression. These models were applied to a stunting prevalence dataset from 2013 to 2020, with the pipeline achieving the smallest MAPE score of 0.00 using Polynomial Regression.

In another study conducted in East Java Province[5], stunting datasets were modeled by comparing three machine learning models: Linear Regression, Support Vector Regression (SVR), and Random Forest Regression. The dataset used was a transformed combination of 20 stunting-related datasets from 2019. In this study, SVR demonstrated the best performance, achieving MAE and MSE scores of 0.91 and 1.30, respectively.

Research on stunting continues to be conducted in Indonesia due to the high prevalence of stunting cases, which remain above the standards set by the WHO. To reduce the prevalence of stunting, the government has identified priority areas for stunting intervention in the country. One study aimed to identify the variables influencing the prioritization of stunting intervention areas in Indonesia. The Geographically Weighted Logistic Regression (GWLR) model[10], an extension of logistic regression that incorporates spatial effects, was employed. The study found that the best GWLR model for stunting cases in Indonesia utilized the Fixed Bisquare kernel weighting function, achieving an AIC value of 622.806477 and a classification accuracy of 0.7257.

Another study analyzing the impact of independent variables on stunting used the Geographically Weighted Regression (GWR) model[11]. Based on spatial heterogeneity tests, stunting rates among children vary across regions. The study then categorized regions into several groups based on significant variables. The first group consisted of provinces with no explanatory variables influencing stunting rates in children. The second group highlighted Low Birth Weight (LBW) as a significant influencing variable, while the third group identified Exclusive Breastfeeding (EBF) and Low Birth Weight (LBW) as key factors affecting stunting rates in children.

Stunting case modeling in children in East Nusa Tenggara Province has also been conducted using the Multivariate Adaptive Regression Splines (MARS) method[12]. The best MARS model consisted of three basis functions with two predictor variables: the percentage of pregnant women at risk of chronic energy deficiency and the percentage of babies with low birth weight. After classification, the model achieved a classification accuracy of 77.27% with an error rate of 22.72%. In addition to machine learning approaches, Expert Systems have also been used to study stunting cases in Indonesia. The causes of stunting are influenced by poor caregiving practices, particularly during the first 1,000 days of life (HPK). One study implemented a Decision Support System using an expert system approach to intervene in stunting management by providing solutions through the Certainty Factor method[13]. The Expert System was developed as a website-based platform using the PHP programming language. The results of the tests showed accelerated stunting reduction due to the accurate percentages and solutions provided for handling and improving government oversight efforts in combating stunting. The system also included intervention percentages that were sensitive and specific to pregnant women, breastfeeding mothers, and adolescent girls, with 100% functionality operating effectively.

In addition to using the Certainty Factor, the Fuzzy Mamdani method has also been applied[14]. The utilization of health applications that make it easier for users to access information can be used to identify stunted children by selecting their symptoms. The developed application can perform early detection of growth and development disorders in children using the Fuzzy Mamdani method. Fuzzy Mamdani is used to categorize stunting criteria for children in the "gray" area. The results of this study showed that the detection accuracy using the Fuzzy Mamdani method was 80.87% compared to expert diagnoses.

3 | METHOD

The research focuses on comparing algorithms to predict the prevalence of stunting in West Java Province using a proposed model, namely LR-XGB2-Opt. The research stages are outlined in a research design presented as an experimental design in Figure 1 .

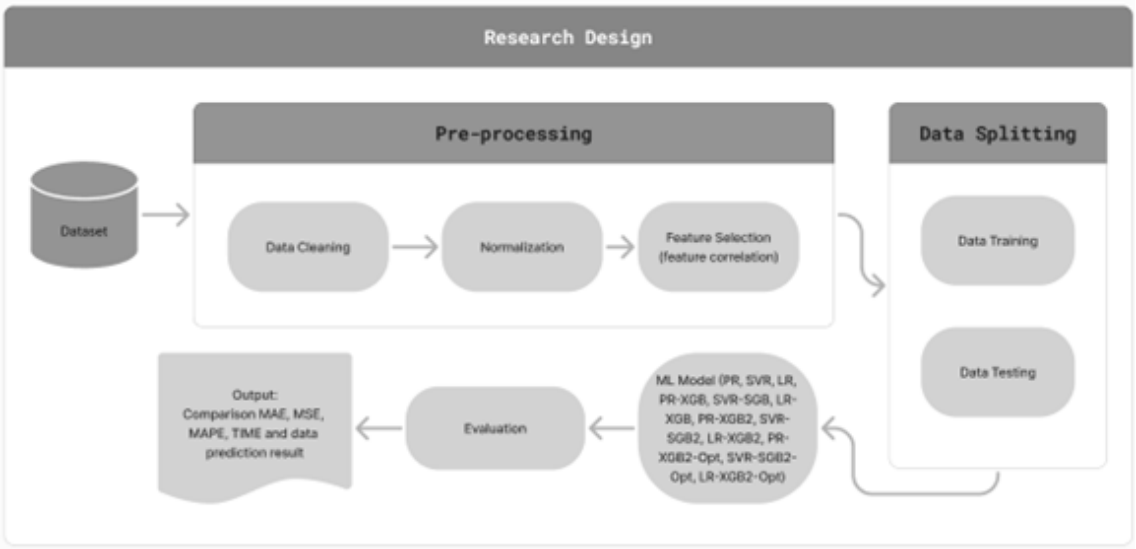

FIGURE 1 Research Design

3.1 | Dataset

This study utilizes an official dataset provided by the West Java Provincial Government, which was collected and publicly accessed through the Open Data Jabar website. The dataset contains data on the prevalence of stunting in cities and regencies within West Java Province from 2014 to 2023, in a comma-separated values (CSV) file format.

The dataset used consists of 29 features with a total of 290 records, including the following features: year,

`p_kab_bogor`, `p_kab_sukabumi`, `p_kab_cianjur`,
`p_kab_bandung`, `p_kab_garut`, `p_kab_tasik`,
`p_kab_ciamis`, `p_kab_kuningan`, `p_kab_cirebon`,
`p_kab_majalengka`, `p_kab_sumedang`, `p_kab_indramayu`,
`p_kab_subang`, `p_kab_purwakarta`, `p_kab_karawang`,
`p_kab_bekasi`, `p_kab_bandungbarat`, `p_kab_pangandaran`,
`p_kot_bogor`, `p_kot_sukabumi`, `p_kot_bandung`,
`p_kot_cirebon`, `p_kot_bekasi`, `p_kot_depok`,
`p_kot_cimahi`, `p_avg`. This dataset is referred to as *StuntingJabar*.

3.2 | Pre-processing Data

Pre-processing data is the stage in data analysis where raw data is prepared and cleaned before it is used for further analysis. This study employs three data pre-processing stages, including data cleaning, feature selection, and normalization.

The StuntingJabar dataset contains 2 missing values in 2 regions for the same year To address the missing values and optimize the performance of the machine learning model, imputation was performed using the median method.

After addressing missing values, the dataset underwent normalization to standardize its scale. The MinMax Scaler was used to adjust feature values, typically to a range between 0 and 1, balancing both variables (x and y) for modeling. The scaling formula is shown in Equation 1.

$$X_{\text{minmax}} = \frac{X\text{-}X_{\text{min}}}{X_{\text{max}}\text{-}X_{\text{min}}} \tag{1}$$

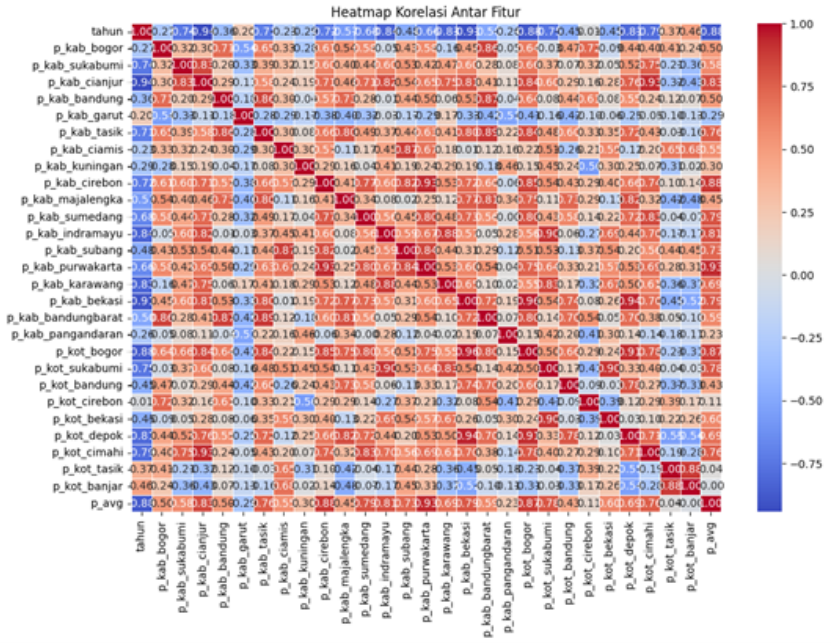

FIGURE 2 Heatmap correlation between features in the StuntingJabar dataset

Following normalization, the dataset proceeded to the feature selection stage, where feature correlations with a threshold of greater than 0.2 or less than -0.2 were identified. The correlations between features are visualized in a heatmap diagram, as shown in Figure 2 .

3.3 | Data Splitting

The processed data is split into two parts, 80% for training and 20% for testing. The training data is used to train the machine learning model, enabling it to make accurate predictions. Meanwhile, the testing data is used to check how well the algorithm performs in making predictions.

3.4 | Supervised Machine Learning Model

The dataset, after pre-processing and data splitting, is ready to be modeled using various Supervised Machine Learning algorithms. These include Polynomial Regression, Support Vector Regression, Linear Regression, and their respective combinations and optimizations with XGBoost. All models will be implemented in Python.

1. Polynomial Regression

   Polynomial Regression is a statistical method in Data Mining aimed at observing and facilitating the relationship between variables, particularly in functional forms. Polynomial Regression is a modified version of the multiple linear regression model[15]. In general, Polynomial Regression is expressed as follows:

$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n \tag{2}$$

2. Support Vector Regression

   The Support Vector Regression (SVR) model is a variation of the Support Vector Machine (SVM), used for regression with continuous outputs.[5]. SVR fits a non-linear relationship between input and output data, formulated as follows[16]:

$$f(x, w) = w.\Phi(x) + b = (w, \Phi) + b \tag{3}$$

TABLE 1 . Hyperparameter values range

No.	Hyperparameter	Value Range
1	Learning_rate	0,1 – 2,0
2	N_estimators	1 – 500
3	Max_depth	1 – Number of dataset attribute

Where w is the weight vector, $\Phi(x)$ is the nonlinear mapping that transforms the input vector x into a higher-dimensional space, b is the bias, and $w \cdot \Phi(x)$ is the linear product of w and Φ.

3. Linear Regression

   Linear Regression is a Machine Learning model designed to establish a quantitative relationship between multiple variables. Linear Regression is simple, fast, and interpretable; however, it may not effectively capture non-linear and complex relationships within the data[17]. To describe the relationship between two variables, Linear Regression uses a straight line for visualization[18]. The mathematical notation for Linear Regression is as follows:

$$\hat{y} = w_0 x_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n \tag{4}$$

4. Extreme Gradient Boosting

   XGBoost, short for Extreme Gradient Boosting, is a gradient boosting algorithm renowned for its exceptional performance across various machine learning tasks. This algorithm works by iteratively combining simple models to create a more robust model. XGBoost optimizes a user-defined loss function by incrementally adding Decision Trees to the model. Each tree aims to reduce the residual errors from previous iterations, resulting in a series of trees that progressively improve prediction accuracy[17]. To better understand how XGBoost models are constructed, it is helpful to review the mathematical notation of a related algorithm, Linear Regression:

$$\hat{y}_i = \phi(x_i) + \sum_{k=1}^{K} f_k(x_i) \tag{5}$$

   Where $\hat{y}_i$ is the predicted value for the i-th sample, $\phi(x_i)$ represents the global bias, K is the total number of trees, and $f_k(x_i)$ is the prediction from the k-th tree for the i-th sample.

5. GridSearchCV

   GridSearchCV is a hyperparameter optimization technique that systematically searches for the best combination of parameters within a predefined grid to enhance model performance. This method is widely adopted in machine learning research due to its effectiveness in fine-tuning models and improving prediction accuracy[19].

   Most of the regularization and model architecture in XGBoost is determined by hyperparameters. The key hyperparameters include the learning rate, the number of base learners in the ensemble, and the maximum depth of the base learners. In Python, these are referred to as `learning_rate`, `n_estimators`, and `max_depth`[20]. The range of hyperparameter values is shown in Table 1 .

3.5 | Evaluation

The evaluation metrics used to measure the performance of the models in this study are as follows:

1. Mean Squared Error (MSE) measures the average squared difference between the predicted and actual values and serves as an indicator of the accuracy of the predictions[17]. The smaller the MSE value, the more accurate the model's predictions. MSE can be mathematically defined as follows:

TABLE 2 Dataset Stunting Jabar after pre-processing phase

Tahun	p_kab_bogor	p_kab_sukabumi	p_kab_cianjur	p_kab_bandung	p_kab_garut	p_kab_tasik	p_kab_ciamis	p_kab_kuningan	p_kab_cirebon	p_kab_majalengka	p_kab_sumedang	p_kab_indramayu	p_kab_subang	p_kab_purwakarta	p_kab_karawang	p_kab_bekasi	p_kab_bandungbarat	p_kab_pangandaran	p_kot_bogor	p_kot_sukabumi	p_kot_bandung	p_kot_bekasi	p_kot_depok	p_kot_cimahi	p_avg
2014	0.7661	0.8506	0.9497	1	0.2265	1	0.28	0.7096	0.8711	1	0.4255	0.5968	0.5835	0.5853	0.7118	1	1	0.2906	1	0.6192	0.8625	0.3849	1	0.6119	0.9398
2015	0.6064	0.5432	1	0.2301	0.2798	0.5460	0.0106	0.8251	0.8195	0.5006	1	0.8980	0.3290	0.7781	0.7292	0.8575	0.6160	0.3401	0.8570	0.6501	1	0.3192	0.9916	1	1
⋮													⋮												
2023	0.0491	0.2667	0	0	0.4685	0	0.0329	0.8020	0.1967	0.0256	0	0	0.1722	0	0	0	0	0	0	0	0.2768	0	0	0.0533	0.2032

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \tilde{y}_i\right)^2 \tag{6}$$

2. Mean Absolute Error (MAE) is an evaluation metric used to measure the accuracy of forecasting/predicting models. The MAE value represents the average absolute error between the predicted results and the observed data[17]. MAE can be explained by the following formula

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|A_i - f_i\right| \tag{7}$$

3. Mean Absolute Percentage Error (MAPE) is an extension of MAE that meets the criteria of reliability, ease of interpretation, and clarity of presentation[21]. MAPE can be defined as follows:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{A_i - f_i}{A_i}\right| \times 100\% \tag{8}$$

4 | RESULT AND DISCUSSION

This chapter presents the analysis and discussion of the machine learning models used in this study, including Polynomial Regression (PR), Support Vector Regression (SVR), Linear Regression (LR), Polynomial Regression with Extreme Gradient Boosting (PR-XGB), Support Vector Regression with Stochastic Gradient Boosting (SVR-SGB), Linear Regression with Extreme Gradient Boosting (LR-XGB), Polynomial Regression with a second implementation of Extreme Gradient Boosting (PR-XGB2), Support Vector Regression with a second implementation of Stochastic Gradient Boosting (SVR-SGB2), Linear Regression with a second implementation of Extreme Gradient Boosting (LR-XGB2), and their optimized versions: Polynomial Regression with optimized Extreme Gradient Boosting (PR-XGB2-Opt), Support Vector Regression with optimized Stochastic Gradient Boosting (SVR-SGB2-Opt), and Linear Regression with optimized Extreme Gradient Boosting (LR-XGB2-Opt). These models were applied to predict stunting prevalence in West Java Province based on the time series dataset processed through data pre-processing steps. These models are applied to a dataset that has undergone data preprocessing. The dataset after pre-processing is shown in Table 2 .

4.1 | Polynomial Regression Model Implementation

The implementation of the Polynomial Regression model was carried out using the PolynomialFeatures() method from the sklearn library in the Python programming language. The PR model was implemented in four variations, which are:

1. Polynomial Regression (PR)

Polynomial Regression (PR) modeling began with data preprocessing, including filling missing values with the median (median imputation) and scaling data to a range between 0 and 1 using MinMaxScaler(), a normalization tool. Features with a correlation coefficient greater than 0.2 were selected. The selected data were then expanded to include combinations of features through PolynomialFeatures(), which creates new polynomial terms, and fitted using LinearRegression(), a standard linear modeling technique. Predictions were generated and evaluated using mean squared error

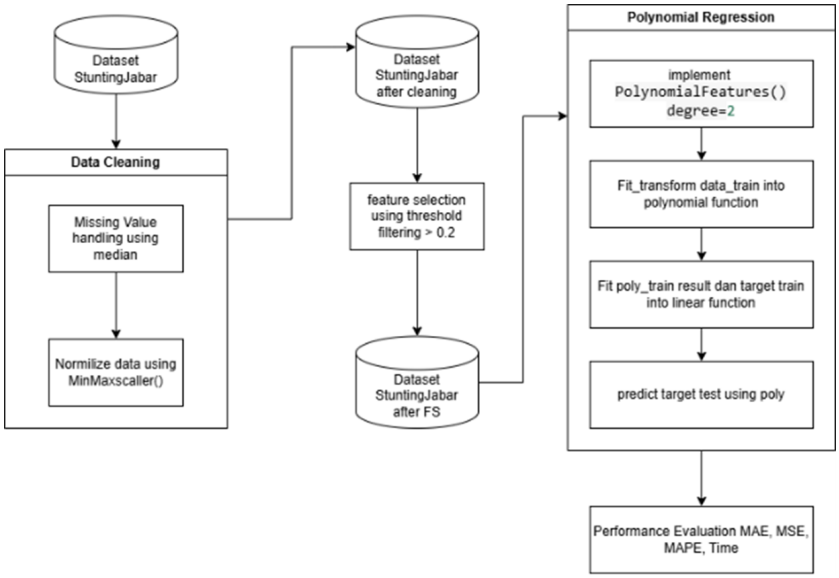

FIGURE 3 Polynomial Regression Modeling Flow

(MSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). The entire PR model implementation process followed the workflow shown in Figure 3 .

2. Polynomial Regression with XGBoost (PR-XGB)

The PR-XGB modeling process begins with data preprocessing. The selected dataset is transformed using PolynomialFeatures() and then fitted with LinearRegression() to predict both training and testing targets. Residuals from the training set are calculated and filtered using a threshold of 0.4, along with the corresponding training data. The filtered residuals and data are then modeled using XGBRegressor() to generate final predictions. The outputs of the PR and XGB models are combined to compute the final evaluation metrics. The implementation process of the PR-XGB model was carried out following the workflow illustrated in Figure 4 .

3. Polynomial Regression with Double XGBoost (PR-XGB2)

The PR-XGB2 modeling framework begins with a data pre-processing phase. The selected features are transformed using PolynomialFeatures() and fitted with LinearRegression() to generate initial predictions. Residuals from the training set are filtered using a 0.4 threshold and modeled with XGBRegressor() (XGB-1). The difference between the first residuals and XGB-1 predictions yields a second residual, which is again filtered and modeled using another XGBRegressor() (XGB-2). The final prediction is obtained by summing the outputs from PR, XGB-1, and XGB-2, and evaluation metrics are computed accordingly. The implementation process of the PR-XGB2 model was carried out following the workflow illustrated in Figure 5 .

4. Polynomial Regression with Double XGBoost Optimation (PR-XGB2-Opt)

The PR-XGB2-Opt modeling process begins with a data preprocessing phase, followed by transformation using PolynomialFeatures() and modeling with LinearRegression(). The initial predictions are used to compute residuals from the training set, which are then filtered using a threshold of 0.4. Both the filtered residuals and corresponding training data are then used in a two-level XGBoost model optimized with GridSearchCV. The hyperparameter tuning grid is shown in Table 3 .

GridSearchCV selects the optimal configuration based on 5-fold cross-validation and negative MSE as the scoring metric. The best parameters are utilized in both the first- and second-level XGBoost regressors. The second residual is computed by subtracting. The implementation process of the PR-XGB2-Opt model was carried out following the workflow illustrated in Figure 6 .

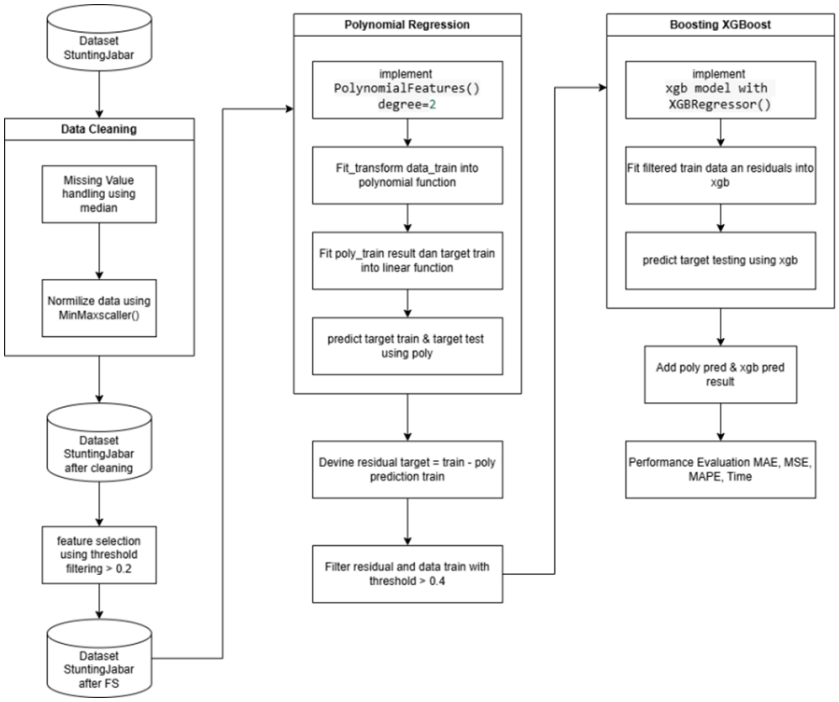

FIGURE 4 PR-XGB Modeling Flow

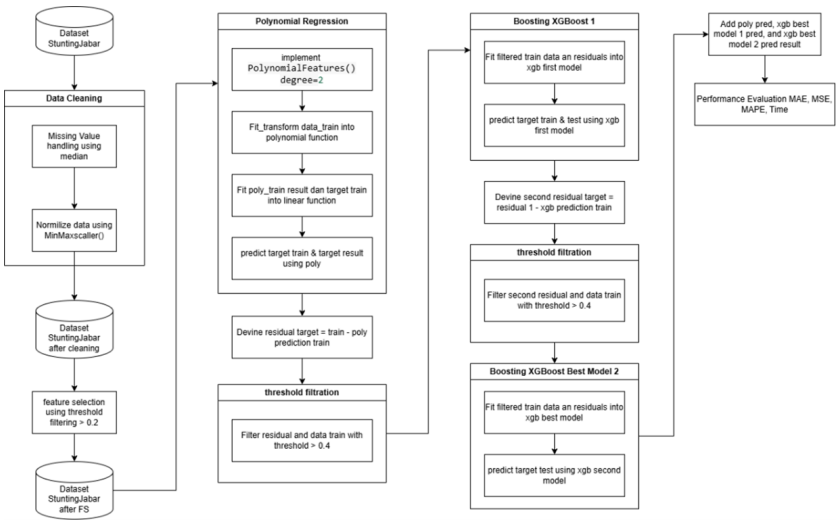

FIGURE 5 PR-XGB2 Modeling Flow

4.2 | Support Vector Regression Model Implementation

The implementation of the Support Vector Regression (SVR) model was carried out using the SVR() method from the sklearn library in the Python programming language. The SVR model was implemented in four variations, which are:

1. Support Vector Regression (SVR)

   Support Vector Regression (SVR) Modeling begins with a data cleaning process, which includes handling missing values using the median and normalizing the data using MinMaxScaler() in Python. Feature selection is then applied using a correlation threshold greater than 0.2 to retain the most relevant features. The selected dataset is modeled using the SVR

TABLE 3 GridSearchCV Hyperparameter result

No.	Hyperparameter	Value Range
1	Learning_rate	0.01, 0.1, 0.2
2	N_estimators	100, 200, 300
3	Max_depth	3, 5, 7
4	Subsample	0.8, 1
5	Colsample_bytree	0.8, 1

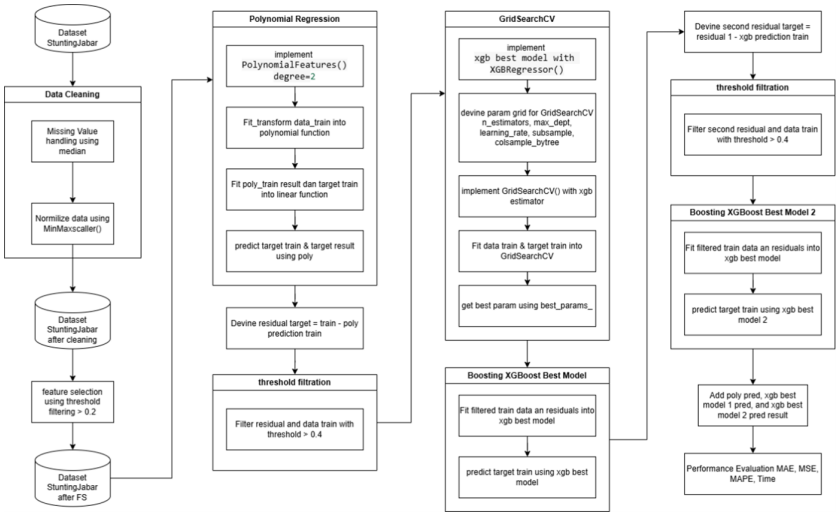

FIGURE 6 PR-XGB2-Opt Modeling Flow

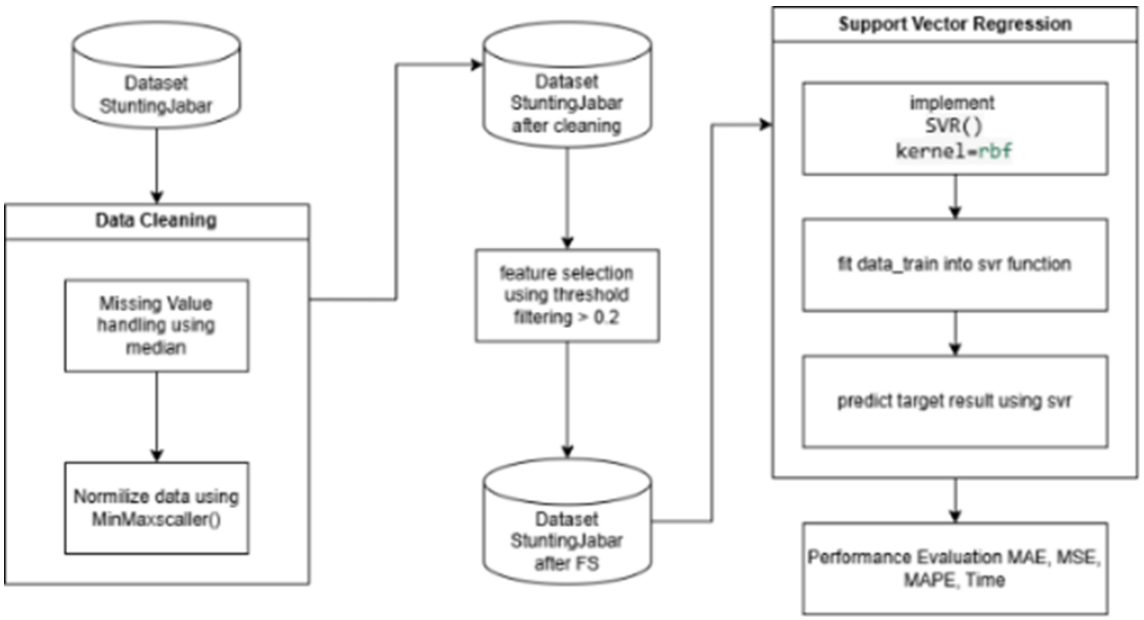

FIGURE 7 SVR Modeling Flow

method. The model is then used to predict the target variable, and its performance is evaluated using MSE, MAE, and MAPE metrics. The SVR modeling flow is shown in Figure 7 .

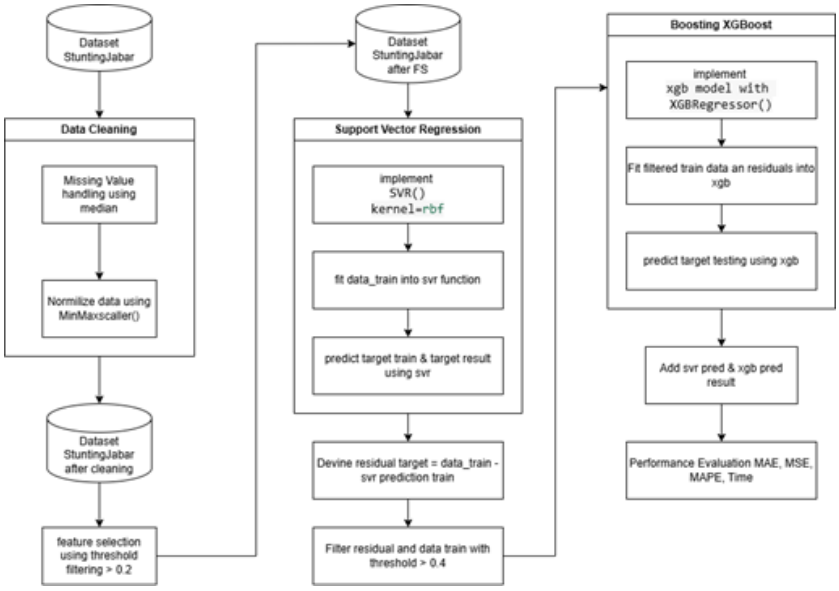

FIGURE 8 SVR-XGB Modeling Flow

2. Support Vector Regression with XGBoost (SVR-XGB)

   SVR-XGB Modeling begins with the data pre-processing phase. The selected dataset is first modeled using the SVR method to generate predictions for both training and testing data.

   Residuals are calculated by subtracting the SVR model's predictions from the actual training targets. These residuals are then filtered by selecting only those exceeding a threshold of 0.4, and the training data is filtered correspondingly. The filtered residuals and corresponding training data are then used to train an XGBRegressor(), which generates predictions on the testing set. Finally, the predictions from the SVR and XGBoost models are combined to compute the model's evaluation metrics. The SVR-XGB model flow is shown in Figure 8 .

3. Support Vector Regression with Double XGBoost (SVR-XGB2)

   SVR-XGB2 Modeling involves preprocessing through data pre-processing. The selected data is first modeled using SVR to obtain initial predictions and residuals. Residuals exceeding a threshold of 0.4 are filtered and modeled with XGBoost. A second residual is computed from this output, filtered again, and modeled with a second XGBoost. Final predictions are obtained by combining outputs from SVR, XGB-1, and XGB-2 for evaluation. The SVR-XGB2 modeling flow is shown in Figure 9 .

4. Support Vector Regression with Double XGBoost Optimation (SVR-XGB2-Opt)

   SVR-XGB2-Opt Modeling begins with data preprocessing. Next, the selected data is modeled using Support Vector Regression (SVR) to generate predictions and calculate residuals. Then, data points where the residuals and corresponding training data exceed a threshold of 0.4 are filtered and modeled using Extreme Gradient Boosting (XGBoost, here referred to as XGB-1).

   XGB-1 is optimized using GridSearchCV, a method that searches for the best hyperparameters for a model using cross-validation. The optimal hyperparameter for this model is shown in Table 3 . The model is trained on the filtered data and predicts outcomes for both training and testing datasets. A second set of residuals is calculated by subtracting XGB-1 predictions from the initial residuals; these are again filtered and modeled with a second XGBoost model (XGB-2). Final predictions are obtained by summing the outputs from SVR, XGB-1, and XGB-2, and evaluated using MSE, MAE, and MAPE metrics. SVR-XGB2-Opt modeling flow is shown in Figure 10 .

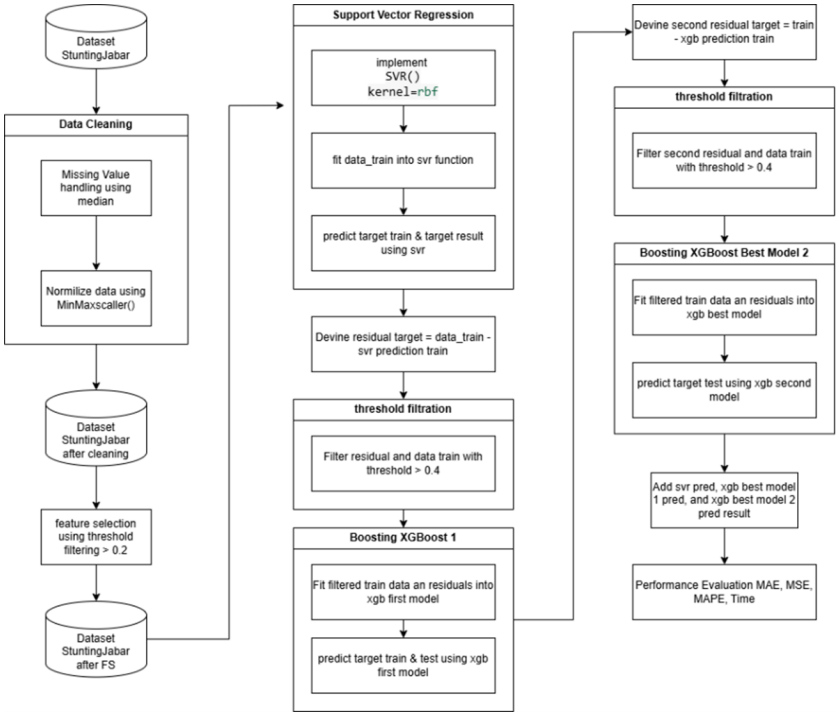

FIGURE 9 SVR-XGB2 Modeling Flow

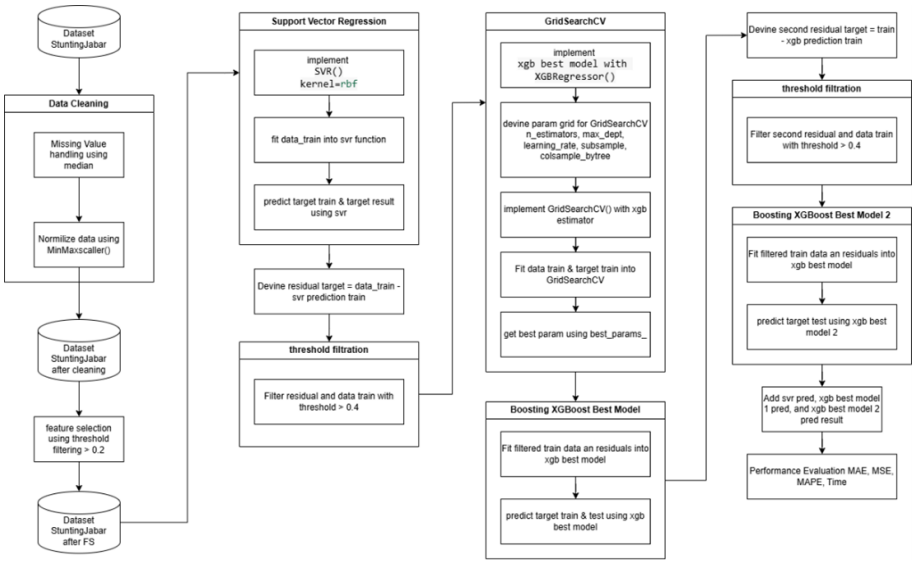

FIGURE 10 SVR-XGB2-Opt Modeling Flow

4.3 | Linear Regression Model Implementation

The implementation of the linear regression model was carried out using the LinearRegression() method from the scikit-learn library in the Python programming language. The Linear Regression model was implemented in four variations as follows:

1. Linear Regression (LR)

   Linear Regression (LR) Modeling begins with data cleaning, including median imputation for missing values and MinMaxScaler() normalization. Feature selection is performed using a correlation threshold of greater than 0.2 to retain

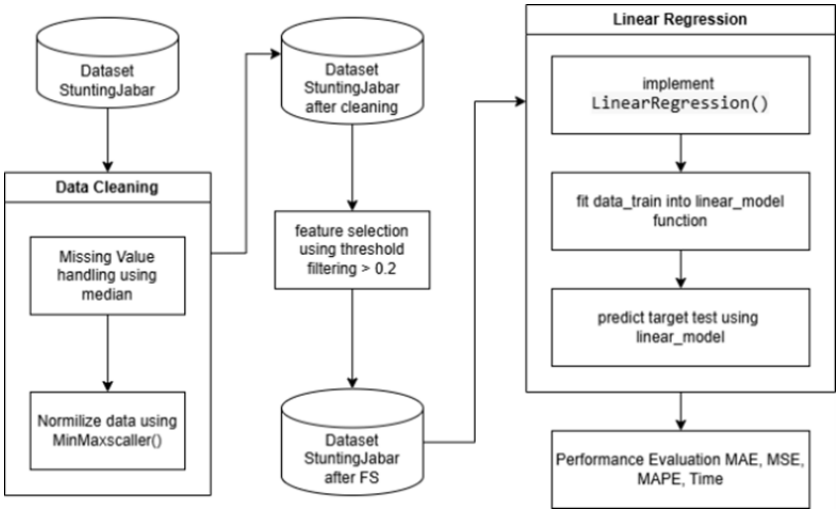

FIGURE 11 LR Modeling Flow

relevant features. The selected dataset is then modeled using LinearRegression() in Python. The model generates predictions for the target variable and is evaluated using standard performance metrics. LR model flow is shown in Figure 11 .

2. Linear Regression with XGBoost (LR-XGB)

   LR-XGB Modeling begins with the data-preprocessing phase. Feature selection is applied using a correlation threshold of greater than 0.2, meaning only features with at least moderate correlation to the target are included. The selected dataset is first modeled using LinearRegression(), a method that predicts the target by fitting a linear relationship to the data. Residuals, which are the differences between actual and predicted training values, are calculated and filtered using a threshold of 0.4, along with the corresponding training data.

   These filtered residuals and data are then modeled using XGBRegressor(), which is a machine learning algorithm based on gradient boosted decision trees, to predict the test set. Final predictions from both the LR and XGB models are combined to evaluate the model's performance. The LR-XGB modeling flow is shown in Figure 12 .

3. Linear Regression with Double XGBoost (LR-XGB2)

   LR–XGB2 modeling starts with the data pre-processing phase. The selected data is modeled with LinearRegression(), and residuals from the training set are computed. Residuals above 0.4, along with matching training data, are modeled using XGBRegressor() (XGB-1). The second residual, derived from the difference between the first residual and XGB-1 predictions, is again filtered with a 0.4 threshold and modeled using XGBRegressor() (XGB-2). Final predictions are the sum of outputs from LR, XGB-1, and XGB-2. The LR-XGB2 modeling flow is shown in Figure13 .

4. Linear Regression with Double XGBoost Optimation (LR-XGB2-Opt)

   The LR-XGB2-Opt modeling begins with the data pre-processing phase. The selected dataset is modeled using Linear-Regression() to generate predictions and calculate residuals.Residuals greater than a threshold of 0.4 are filtered along with the corresponding training data, then modeled using XGBRegressor() (XGB-1). The XGBoost model is optimized using GridSearchCV with parameter tuning, and the best parameters are then used to fit the training data and the first residuals. The hyperparameter results from the GridSearchCV work are shown in Table 3 .

   The second residual is calculated by subtracting the XGB-1 prediction from the first residual, then filtered again using a 0.4 threshold. The filtered data is modeled using another XGBRegressor() (XGB-2). The final model prediction is obtained by summing the outputs of LR, XGB-1, and XGB-2 for evaluation. LR-XGB2-Opt modeling flow is shown in Figure 14 .

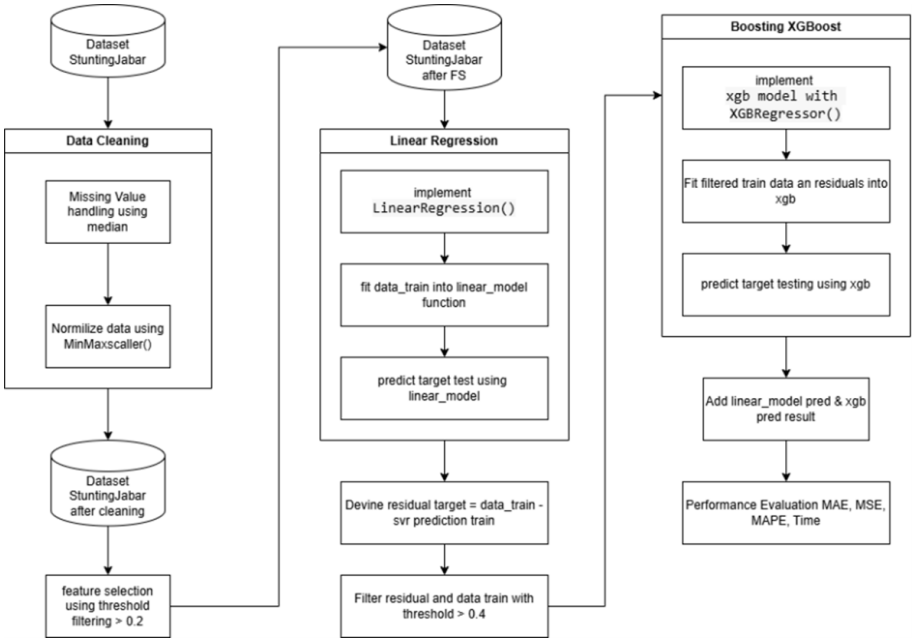

FIGURE 12 LR-XGB Modeling Flow

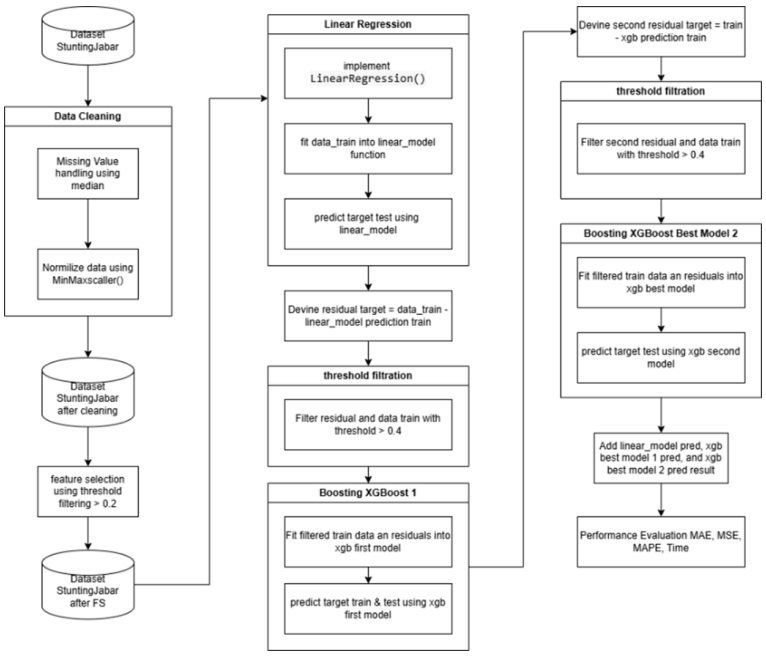

FIGURE 13 LR-XGB2 Modeling Flow

4.4 | Comparison of Model Performance

Figures 15 16 to 17 show the data predictions from each model and compare them with the actual data from the StuntingJabar dataset using a line chart. The performance of evaluation metrics for all models is shown in Table 5. It also displays the training data, actual data, and predicted data generated by each model. Additionally, comparison charts of the metrics for each model's performance are presented in Figure 18 .

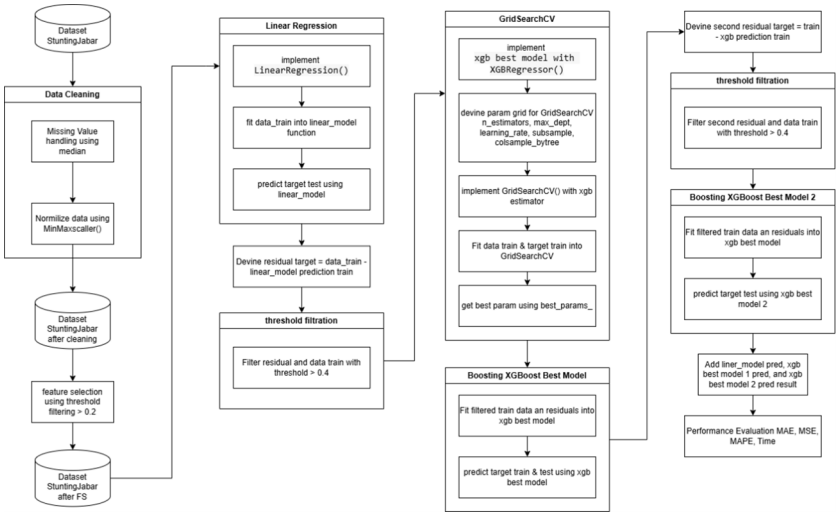

FIGURE 14 LR-XGB2-Opt

TABLE 4 The comparison results of supervised machine learning models in predicting stunting prevalence in West Java Province

No	Model	Data Learn 2022	Data Prediksi 2023	Data Aktual 2023	MSE	MAE	MAPE	TIME
1.	PR		0,109348		0,018217	0,130036	0,314071	0,032241
2.	SVR		0,284464		0,036668	0,169784	0,329022	0,004826
3.	LR		0,109051		0,016474	0,124677	0,309293	0,056037
4.	PR-XGB		0,109348		0,018217	0,130036	0,314071	0,134088
5.	SVR-XGB		0,358651		0,012738	0,088303	0,123611	0,043621
6.	LR-XGB	0,565565	0,109051	0,203231	0,016474	0,124677	0,309293	0,021002
7.	PR-XGB2		0,109348		0,018217	0,130036	0,314071	0,115138
8.	SVR-XGB2		0,359161		0,031485	0,162925	0,344510	0,045803
9.	LR-XGB2		0,109051		0,016474	0,124677	0,309293	0,036301
10.	PR-XGB2-Opt		0,109348		0,018217	0,130036	0,314071	0,046797
11.	SVR-XGB2-Opt		0,114665		0,035000	0,168939	0,342551	0,051905
12.	LR-XGB2-Opt		0,109051		0,016474	0,124677	0,309293	0,064754

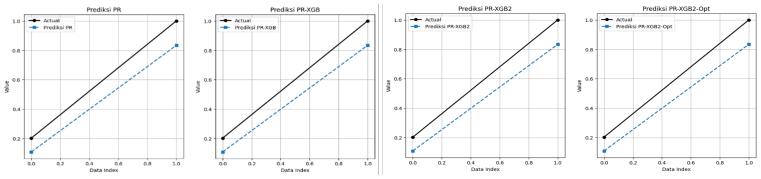

FIGURE 15 Line chart PR model data prediction result

4.5 | Discussion

These findings underscore the potential policy relevance of the LR-XGB2-Opt model. Its high level of accuracy and efficiency make it a reliable decision-support tool for the West Java Provincial Government in addressing stunted growth. By integrating such predictive models into routine health monitoring and planning, policymakers can identify high-risk areas earlier, allocate

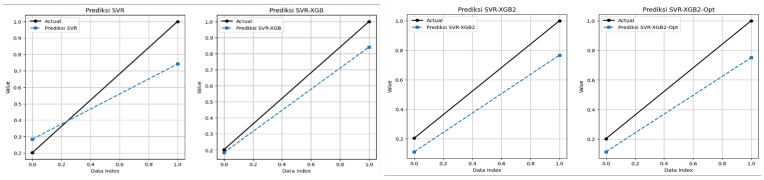

FIGURE 16 Line chart SVR model data prediction result

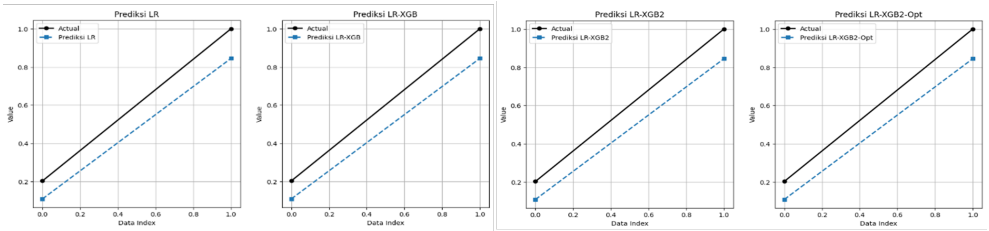

FIGURE 17 Line chart LR model data rediction result

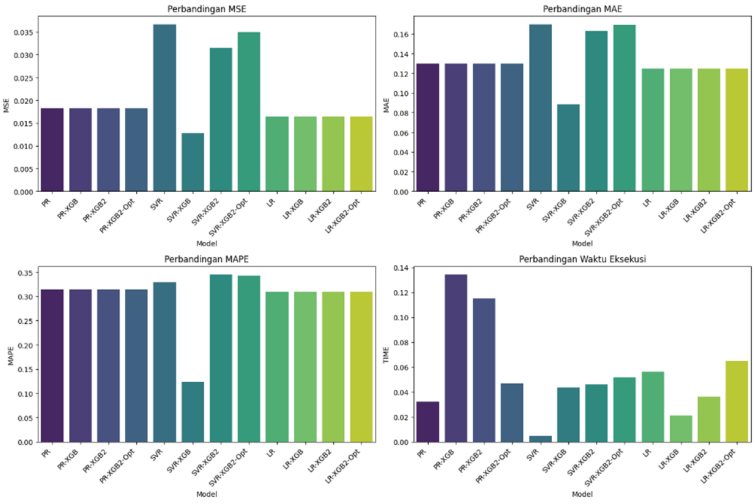

FIGURE 18 Box Chart Metric Comparison of All Models

resources more strategically, and design targeted interventions to address the determinants of stunting. This approach strengthens the government's capacity for proactive and preventive measures. It also helps improve overall child health outcomes, reduce long-term socioeconomic burdens, and support the achievement of regional and national health targets.

5 | CONCLUSION

5.1 | Conclusion

Based on the comparison results between the proposed models and the baseline models, several key findings can be drawn. The predicted average stunting prevalence for 2023, using the best-performing models from each approach, shows that the Linear Regression (LR), LR-XGB2, and LR-XGB2-Opt models consistently predict a value of 0.109051. In contrast, the SVR-XGB model yielded a slightly higher prediction of 0.185219. This indicates that the LR-based hybrid models were more consistent and aligned in estimating the stunting rate.

When evaluating the models using the StuntingJabar dataset, it was found that the proposed models, namely LR, SVR-XGB, and LR-XGB2, achieved the lowest values of MSE, MAE, and MAPE. Among these, the model that applied a two-stage boosting approach with hyperparameter tuning, LR-XGB2-Opt, outperformed the others, demonstrating its robustness and accuracy.

Furthermore, hyperparameter optimization across three models resulted in improvements, primarily in computational efficiency. For instance, the PR-XGB2-Opt model exhibited a 59.36% reduction in computation time compared to the non-optimized PR-XGB2 model, while maintaining similar levels of accuracy. Ultimately, the LR-XGB2-Opt model stood out as the best-performing model overall, achieving the lowest error metrics with an MSE of 0.016474, MAE of 0.124677, and MAPE of 0.309293, making it a highly reliable approach for predicting stunting prevalence.

5.2 | Future Works

Research on stunting prediction still has considerable room for development, particularly in the application of machine learning approaches. Future studies could explore the implementation of the proposed model on larger and more diverse datasets to better understand the effects of boosting. Additionally, testing other models with the same dataset could provide valuable insights into their effectiveness in predicting stunting prevalence in West Java.

References

1. Kementerian Kesehatan RI, Judul Laporan atau Artikel Tentang Stunting. Kementerian Kesehatan RI; 2024. Diakses pada 13 Februari 2026. https://www.kemkes.go.id.
2. Hisbullah R, Hasibuan MS. Pendekatan Bayes-HDSS dalam Menentukan Status Pantauan Gizi Balita. Jurnal Teknologi Informasi dan Ilmu Komputer 2023;10(5):1071–1082.
3. Pemerintah Provinsi Jawa Barat, Data Prevalensi Stunting Jawa Barat. Pemprov Jabar; 2023. Catatan tambahan jika ada.
4. Kementerian Sekretariat Negara Republik Indonesia, 218.286 Balita Stunting di Jabar, Akses Makanan Bergizi Salah Satu Penyebab; n.d. https://stunting.go.id/218-286-balita-stunting-di-jabar-akses-makanan-bergizi-salah-satu-penyebab/.
5. Haris MS, Khudori AN, Kusuma WT. Perbandingan Metode Supervised Machine Learning untuk Prediksi Prevalensi Stunting di Provisi Jawa Timur. Jurnal Teknologi Informasi dan Ilmu Komputer 2022;9(7):1571.
6. Mambang, Marleny FD, Zulfadhilah M. Prediction of linear model on stunting prevalence with machine learning approach. Bulletin of Electrical Engineering and Informatics 2023;12(1):483–492.
7. Li W, Wang W, Huo W. RegBoost: a gradient boosted multivariate regression algorithm. International Journal of Crowd Science 2020;4(1):60–72.
8. Biau G, Cadre B, Rouvière L. Accelerated gradient boosting. Machine Learning 2019;108(6):971–992.
9. Dissanayake K, Johar MGM. Two-level boosting classifiers ensemble based on feature selection for heart disease prediction. Indonesian Journal of Electrical Engineering and Computer Science 2023;32(1):381–391.
10. Alam FK, Widyaningsih Y, Nurrohmah S. Geographically weighted logistic regression modeling on stunting cases in Indonesia. Journal of Physics: Conference Series 2021;1722(1).
11. Sari RP, Winanda RS. Pemodelan Stunting pada Balita di Indonesia Menggunakan Geographically Weighted Regression (GWR). Journal Of Mathematics UNP 2023;8(3):106–116.
12. Azizah DM, Permatasari EO. Modeling of toddler stunting in the province of east nusa tenggara using multivariate adaptive regression splines (mars) method. Journal of Physics: Conference Series 2020;1490(1).
13. Sutoyo IW, Widodo AP, Rochim AF. Decision support system for handling intervention on toddlers stunting cases in Indonesia using the certainty factor method. Journal of Physics: Conference Series 2021;1943(1).

14. Wanti LP, Somantri O, Prasetya NWA, Puspitasari L. Fuzzy expert system design for detecting stunting. Indonesian Journal of Electrical Engineering and Computer Science 2024;34(1):556–564.

15. Putranto BPD, Kholik MA, Nugroho MA. Polynomial Regression Method and Support Vector Machine Method for Predicting Disease Covid-19 in Indonesia. Journal of Intelligent Software Systems 2023;2(1):18.

16. Huang S, Tian L, Zhang J, Chai X, Wang H, Zhang H. Support Vector Regression Based on the Particle Swarm Optimization Algorithm for Tight Oil Recovery Prediction. ACS Omega 2021;6(47):32142–32150.

17. Tricha A, Moussaid L. Evaluating machine learning models for precipitation prediction in Casablanca City. Indonesian Journal of Electrical Engineering and Computer Science 2024;35(2):1325–1332.

18. Yulita IN, Helen A, Suryani M. Machine Learning Prediction of Time Series Covid-19 Data in West Java, Indonesia. Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI) 2023;12(2):174–183.

19. Rasheed S, Kumar GK, Rani DM, Kantipudi MVVP, Anila M. Heart Disease Prediction Using GridSearchCV and Random Forest. EAI Endorsed Trans Pervasive Health Technol 2024;10:1–8.

20. Ørebæk OE, Geitle M. Exploring the hyperparameters of XGBoost through 3D visualizations. In: CEUR Workshop Proceedings, vol. 2846; 2021. p. 0.

21. Kaewchada S, Ruang-On S, Kuhapong U, Songsri-In K. Random forest model for forecasting vegetable prices: a case study in Nakhon Si Thammarat Province, Thailand. International Journal of Electrical and Computer Engineering 2023;13(5):5265–5272.

How to cite this article: Niken Riyanti, Roy Rudolf Huizen, Evi Triandini, Comparison of Supervised Machine Learning Methods for Predicting Stunting Prevalence in West Java Province, *IPTEK The Journal for Technology and Science*, *37(1): 50-67.*