JURNAL INFORMATIKA
PROXIES

Vol.9, No.2, January, 2026

Instructions for Preparing Papers for *Proxies:Jurnal Informatika*

Stefani Evelin Aidjili[1],Yonathan Purbo Santosa[2]

[1] Teknik Informatika, Fakultas Ilmu Komputer, Universitas Katolik Soegijapranata, 22k10023@student.unika.ac.id
[2] Teknik Informatika, Fakultas Ilmu Komputer, Universitas Katolik Soegijapranata, yonathansantosa@unika.ac.id

Corresponding Author Email: yonathansantosa@unika.ac.id

https://doi.org/10.18280/proxies.v9i2.14956

Received: 2026-01-28
Revised: 2026-07-20
Accepted: 2026-07-20
Available online: 2026-07-21

Keywords:
car price prediction, macroeconomic features, machine learning

ABSTRACT

Car price prediction is an important step in the automotive industry because it can help consumers and dealers determine fair market value. However, many previous studies only used vehicle specifications. This study combines vehicle specifications and economic indicators to improve prediction accuracy. Three machine learning algorithms were tested: SVR, XGBoost, and Random Forest. The dataset comprises 2,498 car records from Kaggle, US GDP data from fred.stlouisfed.org, and US annual inflation data. The data preprocessing process includes handling missing values, removing outliers using the Interquartile Range (IQR) technique, encoding categorical data, and creating new features such as the ratio between GDP and inflation.

1. INTRODUCTION

Car prices are influenced by various factors, including vehicle specifications and economic conditions. Macroeconomic conditions like GDP and inflation, which can influence car prices in the market. Support Vector Regression (SVR) is one algorithm that can model non-linear relationships using kernel functions. XGBoost is a boosting-based algorithm that builds multiple decision trees to capture complex patterns in large datasets. Random Forest, which creates multiple decision trees using randomly sampled data and features.

Several previous only use internal aspects without using external aspects like GDP and Inflation. This study used to compare the performance of SVR, Random Forest, and XGBoost in predicting car prices using vehicle specifications and macroeconomic indicators like GDP and inflation. The data used in this research are obtained from Kaggle and include several car attributes like mileage, model, and color. Several preprocessing steps, including data cleaning, handling missing values, and encoding categorical variables, are applied to improve data quality. Evaluation metrics used are RMSE, MAE, and R^2. The results are expected to show how macroeconomic factors affect car price prediction

2. LITERATURE STUDY

Numerous studies have utilized machine learning algorithms to forecast the prices of commodities like cars, gold, and various other assets. Qu et al. [1] implemented SVR along with the Grey Wolf Optimizer algorithm to predict car sales. Their findings indicated that SVR yields good results when feature selection and data optimization are effectively managed. However, this research did not take into account external factors like macroeconomic conditions. XGBoost has shown impressive results in price prediction tasks. Norbert Gono et al. [2] applied XGBoost for predicting silver prices and discovered that it surpassed traditional models based on metrics such as MAE, RMSE, MAPE, and R^2. This suggests that XGBoost is well-suited for regression-based price prediction challenges. Giustino and Santosa [3] pointed out that the performance of algorithms is influenced by data characteristics, including non-linearity and data imbalance. Although their research is about detecting toxic comments, this study is applicable to car price prediction, where price distributions often exhibit non-linearity. Sharma et al. [4] demonstrated that XGBoost outperforms Linear Regression in price prediction tasks, indicating that boosting-based models are more resilient to noise and outliers. This reinforces the necessity to compare XGBoost with other non-linear models like SVR.

Astutiningsih et al. [5] implemented XGBoost to predict stock prices of PT. United Tractors Tbk and achieved high accuracy with a MAPE value of 3.89%. However, macroeconomic variables such as inflation and GDP were not included, showing an opportunity to improve price prediction models by incorporating economic factors. Guo and Zhang [6] compared several machine learning algorithms for used car price prediction and found that XGBoost achieved the best performance based on MSE, MAE, and R^2. However, their study reported overfitting issues with small datasets and did not include macroeconomic indicators. Pinata et al. [7] applied XGBoost to predict traffic accidents in Bali using time series data. Although the context is not price prediction, the study demonstrates XGBoost's ability to handle historical data and produce accurate predictions. The study did not include external factors or model comparisons. Zhu [8] proposed a hybrid model combining LSTM and XGBoost for stock price prediction and achieved better accuracy compared to single models. However, this study did not include external factors such as news or market sentiment, which can be related to macroeconomic conditions in vehicle price prediction.

Ardana [9] used XGBoost to predict stock price movements by optimizing parameters and selecting the most influential features. The study showed that proper parameter tuning significantly improves prediction accuracy, which is relevant for vehicle price prediction research. Everingham et al. [10] applied Random Forest to estimate sugarcane yields by integrating simulation data with weather data. The outcomes demonstrated improved accuracy and lower error rates, suggesting that Random Forest is proficient in managing complex datasets, even though it was not directly compared with other models. Huang [11] assessed SVR, Random Forest, and KNN for stock price forecasting and concluded that Random Forest yielded the best results based on MSE. Although external economic variables were not included, the study emphasizes the significance of considering such factors. Adetunji et al. [12] applied Random Forest to house price prediction using specific property features and evaluated the model using R^2, MAE, and RMSE with K-Fold cross-validation. While the study used only one dataset and lacked statistical significance testing, it still shows the potential of tree-based ensemble methods in price prediction tasks.

3. RESEARCH METHODOLOGY

3.1 Dataset Collection

The methodology used in this study is represented through a flowchart, which can be seen in **Figure 3.1**.

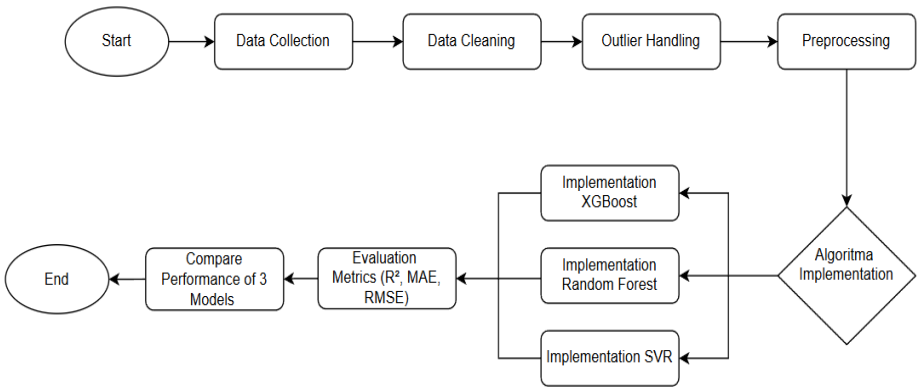

Figure 3.1. Research Methodology

3.2 Dataset Collection

Several datasets are used in this study. The main dataset used is titled "USA - 2025 - Car Price" and is taken from the Kaggle platform. It has 2,498 car data with 13 important attributes. Two additional datasets are also used: US annual inflation data from 1947 to 2023 which also comes from Kaggle, and Gross Domestic Product data from the site *fred.stlouisfed.org* These three datasets are then combined using years as a link.

3.3 Data Cleaning

The procedure for cleansing car data will remove columns deemed unnecessary, such as VIN, lot, country, and title status. Vehicles from the year 2003 were eliminated due to their minimal presence. Additionally, rows where the vehicle colors are labeled as 'no_color' are excluded. In this data cleaning phase, the unnamed:0 column will be removed as it merely serves as an index from the CSV file. The model column is converted to string type since car models can consist of only numbers (for instance Chevrolet model 1500) or a combination of letters and numbers (for instance BMW model X5). Mileage that is less than or equal to zero is removed because mileage never has a negative value.

3.4 Outlier Handling

In this stage, outlier handling is conducted to remove unrealistic car prices that may negatively affect the model's performance. IQR (Interquartile Range) method used to analyze the price range to remove outliers. After all these processes, the data index is reorganized to keep the data.

3.5 Data Preprocessing

At this stage, the car data is equipped and prepared with economic variables like GDP and inflation to improve the accuracy of the model's predictions. This procedure involves modifying the date format, computing the annual average of economic indicators, transforming categorical data into numerical format, and scaling numeric features to suit the needs of the machine learning model.

3.5.1 Handling Missing Values

After being combined with car data based on production year, several rows were found with empty values because there was no year match. These rows were then deleted so that the data used for analysis, especially in economic columns such as GDP and inflation, remained complete and consistent.

3.5.2 Feature Engineering

There will be a process to calculate the average from GDP and Inflation. The calculated average from GDP and Inflation are later merged with the car dataset. Additional economic-based features are created to capture the relationship between macroeconomic variables. The car_age feature represents the age of the vehicle and is calculated by subtracting the production year from the reference year 2025. The mileage_per_year feature is obtained by dividing the total mileage by the vehicle age plus one, to avoid division by zero, and describes the average distance traveled per year. To capture the combined effects of macroeconomic variables, interaction features such as year_gdp_interaction and GDP_Inflation_interaction are created. According to Muris and Wacker [13], interactions between variables can reveal combined effects that may not be visible when variables are analyzed separately. The year_gdp_interaction feature is calculated by multiplying the production year by the GDP value, while GDP_Inflation_interaction is obtained by multiplying GDP and inflation values. These features aim to capture the joint effect of economic growth and price level changes on car prices over time.

3.5.3 Encoding Categorical Variables

This dataset has a number of categorical variables such as brand, model, color, and location (state). In order to utilize this data within a machine learning model, we transform these variables into a numerical format using the LabelEncoder method. Each distinct category is assigned a specific numeric value, which enables the entire dataset to be processed by algorithms that require numeric input.

3.5.4 Feature Scaling and Encoding for SVR Model

In this study, standardization is applied only to the Support Vector Regression (SVR) model because SVR is sensitive to feature scale. The process is carried out using StandardScaler in the SVR preprocessing pipeline. For datasets with economic features, standardization is combined with polynomial transformation and feature selection. For datasets without economic features, only standardization is applied. XGBoost and Random Forest do not require standardization and are trained using the original feature scales. In addition, One-Hot Encoding is used to handle categorical features. Since SVR can only process numerical data, categorical variables such as brand, model, color, and state are converted into numerical form using OneHotEncoder. This step ensures that all features can be processed correctly by the SVR model.

3.6 Modeling and Algorithm Implementation

The process of modeling and algorithm implementation is carried out using three models, namely Support Vector Regression (SVR), Random Forest, and XGBoost. Each model is tested with various parameter values (hyperparameter tuning) to obtain the best combination of parameters (best parameters). The testing process is using the Grid Search method and cross-validation, in order to determine the parameters that produce the most optimal predictive performance.

3.6.1 Support Vector Regression (SVR)

The implementation process starts with the pre-processing and feature engineering stages, which include adding factors like vehicle age, mileage per year, GDP-to-inflation ratio, the interaction between year and GDP, and the interaction for GDP and inflation. After that, the data is split. PolynomialFeatures is used to create a combination of numeric features up to a certain degree (degree 1 and degree 2) so that the SVR model can capture nonlinear relationships between variables. Next, the features are adjusted with StandardScaler so that they have a distribution where the mean is zero and the standard deviation is one. After that, feature selection is carried out using SelectKBest because after the numeric features are expanded using PolynomialFeatures, the number of features increases significantly. Study based on Guyon and Elisseeff [14] in the Journal of Machine Learning Research, which states that external feature selection methods such as SelectKBest are more needed in non-tree models such as SVM, while tree models such as Random Forest and XGBoost already have the ability to determine the importance of features during the training process. Therefore, SelectKBest is only applied to SVR to reduce noise and optimize performance. In this study SelectKBest (5, 10, 15, 20) are tested. A study by Wang et al [15] showed that a high C value makes the model more flexible in capturing complex data, while a small epsilon helps detect small variations in the target. For models that don't include macroeconomic variables, a more cautious C and a bigger epsilon are applied to avoid overfitting on simpler datasets. A study by Annas et all [16] a larger gamma can increase the SVR model's sensitivity to nonlinear patterns in complex data. However, too high a gamma can lead to overfitting. However, a bigger epsilon value makes the model more flexible with small changes, resulting in a smoother model that is less affected by noise. In contrast, a smaller epsilon can capture small changes, but it also raises the chances of overfitting. Finally, the selection of kernels such as RBF, Linear, Polynomial, Sigmoid are tested to know which kernel fits for the dataset used.

Table 3.1. Parameter SVR

Parameter SVR	Value (GDP + Inflation)	Value Without (GDP + Inflation)
C value	2000, 5000	2000, 5000
Epsilon	0.01, 0.05, 0.1	0.01, 0.05, 0.1
Gamma	0.01, 0.05, 0.1	0.01, 0.05, 0.1
Kernel	RBF, Linear, Polynomial, Sigmoid	RBF, Linear, Polynomial, Sigmoid

3.6.2 Extreme Gradient Boosting (XGBoost)

The details on the hyperparameter presented in table 3.2 for the XGBoost model were drilled in two different scenarios, namely using economic features and without economic features. In the scenario with economic features, the model was drilled using the n_estimators parameter , max_ depth and learning_rate. According to Bentéjac et al [17] n_estimators and learning_rate parameters work together. When the learning_rate is low, it usually needs more estimators to get the best results. On the other hand, if the learning_rate is high, using fewer estimators can reduce computational requirements . According to Chen and Guestrin [18] the creators of XGBoost, using a low learning rate can improve prediction accuracy. This is because the model learning process becomes slower and more careful, which is very helpful when dealing with complex data. Like a combination of vehicle specifications and macroeconomic factors. A higher max_depth parameter allows the model to recognize more complex patterns; however, it also increases the risk of overfitting

Table 3.2. Parameter XGBoost

Parameter XGBoost	Value (GDP + Inflation)	Value Without (GDP + Inflation)
n_estimator	200, 300, 500, 800	200, 300, 500, 800
max_depth	5, 6, 8, 10, 30	5, 6, 8, 10, 30
learning rate	0.001, 0.01 0.05, 0.1	0.001, 0.01 0.05, 0.1
random_state	0, 42	0, 42

3.6.3 Random Forest Regressor

The details on the hyperparameter are presented in table 3.3. The model was drilled using the n_estimators parameter, max_depth and min_samples_split. Probst et all [19] said that *n_estimators* and *max_depth* are among the most influential hyperparameters in Random Forest. Increasing the number of trees (n_estimators) improves model stability and prediction accuracy, with deeper trees capable of identifying complex data. According to Zhu et al [20] the *min_samples_split* parameter, which sets the least number of samples needed to be divided, is crucial in managing the complexity of Random Forest models. A smaller min_samples_split making it more sensitive to slight changes in the data. But, a larger min_samples_split value limits the number of splits in the tree.

Table 3.3. Parameter Random Forest

Parameter Random Forest	Value (GDP + Inflation)	Value Without (GDP + Inflation)
n_estimator	200, 300, 500, 800	200, 300, 500, 800
max_depth	5, 6, 8, 10, 30	5, 6, 8, 10, 30
min_sample_split	3, 5, 10	3, 5, 10
random_state	0, 42	0, 42

4. RESULT AND DISCUSSION

4.1 Result

This section presents the test results of three machine learning algorithms, namely Support Vector Regression (SVR), Random Forest, and XGBoost, applied to car price data, considering vehicle specifications and macroeconomic variables such as GDP and inflation.

4.1.1 Table Matrix Comparison

This section presents the test results of three machine learning algorithms, namely Support Vector Regression, Random Forest, and XGBoost, applied to car price data, considering vehicle specifications and macroeconomic variables such as GDP and inflation.

Table 4.1. Table Matrix Comparison

Model	R^2	MAE	RMSE
SVR (GDP+Inflation)	0.6962	3845.55	5741.16
SVR (No GDP+Inflation)	0.6927	3906.89	5774.45
XGBoost (GDP+Inflation)	0.7100	3882.59	5609.38
XGBoost (No GDP+Inflation)	0.7052	3879.68	5656.37

Table 4.1 illustrates matrix comparison between 3 algorithms (SVR, Random Forest and XGBoost). Each algorithm has 2 versions, one includes macroeconomics indicators (GDP and inflation) and another one only uses car specifications. The results are presented in 3 matrices (R^2, MAE, RMSE). Based on the result shown in the table, XGBoost with macroeconomic indicators achieves the highest R^2 value of 0.7100, meaning it explains 71.00% of the variation in car prices, and also has the lowest RMSE of 5609.38, indicating the best predictive performance. The model without GDP and inflation has achieved R^2 = 0.7052, MAE = 3879.68, and RMSE = 5656.37, the XGBoost model demonstrates a small decrease in prediction error. This shows that macroeconomic indicators have benefits for the model. The SVR model, which includes GDP and inflation, comes in second with an R^2 of 0.6962, MAE of 3845.55, and RMSE of 5741.16. This result is slightly better than the version without macroeconomic indicators (R^2 = 0.6927, MAE = 3906.89, RMSE = 5774.45). Random Forest with macroeconomic indicators has an R^2 of 0.6842, MAE of 4022.60, and RMSE of 5854.19, this performance is a bit lower than that of SVR and XGBoost. Overall, XGBoost (GDP + Inflation) achieved the best out of all the models tested.

4.1.2 Heatmap Correlation

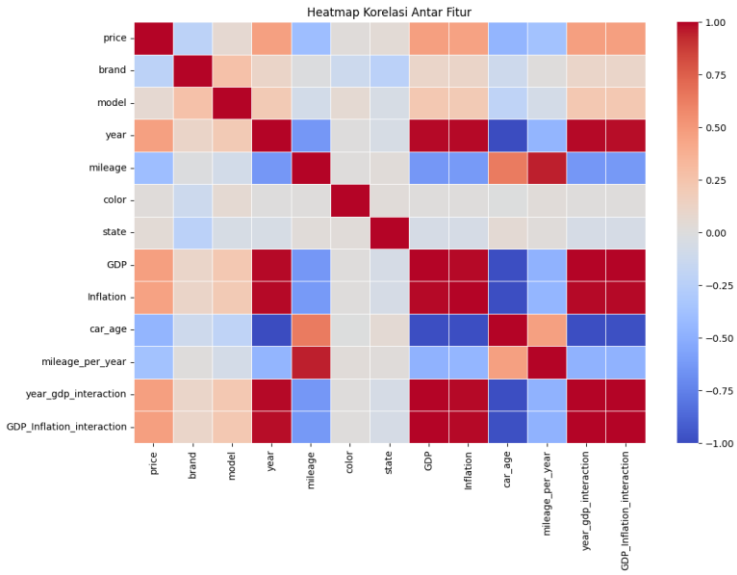

Figure 4.1. Heatmap Correlation

Figure 4.1 shows heatmap correlation between features in the dataset. The heatmap uses a color gradient where red indicates a strong positive correlation (values close to +1), meaning that when one variable increases, the other tends to increase as well. Conversely, blue represents a strong negative correlation (values close to -1), indicating that when one variable increases, the other tends to decrease. Meanwhile, light colors or white areas (around 0) suggest little to no correlation between the variables. From the visualization, car price demonstrates a moderately strong positive correlation (approximately +0.6 to +0.75) with year, GDP, and inflation. This suggests that newer cars and better macroeconomic conditions (higher GDP and controlled inflation) are generally associated with higher vehicle prices. In contrast, mileage has a negative relationship with price (about -0.6), it means that as a vehicle's mileage goes up, its market value usually decreases because of wear and tear. In this dataset, GDP and inflation show a strong positive relationship, indicating that these economic indicators usually change in the same direction during the time. The interaction features, especially GDP_Inflation_interaction and year_gdp_interaction, shows strong positive correlations with their base variables. In general, this figure shows that the specific features of vehicles and economic factors play a role in car price prediction.

4.2 Discussion

Based on the results of three models, XGBoost has the highest prediction accuracy. When macroeconomic factors are included, it achieves an R^2 value of 0.7100, the MAE for XGBoost is 3882.59, and the RMSE is 5609.38. This indicates that XGBoost is the top algorithm for predicting car prices. Additionally, using GDP and inflation data can improve the model's performance The correlation heatmap shows the correlation values among the features in the dataset. The red color shows positive correlations nearing +1, and the blue color shows negative correlations nearing -1. From the visualization, it is evident that car price positively correlates with year, GDP, and inflation. Mileage has a negative correlation with price.

5. CONCLUSION

Overall models with GDP and inflation significantly improve the performance of all models. It can be seen that XGBoost (GDP+Inflation) achieved the best performance with an R^2 value of 0.7100. It means that XGBoost (GDP+Inflation) is able to explain 71% of the data variation, which makes the model with the strongest explanatory power. In terms of error evaluation, XGBoost (GDP+Inflation) is the best with the lowest RMSE value of 5609.38. The low RMSE value in XGBoost shows that this model is better able to suppress error variance and is more stable in avoiding extreme prediction errors (outliers) compared to other models. On the other hand, model SVR (GDP+Inflation) shows the lowest MAE value of 3845,55. Indicating superior overall average prediction accuracy. From heatmap correlation, it can be seen that GDP and inflation have a positive relationship with price. It means that if GDP and inflation are going up it will make the car price grow up. This shows that economic factors play a role as a supporting factor for the market value of a car. According to feature importance, mileage becomes the most dominant feature influencing model prediction. GDP and inflation also contributed to the prediction results, although with a lower weight than vehicle specification. It shows that the price target is determined by vehicle specifications.

REFERENCES

[1] Qu F, Wang Y-T, Hou W-H, et al. Forecasting of Automobile Sales Based on Support Vector Regression Optimized by the Grey Wolf Optimizer Algorithm. *Mathematics* 2022; 10: 2234.

[2] Gono DN, Napitupulu H, Firdaniza. Silver Price Forecasting Using Extreme Gradient Boosting (XGBoost) Method. *Mathematics* 2023; 11: 3813.

[3] Giustino JK, Santosa YP. TOXIC COMMENT CLASSIFICATION COMPARISON BETWEEN LSTM, BILSTM, GRU, AND BIGRU. *Proxies J Inform* 2024; 7: 115–127.

[4] Sharma H, Harsora H, Ogunleye B. An Optimal House Price Prediction Algorithm: XGBoost.

[5] Astutiningsih T, Saputro DRS, Sutanto. Optimasi Algoritme Xtreme Gradient Boosting (XGBoost) pada Harga Saham PT. United Tractors Tbk. *SPECTA J Technol* 2023; 7: 632–641.

[6] Guo S, Zhang B. Revolutionizing the used car market: Predicting prices with XGBoost. *Appl Comput Eng* 2024; 48: 173–180.

[7] Pandika Pinata NN, Sukarsa IM, Dwi Rusjayanthi NK. Prediksi Kecelakaan Lalu Lintas di Bali dengan XGBoost pada Python. *J Ilm Merpati Menara Penelit Akad Teknol Inf* 2020; 188.

[8] Zhu Y. Stock Price Prediction based on LSTM and XGBoost Combination Model. *Trans Comput Sci Intell Syst Res* 2023; 1: 94–109.

[9] Ardana A. Performance Analysis of XGBoost Algorithm to Determine the Most Optimal Parameters and Features in Predicting Stock Price Movement. *Telematika* 2023; 20: 91.

[10] Everingham Y, Sexton J, Skocaj D, et al. Accurate prediction of sugarcane yield using a random forest algorithm. *Agron Sustain Dev* 2016; 36: 27.

[11] Huang Y. Research on the Google Stock Price Prediction Based on SVR, Random Forest, and KNN Models. *Highlights Bus Econ Manag* 2024; 24: 1054–1058.

[12] Adetunji AB, Akande ON, Ajala FA, et al. House Price Prediction using Random Forest Machine Learning Technique. *Procedia Comput Sci* 2022; 199: 806–813.

[13] Muris C, Wacker K. Estimating interaction effects with panel data. Epub ahead of print 14 March 2025. DOI: 10.48550/arXiv.2211.01557.

[14] Guyon I, Elisseeff A. An Introduction to Variable and Feature Selection.

[15] Wang S, Dong L, Hua H. Parameter optimization of support vector machine based on improved grid algorithm. *J Phys Conf Ser* 2020; 1693: 012108.

[16] Annas S, Rais Z, Aswi A, et al. Implementation of Support Vector Regression (SVR) Analysis in Predicting Gold Prices in Indonesia. In: Djam'an N, Sidjara S, Fachry S, et al. (eds) *Proceedings of the 5th International Conference on Statistics, Mathematics, Teaching, and Research 2023 (ICSMTR 2023)*. Dordrecht: Atlantis Press International BV, pp. 97–107.

[17] Bentéjac C, Csörgő A, Martínez-Muñoz G. A Comparative Analysis of XGBoost. *Artif Intell Rev* 2021; 54: 1937–1967.

[18] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco California USA: ACM, pp. 785–794.

[19] Probst P, Wright M, Boulesteix A-L. Hyperparameters and Tuning Strategies for Random Forest. *WIREs Data Min Knowl Discov* 2019; 9: e1301.

[20] Zhu N, Zhu C, Zhou L, et al. Optimization of the Random Forest Hyperparameters for Power Industrial Control Systems Intrusion Detection Using an Improved Grid Search Algorithm. *Appl Sci* 2022; 12: 10456.