Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)
Vol. 10, No. 4, December 2024, pp. 870-884
ISSN: 2338-3070, DOI: 10.26555/jiteki.v10i4.30484

Impact of Feature Selection on XGBoost Model with VGG16 Feature Extraction for Carbon Stock Estimation Using GEE and Drone Imagery

I Made Darma Cahya Adyatma, Erwin Budi Setiawan
Faculty of Informatics, Telkom University, Jl. Terusan Buah Batu, Bandung 40257, Indonesia

ARTICLE INFO

Article history:

Received December 21. 2024
Revised January 09, 2025
Accepted January 14, 2025

Keywords:

XGBoost;
VGG16;
Feature Importance;
Information Gain;
RFE

ABSTRACT

Carbon stocks are critical to climate change mitigation by capturing atmospheric carbon and storing it in biomass. However, carbon stock estimation faces challenges due to data complexity and the need for efficient analytical methods. This study introduces a carbon stock estimation method that integrates the XGBoost algorithm with VGG16 feature extraction and feature selection techniques to analyze GEE and Drone image datasets. The model is evaluated through four scenarios: without feature selection, using Information Gain, using Feature Importance, and using Recursive Feature Elimination. These scenarios aim to compare feature selection methods to identify the best one for processing complex environmental data. The experimental results show that RFE significantly outperforms other methods, achieving an average RMSE of 6651.62, MAE of 2297.57, and R^2 of 0.7673. These findings underscore the importance of feature selection in optimizing model performance, particularly for high-dimensional environmental datasets. RFE shows superior accuracy and efficiency by retaining the most relevant features but requires more computational resources. For applications that prioritize time and resource efficiency, Information Gain or Feature Importance can serve as a practical alternative with slightly reduced accuracy. This research highlights the value of integrating feature selection techniques into machine learning models for environmental data analysis. Future research could explore alternative feature extraction methods, combine RFE with other approaches, or apply advanced techniques such as Boruta or genetic algorithms. These efforts will further refine carbon stock estimation models, paving the way for broader applications in environmental data analysis.

Corresponding Author:

Erwin Budi Setiawan, Telkom University, Jl. Terusan Buah Batu, Bandung 40257, Indonesia
Email: erwinbudisetiawan@telkomuniversity.ac.id

1. INTRODUCTION

The quantity of carbon stored in an ecosystem's above-ground and below-ground biomass is known as its carbon stocks [1], [2]. Carbon stocks are significant in addressing climate change due to the ability of plants to absorb or retain carbon from the atmosphere and store it in the form of biomass, thereby reducing greenhouse gas concentrations [3], [4], [5]. Forests, as one of the largest carbon sinks, play a crucial role in regulating the global climate [6], [7]. Therefore, studies related to carbon stock estimation are crucial in supporting environmental conservation and climate change mitigation [8].

However, a significant challenge in carbon stock studies is the management of complex data and the need for efficient analysis methods. The use of technologies such as machine learning (ML) has become increasingly popular due to its ability to analyze large-scale data and discover patterns that are difficult to recognize with traditional methods [9]. In the context of carbon stocks, many studies have explored the potential of ML to

analyze imagery and other datasets to improve the quality of analysis results. For example, in [10] a study was conducted using Landsat 8 OLI data using several regression algorithms such as Support Vector Machine (SVM), Random Forest (RF), k-Nearest Neighbors (kNN), and XGBoost and using the Boruta Method feature selection method. This research shows that XGBoost gives the best performance R^2 = 0,89. Another study [11] used the XGBoost model with the Gradient Boosting selection feature to predict Soil Organic Carbon Stock (SOCS) on sentinel-1 and sentinel-2 datasets and field data and showed the results of the above model were R^2 = 0/59. In addition, research [12] applies various ML models to predict soil organic carbon content (SOC), one of which is XGBoost with a feature selection Genetic Algorithm. This study uses several types of datasets, such as soil data and Auxiliary variables, which include 105 predictor variables derived from various sources, including 60 variables generated from Landsat 8 and MODIS satellite images. The results of the XGBoost model in this study Mean Absolute Error (MAE) = 0.66%, Root Mean Square Error (RMSE) = 0.82%, R^2 = 0/57.

Many studies have explored the use of XGBoost in environmental data analysis. However, few have integrated the advanced feature extraction capabilities of VGG16, especially in the context of complex data such as carbon stock estimation. Based on the above research, this study uses XGBoost and VGG16 due to their advantages in complex data analysis tasks. VGG16 is used for its ability to extract visual features automatically and efficiently, especially on images with complex structures such as satellite and drone images [13], [14]. The VGG16 model pre-trained with the ImageNet dataset is used to process and extract important features from the image dataset. The obtained features were then utilized as input for the XGBoost model, which was selected on the basis of its superior ability to manage high-dimensional regression data. XGBoost's ability to utilize these high-dimensional features is critical as it allows the model to make more accurate predictions about carbon stocks by utilizing the extracted features. It has built-in features for feature selection, which helps to reduce noise, prevent overfitting, and improve model accuracy [12], [15], [16], [17], [18]. In addition, previous research shows that XGBoost consistently provides the best results compared to other algorithms, such as Random Forest and SVM [10], especially in tasks involving environmental data. As far as the researchers know, no studies have explored feature selection on XGBoost models with features extracted using the VGG16 architecture, particularly in regression models with imagery datasets related to carbon stock. This study not only uses imagery datasets but also integrates field data that measures the total carbon content at locations corresponding to the imagery datasets. The field data is used to verify and accurately label the imagery dataset, improving the accuracy of the carbon stock estimation model. By combining direct field measurements with imagery datasets, the developed model is able to provide more accurate and reliable predictions, reflecting actual conditions on the ground. This study evaluates the impact of feature selection techniques on the performance of carbon stock estimation models by implementing four different scenarios: a baseline model without feature selection, a model that uses Information Gain, a model that uses Feature Importance, and a model that will use Recursive Feature Elimination (RFE). The selection of these feature selection techniques is based on their potential to improve model accuracy and efficiency. Information Gain is used as a feature selection technique to reduce data dimensionality by prioritizing features that have a high level of importance and help sort the most informative features [19], [20]. Feature Importance, generated through the trained XGBoost model, allocates a score to each feature based on its contribution to model accuracy [21]. This technique allows the selection and focus on the most significant features, and later, Information Gain and Feature Importance will use Top N Features starting from 500 to 5000. Recursive Feature Elimination (RFE) will be implemented to iteratively reduce the number of features, eliminating variables that contribute the least to the predictive power of the model and aiming to build a more compact and efficient model without compromising its performance. The contribution of the research is to propose the use of XGBoost and VGG16 algorithms for feature extraction complemented by the application of feature selection techniques to enhance model accuracy. This research also makes an important contribution in the form of a comprehensive comparison between various models developed using different feature selection techniques. This analysis aims to identify the most effective models for accurately estimating carbon stocks. By conducting an in-depth evaluation, we were able to determine the optimal feature selection, which significantly improved the accuracy of carbon stock prediction. This method utilizes field data and image data to improve accuracy in carbon stock estimation.

2. METHODS

This research was conducted through several stages, as depicted in the flowchart in Fig. 1. The research starts from the data collection stage, where the data consists of field data and imagery data (drone images and satellite imagery), which is then continued with the data preprocessing stage. The preprocessing stage includes data labeling, data padding, data augmentation, and feature extraction, which will be explained further in the

next sub-chapter. After that, the data is processed through 3 different scenarios, namely: Baseline Model, Feature Selection with Feature Importance, and Feature Selection with Information Gain. Furthermore, each scenario data is separated into training data (80%) and test data (20%) to ensure that the model evaluation can measure the scenario performance fairly and consistently.

Furthermore, the training data is used for XGBoost model training, while the test data is used for model performance evaluation. The final stage is to evaluate the results of each scenario. This evaluation aims to compare the effectiveness of each feature selection technique in improving the accuracy and prediction efficiency of the XGBoost model. These results will provide important insights into the most effective approach to data processing.

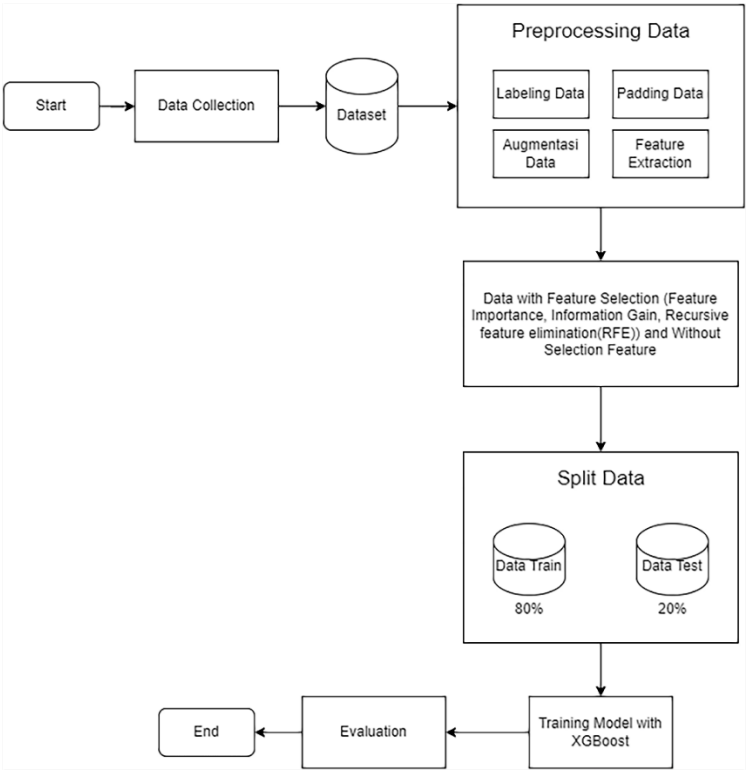

Fig. 1. Research flowchart

2.1. Data Collection

The data in this study was collected through three main methods, namely field data collection, satellite image collection, and data collection using drones. Field data and satellite imagery were collected in Bandung, Semarang, Cirebon, and Banten. Furthermore, due to limited resources for drones, drones were taken only in Bandung, namely the Telkom University area.

In the field data collection method, data was obtained directly by sending a team consisting of lecturers and trained students to the research site, which was located in an area that received CSR assistance from Telkom. Data collection was carried out in several predetermined zones, where each zone was separated by a distance of 50 meters to ensure the accuracy and sustainability of the sampling and data collection process in accordance with Indonesian National Standards Carbon stock measurement and accounting – Field measurements for land-based carbon accounting. Each zone was divided into 20×20 meter plots. As depicted in Fig. 2 each plot was further divided into several sub-plots based on the following categories: sub-plot 'A' for seedlings, litter, and understory with a minimum area of 1 m²; sub-plot 'B' for saplings with a minimum area of 25 m²; sub-plot 'C' for poles with a minimum area of 100 m²; and sub-plot 'D' for trees with a minimum area of 400 m².

At this stage, seedling and understorey biomass from each plot was collected using the prepared sample containers. Samples were weighed to measure the wet weight of approximately ±300 grams and labeled with the naming format ZxHxPx. The labeled samples were then sent to the laboratory for carbon content measurement using appropriate equipment, such as measurement tools and sample containers. Furthermore, during the field data collection, the team recorded the coordinates at the center of each plot. Then these

coordinates will be used to retrieve the imagery dataset through Google Earth Engine and show the plot condition from these coordinates through satellite images. Drone imagery was taken of the area around the measurement plots. The dataset of plot images from Google Earth Engine and drones was processed and cropped according to the plot size on the ground.

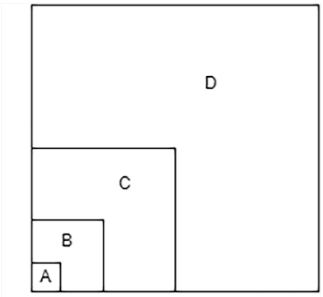

Fig. 2. Example of 20×20meter plot

In addition to field data, the satellite image retrieval method was conducted using the Google Earth Engine (GEE) platform. This cloud-based platform that enables efficient geospatial data processing with access to a wide range of satellite imagery [22], [23]. This process consists of various steps, namely identifying the study area based on the location of the field data collection plot, creating polygon plots according to the size of the field data collection plot, and storing the polygon plot data in image format (PNG) as shown in Fig. 3. The saved data will then be processed further in the preprocessing stage.

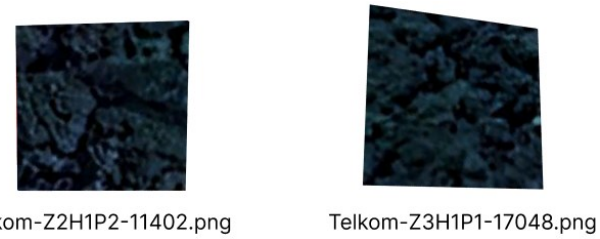

Fig. 3. Example of GEE dataset

The final technique is drone image data gathering, which uses a DJI Mavic 2 Pro drone fitted with a Hasselblad L1D-20c camera to collect image data. This process produces very high resolution images, which were taken from the research location in the Telkom University area. An example of the resulting image can be seen in Fig. 4. Through these three methods, the data collected includes field information as well as satellite and drone imagery that supports this research.

Fig. 4. Example of Drone dataset

2.2. Preprocessing Data

The datasets that have been collected through various sources are then processed to ensure their quality and consistency according to the needs of the model. The first step in preprocessing is data labeling, where each image is named with the format: [Location Name]-[Zone Number][Expanse Number][Plot Number]-[Carbon Value]. This naming was designed to reflect the location of data collection as well as the pre-calculated carbon value, making it easier to manage and analyze the data. Table 1 shows an example of data labeling.

Table 1. Example of Labeling Dataset

No	File Name
1	Telkom-Z4H1P1-2826.png
2	cirebon-S3J2P1-112,09.png
3	semarang-J3P1-14,76.png

Next, an image cropping process is performed to increase the amount of training data. The plot image with an initial size of 20×20 meters was cropped into several smaller sizes, namely 10×10 meters, 5×5 meters, and 1×1 meters. For images smaller than 20×20 meters, the carbon value of the plot is adjusted by adding a certain percentage based on the cropping size. This approach allows diversification of the dataset while maintaining the accuracy of the carbon value information on each plot.

In the next stage, data padding is applied to ensure consistency of image size without changing the original proportions [24]. This process uses Zero-padding [25], [26], which is the addition of a blank black area (RGB: 0, 0, 0) around the image, allowing all images to reach the standard dimensions of 224×24 pixels required by the VGG16 model for feature extraction. This step aims to ensure that all images are of uniform size so that they are compatible with the input model while improving efficiency in the training stage [24], [27], [28].

Data augmentation is carried out using a variety of transformation techniques, including cropping, flipping, rotating, and scaling, to further enhance the dataset's quality [29], [30], [31]. These augmentation techniques aim to increase the variety of training data without adding new data so that the model becomes more robust to variations in the data [32], [33]. In addition, the augmentation process is done without changing the data labels to maintain consistency between input and output [34]. By providing more diverse data, augmentation also helps reduce the risk of overfitting during the model training process [29], [35]. Through these stages, the preprocessing dataset is designed to produce uniform, varied, and high-quality data, thus supporting optimal performance in the developed model. This research uses several augmentation techniques, including image rotation techniques up to 30 degrees that help the model recognize objects from various angles, rescaling techniques by changing the scale of pixel values from [0, 255] to [0, 1] to facilitate the training process by simplifying input data, then flipping techniques, both horizontal and vertical flipping to teach the model to recognize patterns without depending on the original orientation of the image, increasing its generalization ability [29].

2.3. Feature Extraction using VGG16

In the feature extraction stage, the VGG16 model is used to extract features from the input image without including a final classification layer and trained using the ImageNet dataset. VGG16 is a pre-trained model with deep convolutional neural network architecture [14], [36], [37]. The model is designed to recognize visual patterns by utilizing 13 convolutional layers and three pooling layers, making it highly effective in extracting hierarchical features from images [13]. The input images are first resized to 224×224 pixels, converted into NumPy arrays, and preprocessed to align with the configurations of the VGG16 training environment on ImageNet, ensuring that each image is appropriately normalized and ready for feature extraction. The extracted features are then flattened into one-dimensional vectors comprising 25,088 features each, ensuring compatibility for subsequent processing steps. All features generated from the images in the dataset are used in the next stage for feature selection, which is the main focus of this research. Fig. 5 illustrates the architecture of the VGG16 model used for feature extraction in this study.

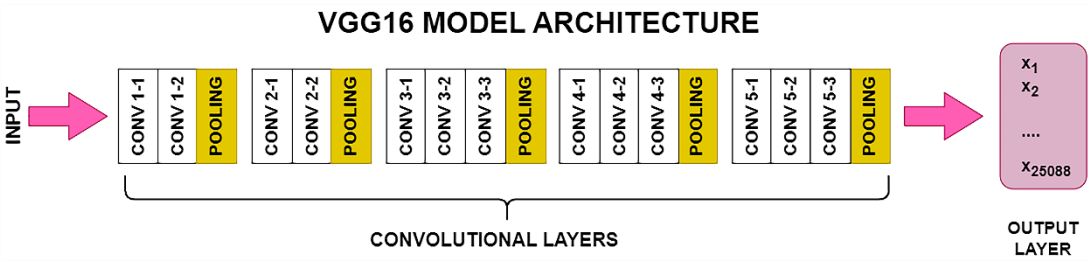

Fig. 5. Architecture VGG16

2.4. Feature Selection

Feature selection is the process of selecting a subset of all the features available in a dataset that are deemed most relevant to achieve a particular analysis goal [38], [39]. It helps reduce the dimensionality and complexity of the dataset, enabling more efficient and accurate analysis [38], [40], [41]. In feature selection,

there are several main categories divided by label information and search strategy. This research will use a selection of features from the category of search strategies, which are commonly used to improve the efficiency and effectiveness of the model [42]. There are several commonly used feature selection methods to improve the efficiency and effectiveness of the model. First, filter methods rely on the general characteristics of the training data to select features independent of any predictors. Secondly, wrapper methods involve optimization of predictors as part of the selection process, and thirdly, embedded methods combine the advantages of both filter and wrapper methods in feature selection [43], [44]. This research uses Information Gain, which is a filter method because it can handle datasets with large dimensions [19], [20]; Feature Importance derived from XGBoost is an embedded method [21]. RFE, which is a wrapper method, is used because it is effective for high-dimensional data. After all, RFE helps reduce the dimension significantly without losing important information but is slower than the filter method because each RFE iteration involves training a model to assess the importance of features, so it takes longer, especially in this dataset which has 25088 features [45]. To overcome these challenges and optimize resource usage, this study uses a step size of 1,254 features per iteration, which is approximately 5% of the total features. This step size was chosen to reduce the number of iterations required, thereby improving computational efficiency and reducing runtime. Using a smaller step size, such as 100 features or less, would require more iterations, resulting in higher computational cost and longer processing time.

2.5. Information Gain

Information Gain is employed in this study as a feature selection method for the research model. Information Gain is a feature selection method that takes into account features with a high degree of relevance in order to reduce the dimensionality of the data [19], [20]. This method evaluates each feature based on its contribution to providing information to separate target classes in a dataset. Information Gain helps sort the most informative features so that only the attributes with the highest weights are retained for use in the next process [46]. Scenario To investigate the effect of feature quantity on model performance, Information Gain will choose the top N features in increments of 500, ranging from 500 to 5000. This method allows us to evaluate the usefulness of features in terms of their ability to improve the prediction accuracy of the model. The formula for calculating information gain is shown in (1).

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{Si}{S} \times Entropy(Si) \quad (1)$$

Explanation of (1), S is the dataset, A is the evaluated feature, Si is the subset of the dataset that has a certain value for feature A, and n is the number of unique values of feature A [47].

2.6. Extreme Gradient Boosting (XGBoost)

XGBoost is a gradient-boosting algorithm optimized to improve computational efficiency and performance in regression and classification models [15], [16]. It was developed by Tianqi Chen and Carlos Guestrin and designed as an efficient and scalable implementation to handle large-scale data [12], [17]. By simplifying the objective function and supporting parallel computation during the training process, XGBoost is able to reduce the risk of overfitting and improve the speed and accuracy of computation [12].

For the feature selection scenario, the model used the feature importance score generated by the XGBoost model [21]. In this scenario, the model is first trained using all available features to establish a baseline. Once training is complete, the feature_importances_ attribute of the model assigns a score to each feature, reflecting their relative importance based on their contribution to the model's accuracy. This score is then used to rank and select the N features with the highest importance. Started with the top 500 features and increased the number in 500 increments, up to a maximum of 5000 features. This process allows us to evaluate the effect of the number of features on model performance, ensuring that only the most informative features are used in further analysis.

2.7. Recursive Feature Elimination (RFE)

Recursive Feature Elimination (RFE) is a feature selection method that aims to select the most relevant subset of features to reduce data dimensionality. The process starts by considering all the variables available in the model. Iteratively, RFE removes one or more features with the smallest contribution to model performance at each step. This process continues until the desired number of features is reached or only one feature remains, depending on the selection objective [45]. In this study, RFE is used to reduce the dimensionality of the data generated from the VGG16 model, which has a total of 25088 features per iteration. The model used in the RFE process is the Random Forest Regressor, which is able to assess the importance of features based on their influence on the regression target. In this study, 1254 features were eliminated per

iteration, leaving only the single most important feature. After that, RFE generates a feature ranking, where rank 1 is the most relevant feature. The resulting features are selected top N features, ranging from the top 500 to the top 5000, to be used in training the XGBoost model. This process allows the exploration of model performance with various numbers of features, helping to determine the optimal number of features that provide the best accuracy and efficiency.

2.8. Model Evaluation

Model evaluation for regression methods is performed using several key evaluation metrics, namely Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Coefficient of Determination (R^2). These metrics are used to measure how well the regression model predicts the target value compared to the actual value.

RMSE is simply the square root of MSE [48], represented in (2). RMSE helps assess how well the regression model predicts the target value. MAE is used to measure the absolute difference between the predicted and actual values, as shown in (3). R^2 is a statistical measure used in the context of linear regression to show how well the independent variables explain the variability of the dependent variable [49], [50], detailed in (4).

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(P_i - r_i)^2} \quad (2)$$

N is the number of observations, P_i is the value predicted by the model for the i-th observation, r_i is the actual value for the i-th observation.

$$MAE = \frac{\sum_{i=1}^{N}|y_i - x_i|}{N} \quad (3)$$

where N is the number of observations. y_i is the value predicted by the model for the i-th observation x_i is the actual value of the i-th observation.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (4)$$

where y_i is the actual value for the i-th observation. $\hat{y}_i$ is the value predicted by the model for the i-th observation. $\bar{y}$ is the average of the actual values. n is the number of observations.

3. RESULTS AND DISCUSSION

This section contains the results of the research conducted, specifically explaining the research findings that utilize XGBoost and the features extracted using VGG16 with the Imagery dataset consisting of the GEE dataset and the Drone dataset, where 8,762 images from GEE and 2,072 images from drone images and the total data is 10,834 images. Each data has passed the data preprocessing process consisting of data labeling, data padding, data augmentation, and feature extraction using VGG16. After preprocessing, the dataset will be used to evaluate the performance of the XGBoost model in four different scenarios. The impact of each preprocessing step on the model's performance was significant: data padding helped maintain image integrity, data augmentation increased the robustness of the model against overfitting, and careful feature extraction using VGG16 ensured that the most relevant features were used for training the model.

3.1. Baseline Model

In the first scenario, the preprocessed dataset is used without additional feature selection. The model training process is used using XGBoost, where the dataset will be divided into train data and test data with a ratio of 80:20. Model performance is evaluated using RMSE, MAE, and R^2 metrics, which are shown in Table 2. A 5-fold cross-validation technique was employed to provide a more robust evaluation of the model, a 5-fold cross-validation technique was also applied, ensuring that the model is reliable across different subsets of data. The results of the model evaluation are shown in Table 2 and cross-validation in Table 3.

Table 2. Evaluation Result of the Baseline Model

Metric	Result
RMSE	7455.164635748977
MAE	2555.294054016815
R^2	0.7046865389897251

Table 3. Cross Validation Baseline Model

Fold	RMSE	MAE	R^2	Time
1	7492.6491	2629.9569	0.7017	86.58s
2	7608.3610	2723.7737	0.7269	85.66s
3	6612.5663	2247.9768	0.7662	87.45s
4	6830.3134	2371.8654	0.7300	92.32s
5	6893.1691	2388.1574	0.7542	91.51s
Average ± Standard Deviation	7087.4118 ± 391.1300	2472.3460 ± 176.3988	0.7358 ± 0.0225	
Total Time				443.53s

Based on Table 2 the result of the baseline model is that the RMSE value of 7455.16 indicates the average magnitude of the prediction error, where a lower value would reflect better model accuracy. The MAE of 2555.29 indicates that the model, on average, misses about 2555 units in its predictions. The R^2 value of 0.7047 indicates that the model explains about 70.47% of the variability in the target variable, which is considered a moderate fit for the regression task. These results provide an initial idea of the model's performance before performing advanced steps such as feature selection. This baseline model will be used as a reference to compare the performance of more complex models in the next scenario. Table 3 presents the results of the 5-fold cross-validation analysis performed on the baseline model to analyze model performance. Cross-validation was performed to statistically ensure the validity of the model and to test its stability against variations in the training and testing data. The results of each fold, consisting of the metrics RMSE, MAE, R^2, and computation time, were recorded to provide a comprehensive picture of the model's performance. In the first fold, the model recorded an RMSE of 7492.6491, MAE of 2629.9569, and coefficient of determination (R^2) of 0.7017, with a processing time of approximately 86.58 seconds. The second fold showed a slight increase in R^2 to 0.7269 with RMSE and MAE of 7608.3610 and 2723.7737, respectively, and a faster processing time of 85.66 seconds. The third and fourth folds showed a further increase in R^2, as well as a decrease in RMSE and MAE values, indicating increased effectiveness of the model in analyzing that subset of data. The time taken for these two folds was 87.45 and 92.32 seconds. The fifth fold produced an R^2 of 0.7542 with an RMSE of 6893.1691, an MAE of 2388.1574, and a time of 91.51 seconds. The average of the five folds resulted in an RMSE of 7087.4118 with a standard deviation of 391.1300, MAE of 2472.3460 with a standard deviation of 176.3988, and R^2 of 0.7358 with a standard deviation of 0.0225, indicating the model's consistency in performance across the five-fold cross-validation. The total time taken to complete all folds was 443.53 seconds, showing relatively good time efficiency in terms of computation.

3.2. Feature Selection with Feature Importance

In this scenario, the XGBoost Model is trained with features that have been selected based on Feature Importance. This process aims to reduce data complexity by retaining the most influential features, which is expected to improve model performance and computational efficiency. In this scenario, Top-N features chosen at different points in time, ranging from 500 to 5000, were used to retrain the model. The goal of this progressive selection was to track how model performance changed as more features were kept. Finding the ideal feature count that would strike a balance between computing efficiency and model correctness was the goal. Model performance was evaluated using the RMSE, MAE, and R^2 metrics in Table 4. A 5-fold cross-validation technique was employed to ensure that the model performance assessment is robust and reliable. A 5-fold cross-validation technique was applied. The results of this cross-validation, which show the performance of the model more comprehensively through various subsets of data, are presented in Table 5.

Table 4. Evaluation Result of the Feature Importance Model

Top-N Features	RMSE	MAE	R^2
500	7063.5704	2412.7635	0.7349
1000	7030.6961	2422.5287	0.7374
1500	7028.3192	2421.4212	0.7375
2000	7028.3192	2421.4212	0.7375
2500	7028.3192	2421.4212	0.7375
3000	7028.3192	2421.4212	0.7375
3500	7028.3192	2421.4212	0.7375
4000	7028.3192	2421.4212	0.7375
4500	7028.3192	2421.4212	0.7375
5000	7028.3192	2421.4212	0.7375

Table 5. Cross Validation Feature Importance

Top-N Features	RMSE (Average ± Std)	MAE (Average ± Std)	R^2 (Average ± Std)	Time
500	6401.89 ± 251.21	2207.16 ± 94.06	0.7851 ± 0.0185	3.920249s
1000	6531.96 ± 226.30	2255.89 ± 93.32	0.7765 ± 0.0161	7.228177s
1500	6528.67 ± 203.96	2251.26 ± 96.04	0.7768 ± 0.0141	9.640155s
2000	6518.11 ± 219.36	2244.07 ± 87.93	0.7775 ± 0.0152	11.044267s
2500	6512.85 ± 194.19	2241.53 ± 93.81	0.7778 ± 0.0150	13.613242s
3000	6542.26 ± 169.04	2271.95 ± 78.07	0.7757 ± 0.0155	13.959764s
3500	6458.87 ± 176.46	2237.76 ± 92.92	0.7816 ± 0.0127	16.544172s
4000	6564.74 ± 175.60	2273.17 ± 100.07	0.7744 ± 0.0117	17.011002s
4500	6514.42 ± 170.63	2259.86 ± 105.87	0.7779 ± 0.0108	19.983680s
5000	6552.04 ± 108.03	2269.21 ± 72.22	0.7753 ± 0.0092	20.058310s
Overall Average	6512.5810 ± 189.4780	2251.1860 ± 91.4310	0.7779 ± 0.0139	
Total Time				133.003018s

Based on Table 4, selecting Top-N Features for training the XGBoost model shows improved performance up to 1500 features, with an RMSE of 7028.32, MAE of 2421.42, and R^2 of 0.7375. These values indicate that the model can explain about 73.75% of the variability of the carbon stock data, which reflects good performance. However, after 1500 features were selected, even if the number of features was increased to 5000, the RMSE, MAE, and R^2 metrics remained stable, indicating that adding further features did not improve model performance. The model was already effective enough with 1500 features, so additional features did not contribute significantly to accuracy or reduction in prediction error. Adding irrelevant features only adds complexity to the model without improving its performance, as well as extending computation time and complicating interpretation. Thus, 1500 features are the optimal number for this model, and further feature selection does not provide significant benefits. Table 5 presents the results of the 5-fold cross-validation analysis performed on the model with feature importance to assess the impact of feature selection on model performance. The results, including metrics RMSE, MAE, R^2, and computation time, were meticulously recorded to provide a detailed view of the model's effectiveness across different feature subsets. The overall analysis from 500 to 5000 features indicates that while adding more features did not significantly enhance the RMSE or R^2 values beyond the initial 500 features, the model maintained a robust performance across all metrics. The average and standard deviation of RMSE across all feature sets were 6586.09 ± 109.05, MAE was 2258.19 ± 95.06, and R^2 was 0.7771 ± 0.015, demonstrating the model's consistency in handling feature variability. The total computation time showed a linear increase with the number of features, emphasizing the trade-off between computational efficiency and model complexity.

3.3. Feature Selection with Information Gain

In this scenario, the feature selection process is performed using the Information Gain method. This method aims to evaluate each feature based on its contribution to providing information to the target. The features with the highest scores are selected for use in training the XGBoost model. The feature selection data is divided into training data and testing data with a ratio of 80:20. The XGBoost model was trained using the training data and evaluated on the testing data. The evaluation is done using the RMSE, MAE, and R^2 metrics in Table 6. A 5-fold cross-validation technique was applied, and the results of this validation are presented in Table 7.

Based on Table 6, the selection of Top-N Features using Information Gain shows improved performance up to 1500 features, with RMSE of 7287.76, MAE of 2452.80, and R^2 of 0.7178. These values indicate that the model can explain about 71.78% of the variability of the carbon stock data, which reflects a good level of accuracy. However, after 1500 features were selected, increasing the number of features to 5000 did not result in significant improvements in the RMSE, MAE, or R^2 metrics. Indicates that after a certain point, adding more features does not provide valuable additional information to the model, as the initial features have already captured the relevant patterns. Adding irrelevant or redundant features will only increase the complexity of the model without providing any meaningful improvement, which in turn will extend the computation time and complicate the interpretation of the model. Therefore, 1500 features were identified as the optimal number for this model, and further feature selection did not provide any significant benefit to the model performance. Table 7 presents the results from a 5-fold cross-validation analysis conducted on the model utilizing Information Gain for feature selection. The analysis showcases that increasing the number of features generally maintains the model's robustness without significantly enhancing performance metrics beyond the initial set of features. Specifically, the RMSE and R^2 values do not show substantial improvement as more features are added, which suggests a point of diminishing returns in feature inclusion. The overall averages for RMSE and MAE are

7074.58 and 2448.67, respectively, with a slight variability indicated by their standard deviations-similarly, R^2 averages at 0.7368, reflecting consistent predictive accuracy across different feature sets. The table also notes the total computation time of 240.66 seconds, highlighting the computational demands of handling larger feature sets.

Table 6. Evaluation Result of the Information Gain Model

Top-N Features	RMSE	MAE	R^2
500	7542.5688	2578.0617	0.6977
1000	7490.0680	2539.2658	0.7019
1500	7287.7600	2452.8006	0.7178
2000	7511.4340	2615.1572	0.7002
2500	7464.5636	2558.1963	0.7039
3000	7504.3945	2604.6947	0.7008
3500	7480.3058	2593.4742	0.7027
4000	7393.7409	2569.8018	0.7095
4500	7328.0883	2570.3598	0.7147
5000	7350.2846	2536.5060	0.7129

Table 7. Cross Validation Information Gain

Top-N Features	RMSE (Average ± Std)	MAE (Average ± Std)	R^2 (Average ± Std)	Time
500	7037.3912 ± 497.6351	2429.0394 ± 191.5723	0.7398 ± 0.0246	5.40s
1000	7012.5462 ± 354.6951	2419.0379 ± 150.9752	0.7411 ± 0.0234	9.53s
1500	7113.0591 ± 266.4548	2447.8339 ± 133.4602	0.7342 ± 0.0116	15.20s
2000	7068.7249 ± 276.2367	2439.4110 ± 110.5843	0.7373 ± 0.0148	18.84s
2500	7045.9459 ± 507.2022	2440.1798 ± 180.6700	0.7386 ± 0.0309	23.45s
3000	7098.3828 ± 431.9045	2447.1553 ± 168.9598	0.7347 ± 0.0273	26.74s
3500	7104.8367 ± 377.8103	2474.7229 ± 154.3310	0.7346 ± 0.0208	32.25s
4000	7118.4804 ± 296.0395	2489.2646 ± 124.8133	0.7335 ± 0.0169	32.73s
4500	7107.2879 ± 337.1540	2452.1383 ± 164.0920	0.7345 ± 0.0171	38.07s
5000	7039.1440 ± 425.2726	2447.8960 ± 182.0567	0.7398 ± 0.0184	38.45s
Overall Average	7074.5799 ± 377.0405	2448.6679 ± 156.1515	0.7368 ± 0.0206	
Total Time				240.66s

3.4. Feature Selection with Recursive Feature Elimination

In this last scenario, this scenario uses the Recursive Feature Elimination (RFE) method. RFE systematically eliminates features, starting with an initial set of 25,088 features and removing 1,254 features per iteration. This iterative process continues until only the most influential features are retained. The goal is to determine the most effective subset of features that contribute significantly to the model's accuracy. This step is crucial for optimizing the model's performance, reducing complexity, and improving computational efficiency. RFE utilizes a Random Forest Regressor to evaluate the importance of each feature, ensuring that only the most relevant features are selected for the final model training with XGBoost. This methodical reduction helps pinpoint the optimal number of features, enhancing the model's predictive power and efficiency. The outcomes of this scenario are evaluated using RMSE, MAE, and R^2 metrics, with detailed results presented in Table 8.

Table 8. Evaluation Result of the RFE

Top-N Features	RMSE	MAE	R^2
500	6723.4443	2283.7496	0.7598
1000	6841.4536	2333.2321	0.7513
1500	7008.4801	2401.8873	0.7390
2000	7012.7800	2387.5935	0.7387
2500	7047.5720	2415.8216	0.7361
3000	6934.5124	2390.1401	0.7445
3500	7057.4485	2432.4566	0.7354
4000	7076.0014	2434.1331	0.7340
4500	7053.6122	2422.0173	0.7356
5000	7056.8830	2447.1458	0.7354

Based on Table 8, this model shows a consistent trend across different subsets of features, ranging from 500 to 5000 features. The RMSE and MAE metrics display a relatively narrow range of values, indicating

stable error rates regardless of the number of features used. The RMSE begins at 6723.4443 for 500 features and shows minor fluctuations throughout the feature set, reaching a peak at 7076.0014 for 4000 features before slightly tapering off to 7056.8830 for 5000 features. Similarly, the MAE starts at 2283.7496, gradually increasing as more features are incorporated, with a peak value of 2447.1458 for 5000 features. The R^2 value changes slightly as the number of features increases. Starting from 0.7598 with 500 features, the R^2 shows a general decrease and then stabilizes around 0.7354 when the number of features reaches 5000. This trend suggests that adding additional features does not significantly improve the model's ability to explain variations in the data. In other words, despite the increase in the number of features used, there is no proportional improvement in the accuracy of the model. Table 9 presents the results from a 5-fold cross-validation analysis using Recursive Feature Elimination (RFE) for feature selection in the model. The analysis indicates that as the number of features increases from 500 to 5000, the RMSE and R^2 values experience slight fluctuations but do not show substantial improvements, suggesting a plateau in performance gains with additional features. The overall averages for RMSE and MAE are 6651.62 and 2297.57, respectively, displaying a consistent model performance with minor variability as indicated by their standard deviations. The R^2 value averages at 0.7673, maintaining a relatively stable predictive accuracy across different subsets of features. Additionally, the total computation time for these analyses is noted at 1596.82 seconds, indicating the computational effort required as more features are considered in the model.

Table 9. Cross Validation RFE

Top-N Features	RMSE (Average ± Std)	MAE (Average ± Std)	R^2 (Average ± Std)	Time
500	6594.1985 ± 226.8203	2301.8579 ± 117.8836	0.7714 ± 0.0118	37.24s
1000	6602.9910 ± 367.0854	2273.5961 ± 150.5800	0.7705 ± 0.0209	66.02s
1500	6630.0703 ± 385.7827	2287.4550 ± 156.0974	0.7686 ± 0.0225	95.82s
2000	6676.8039 ± 384.1197	2302.1001 ± 139.4578	0.7656 ± 0.0196	125.09s
2500	6632.4630 ± 349.2750	2289.3591 ± 148.3073	0.7687 ± 0.0174	154.05s
3000	6677.4986 ± 340.0591	2307.4626 ± 155.6805	0.7657 ± 0.0162	180.49s
3500	6649.3519 ± 296.4507	2292.5783 ± 133.8495	0.7675 ± 0.0162	204.80s
4000	6669.8537 ± 314.9440	2304.0193 ± 140.5359	0.7660 ± 0.0185	226.27s
4500	6702.6080 ± 347.4423	2314.4365 ± 155.1583	0.7639 ± 0.0171	248.43s
5000	6680.3773 ± 347.6081	2302.8263 ± 148.1947	0.7655 ± 0.0168	258.61s
Overall Average	6651.62 ± 335.96	2297.57 ± 144.57	0.7673 ± 0.0177	
Total Time				1596.82s

3.5. Discussion

This study evaluated the XGBoost model using four feature selection methods: baseline, Feature Importance, Information Gain, and Recursive Feature Elimination (RFE), with cross-validation ensuring stability across data splits. RFE was the standout method, systematically reducing features from 25,088 to between 500 and 5000 and demonstrating superior stability and reliability with the best R^2 values and minimal variations in RMSE and MAE.

The average performance metrics for RFE were an RMSE of 6651.62, an MAE of 2297.57, and an R^2 of 0.7673, which underscore its efficiency in optimizing both the model's complexity and computational demands. Detailed evaluation metrics from Table 8 show that the RFE model began with an RMSE of 6723.4443 for 500 features and experienced slight variations, peaking at 7076.0014 for 4000 features before slightly decreasing to 7056.8830 for 5000 features. The MAE started at 2283.7496 and modestly increased to 2447.1458 for 5000 features. The R^2 began at 0.7598 for 500 features and demonstrated minimal variation, stabilizing around 0.7354 for 5000 features. The results confirm RFE's effectiveness in identifying and retaining the most impactful features, thus optimizing the model's performance without unnecessary complexity. Despite its advantages, RFE requires significant computational resources and time, especially as the feature set size increases. In this study, we implemented a step reduction of 5% of the total features to mitigate extensive computational demands. This approach was chosen to reduce the number of iterations and, consequently, the resources and time required. Using smaller step sizes, such as 100 features or fewer, would increase the number of iterations significantly, thus escalating the computational burden. This limitation is crucial for applications where resource constraints are a significant consideration, and it suggests a potential trade-off between model refinement and practical feasibility. The baseline model showed less accuracy, and though Feature Importance and Information Gain improved over the baseline, they did not reach the performance levels of RFE. This study affirms the importance of consistent model performance across varied data subsets, suggesting RFE's suitability for practical applications where stability and efficiency are crucial. Future research might explore combining RFE with other techniques to enhance performance in complex data scenarios.

In comparing the performance of the Recursive Feature Elimination (RFE) enhanced XGBoost model to other studies. Firstly, the RFE method achieved an R^2 of 0.7673, which, while effective for the datasets used in this study (GEE and drone imagery), falls short of the 0.89 R^2 achieved by a study utilizing the Boruta method with Landsat 8 OLI data. This discrepancy suggests that Boruta may be better at capturing relevant features specific to the Landsat dataset, providing a more accurate model for that particular type of imagery [10]. Conversely, when compared to another study that employed XGBoost with Gradient Boosting for feature selection on Sentinel-1 and Sentinel-2 datasets, which reported an R^2 of 0.59, the RFE method shows superior performance, indicating its potential better suitability for the imagery types analyzed in this research [11]. Additionally, the RFE method outperforms the Genetic Algorithm used with XGBoost in another study, which achieved an R^2 of 0.57 in predicting soil organic carbon content using various datasets [12]. This highlights RFE's efficiency in feature reduction and its capability to enhance predictive accuracy more effectively than some other feature selection methods in similar contexts.

To enhance the model's quality and broaden this study's scope, future research should consider several avenues. Firstly, experimenting with different feature extraction methods could provide valuable insights into how alternative approaches compare to those employed in this study. Additionally, integrating the Recursive Feature Elimination (RFE) method with other models to become hybrid or embedded models could potentially amplify the predictive accuracy and robustness of the results. Another promising direction could involve applying other high-dimensional data suitable models like Boruta or genetic algorithms as alternative wrapper methods to assess their effectiveness against the current methodologies. Such investigations would not only address the limitations observed but also optimize the performance of the XGBoost model applied to diverse imagery datasets like GEE and Drones, potentially leading to more refined and accurate environmental analyses.

4. CONCLUSION

This study evaluates the impact of feature selection on the performance of the XGBoost model integrated with VGG16 feature extraction for processing image datasets from GEE and drone sources, where the model is organized in four different scenarios: without feature selection, using Feature Importance, using Information Gain and using Recursive Feature Elimination. The experimental results have clearly shown that the application of feature selection significantly improves the accuracy and efficiency of the model. The Recursive Feature Elimination (RFE) method emerged as the most effective strategy in this study, outperforming Feature Importance and Information Gain. RFE demonstrated its capability to identify and retain impactful features, optimizing model complexity without compromising accuracy. However, as a wrapper method, RFE requires more computational resources and processing time, making Information Gain and Feature Importance practical alternatives for applications with limited resources, albeit with a trade-off in accuracy. These findings emphasize the importance of feature selection in improving machine learning model performance, especially for high-dimensional and complex environmental datasets. Feature selection facilitates the removal of irrelevant features, reducing dataset dimensionality and improving computational efficiency while enhancing prediction accuracy and preventing overfitting. The study highlights the contributions of all three feature selection methods in optimizing the efficiency and accuracy of the XGBoost model, underscoring their relevance for environmental data analysis. Future research can build on this study by exploring alternative feature extraction methods and integrating Recursive Feature Elimination (RFE) with different feature selection techniques to enhance predictive performance and robustness. Additionally, evaluating sophisticated feature selection methods like Boruta or genetic algorithms could enhance the understanding and handling of high-dimensional datasets. These efforts aim to refine machine learning applications to environmental data, particularly for complex datasets like those from GEE and drones, potentially leading to more accurate environmental analyses.

REFERENCES

[1] W. H. Zeng, S. D. Zhu, Y. H. Luo, W. Shi, Y. Q. Wang, and K. F. Cao, "Aboveground biomass stocks of species-rich natural forests in southern China are influenced by stand structural attributes, species richness and precipitation," *Plant Divers*, vol. 46, no. 4, pp. 530–536, Jul. 2024, https://doi.org/10.1016/j.pld.2024.04.012.

[2] A. Raihan, R. A. Begum, M. N. M. Said, and J. J. Pereira, "Assessment of carbon stock in forest biomass and emission reduction potential in Malaysia," *Forests*, vol. 12, no. 10, Oct. 2021, https://doi.org/10.3390/f12101294.

[3] J. H. Lee, J. G. Lee, S. T. Jeong, H. S. Gwon, P. J. Kim, and G. W. Kim, "Straw recycling in rice paddy: Trade-off between greenhouse gas emission and soil carbon stock increase," *Soil Tillage Res*, vol. 199, p. 104598, May 2020, https://doi.org/10.1016/J.STILL.2020.104598.

[4] A. A. Dar and N. Parthasarathy, "Patterns and drivers of tree carbon stocks in Kashmir Himalayan forests: implications for climate change mitigation," *Ecol Process*, vol. 11, no. 1, p. 58, 2022, https://doi.org/10.1186/s13717-022-00402-z.
[5] D. D. T. L. Dayathilake, E. Lokupitiya, and V. P. I. S. Wijeratne, "Estimation of Soil Carbon Stocks of Urban Freshwater Wetlands in the Colombo Ramsar Wetland City and their Potential Role in Climate Change Mitigation," *Wetlands*, vol. 41, no. 2, Feb. 2021, https://doi.org/10.1007/s13157-021-01424-7.
[6] D. Rajasugunasekar, A. K. Patel, K. B. Devi, A. Singh, P. Selvam, and A. Chandra, "An Integrative Review for the Role of Forests in Combating Climate Change and Promoting Sustainable Development," *International Journal of Environment and Climate Change*, vol. 13, no. 11, pp. 4331–4341, 2023, http://classical.goforpromo.com/id/eprint/4861/.
[7] Z. Zhang, J. He, M. Huang, and W. Zhou, "Is Regulation Protection? Forest Logging Quota Impact on Forest Carbon Sinks in China," *Sustainability (Switzerland)*, vol. 15, no. 18, Sep. 2023, https://doi.org/10.3390/su151813740.
[8] L. Nel *et al.*, "InVEST Soil Carbon Stock Modelling of Agricultural Landscapes as an Ecosystem Service Indicator," *Sustainability (Switzerland)*, vol. 14, no. 16, Aug. 2022, https://doi.org/10.3390/su14169808.
[9] S. R. Byrapu Reddy, P. Kanagala, P. Ravichandran, D. R. Pulimamidi, P. V. Sivarambabu, and N. S. A. Polireddi, "Effective fraud detection in e-commerce: Leveraging machine learning and big data analytics," *Measurement: Sensors*, vol. 33, p. 101138, Jun. 2024, https://doi.org/10.1016/J.MEASEN.2024.101138.
[10] S. Uniyal, S. Purohit, K. Chaurasia, S. S. Rao, and E. Amminedu, "Quantification of carbon sequestration by urban forest using Landsat 8 OLI and machine learning algorithms in Jodhpur, India," *Urban For Urban Green*, vol. 67, p. 127445, Jan. 2022, https://doi.org/10.1016/J.UFUG.2021.127445.
[11] J. Lei *et al.*, "Prediction of soil organic carbon stock combining Sentinel-1 and Sentinel-2 images in the Zoige Plateau, the northeastern Qinghai-Tibet Plateau," *Ecol Process*, vol. 13, no. 1, Dec. 2024, https://doi.org/10.1186/s13717-024-00515-7.
[12] M. Emadi, R. Taghizadeh-Mehrjardi, A. Cherati, M. Danesh, A. Mosavi, and T. Scholten, "Predicting and mapping of soil organic carbon using machine learning algorithms in Northern Iran," *Remote Sens (Basel)*, vol. 12, no. 14, Jul. 2020, https://doi.org/10.3390/rs12142234.
[13] Z. Ashani, "Comparative Analysis of Deepfake Image Detection Method Using VGG16, VGG19 and ResNet50," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 47, pp. 16–28, Dec. 2024, https://doi.org/10.37934/araset.47.1.1628.
[14] S. Kumar and H. Kumar, "Classification of COVID-19 X-ray images using transfer learning with visual geometrical groups and novel sequential convolutional neural networks," *MethodsX*, vol. 11, p. 102295, Dec. 2023, https://doi.org/10.1016/J.MEX.2023.102295.
[15] Y. Xia, S. Jiang, L. Meng, and X. Ju, "XGBoost-B-GHM: An Ensemble Model with Feature Selection and GHM Loss Function Optimization for Credit Scoring," *Systems*, vol. 12, p. 254, Dec. 2024, https://doi.org/10.3390/systems12070254.
[16] E. Sahin, "Assessing the predictive capability of ensemble tree methods for landslide susceptibility mapping using XGBoost, gradient boosting machine, and random forest," *SN Appl Sci*, vol. 2, Dec. 2020, https://doi.org/10.1007/s42452-020-3060-1.
[17] Y. Cai, J. Feng, Y. Wang, Y. Ding, Y. Hu, and H. Fang, "The Optuna–LightGBM–XGBoost Model: A Novel Approach for Estimating Carbon Emissions Based on the Electricity–Carbon Nexus," *Applied Sciences*, vol. 14, p. 4632, Dec. 2024, https://doi.org/10.3390/app14114632.
[18] N. Pudjihartono, T. Fadason, A. Kempa-Liehr, and J. O'Sullivan, "A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction," *Frontiers in Bioinformatics*, vol. 2, p. 927312, Dec. 2022, https://doi.org/10.3389/fbinf.2022.927312.
[19] P. V Agrawal and D. D. Kshirsagar, "Information Gain-based Feature Selection Method in Malware Detection for MalDroid2020," in *2022 International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN)*, pp. 1–5, 2022, https://doi.org/10.1109/ICSTSN53084.2022.9761336.
[20] K. Qu, J. Xu, Q. Hou, K. Qu, and Y. Sun, "Feature selection using Information Gain and decision information in neighborhood decision system," *Appl Soft Comput*, vol. 136, p. 110100, Mar. 2023, https://doi.org/10.1016/J.ASOC.2023.110100.
[21] H.-T. Wen, H.-Y. Wu, and K.-C. Liao, "Using XGBoost Regression to Analyze the Importance of Input Features Applied to an Artificial Intelligence Model for the Biomass Gasification System," *Inventions*, vol. 7, p. 126, Dec. 2022, https://doi.org/10.3390/inventions7040126.
[22] A. Velastegui-Montoya, N. Montalván-Burbano, P. Carrión-Mero, H. Rivera-Torres, L. Sadeck, and M. Adami, "Google Earth Engine: A Global Analysis and Future Trends," *Multidisciplinary Digital Publishing Institute (MDPI)*, vol.15, no. 14, p. 3675, 2023, https://doi.org/10.3390/rs15143675.
[23] M. Amani *et al.*, "Google Earth Engine Cloud Computing Platform for Remote Sensing Big Data Applications: A Comprehensive Review," *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 13, pp. 5326–5350, 2020, https://doi.org/10.1109/JSTARS.2020.3021052.
[24] C. Ning, H. Gan, M. Shen, and T. Zhang, "Learning-based padding: From connectivity on data borders to data padding," *Eng Appl Artif Intell*, vol. 121, p. 106048, May 2023, https://doi.org/10.1016/J.ENGAPPAI.2023.106048.

[25] C. Yu, P.-H. Hung, J.-H. Hong, and H.-Y. Chiang, "Efficient Max Pooling Architecture with Zero-Padding for Convolutional Neural Networks," in *IEEE 12th Global Conference on Consumer Electronics (GCCE)*, pp. 747–748, 2023, https://doi.org/10.1109/GCCE59613.2023.10315268.
[26] Y.-H. Huang, M. Proesmans, and L. Van Gool, "Padding Investigations for CNNs in Scene Parsing Tasks," in *2023 18th International Conference on Machine Vision and Applications (MVA)*, pp. 1–5, 2023, https://doi.org/10.23919/MVA57639.2023.10216084.
[27] F. Alrasheedi, X. Zhong, and P.-C. Huang, "Padding Module: Learning the Padding in Deep Neural Networks," *IEEE Access*, vol. 11, pp. 7348–7357, 2023, https://doi.org/10.1109/ACCESS.2023.3238315.
[28] S. Ullah and S.-H. Song, "Design of compensation algorithms for zero padding and its application to a patch based deep neural network," *PeerJ Comput Sci*, vol. 10, p. e2287, Aug. 2024, https://doi.org/10.7717/peerj-cs.2287.
[29] H. Hassan *et al.*, "Review and classification of AI-enabled COVID-19 CT imaging models based on computer vision tasks," *Comput Biol Med*, vol. 141, p. 105123, Feb. 2022, https://doi.org/10.1016/J.COMPBIOMED.2021.105123.
[30] K. Alomar, H. I. Aysel, and X. Cai, "Data Augmentation in Classification and Segmentation: A Survey and New Strategies," *J Imaging*, vol. 9, no. 2, Feb. 2023, https://doi.org/10.3390/jimaging9020046.
[31] R. Akter and M. I. Hosen, "CNN-based Leaf Image Classification for Bangladeshi Medicinal Plant Recognition," in *2020 Emerging Technology in Computing, Communication and Electronics (ETCCE)*, pp. 1–6, 2020, https://doi.org/10.1109/ETCCE51779.2020.9350900.
[32] W. Zeng, "Image data augmentation techniques based on deep learning: A survey," *Mathematical Biosciences and Engineering*, vol. 21, pp. 6190–6224, Dec. 2024, https://doi.org/10.3934/mbe.2024272.
[33] A. Moisés, I. Vitoria, J. J. Imas, and C. Zamarreño, "Data Augmentation Techniques for Machine Learning Applied to Optical Spectroscopy Datasets in Agrifood Applications: A Comprehensive Review," *Sensors*, vol. 23, p. 8562, Dec. 2023, https://doi.org/10.3390/s23208562.
[34] C. Xu, W. Liu, Y. Zheng, S. Wang, and C.-H. Chang, "An Imperceptible Data Augmentation Based Blackbox Clean-Label Backdoor Attack on Deep Neural Networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, pp. 1–14, Dec. 2023, https://doi.org/10.1109/TCSI.2023.3298802.
[35] P. Thanapol, K. Lavangnananda, P. Bouvry, F. Pinel, and F. Leprevost, "Reducing Overfitting and Improving Generalization in Training Convolutional Neural Network (CNN) under Limited Sample Sizes in Image Recognition," *International Conference on Information Technology (InCIT)*, pp. 300–305, 2020, https://doi.org/10.1109/InCIT50588.2020.9310787.
[36] G. Singh, K. Guleria, and S. Sharma, "A Transfer Learning-based Pre-trained VGG16 Model for Skin Disease Classification," in *2023 IEEE 3rd Mysore Sub Section International Conference (MysuruCon)*, pp. 1–6, 2023, https://doi.org/10.1109/MysuruCon59703.2023.10396942.
[37] W. Bakasa and S. Viriri, "VGG16 Feature Extractor with Extreme Gradient Boost Classifier for Pancreas Cancer Prediction," *J Imaging*, vol. 9, no. 7, Jul. 2023, https://doi.org/10.3390/jimaging9070138.
[38] Y. Bouchlaghem, Y. Akhiat, and S. Amjad, "Feature Selection: A Review and Comparative Study," *E3S Web of Conferences*, vol. 351, p. 1046, Dec. 2022, https://doi.org/10.1051/e3sconf/202235101046.
[39] B. Akyapı, "Machine learning and feature selection: Applications in economics and climate change," *Environmental Data Science*, vol. 2, 2023, https://doi.org/10.1017/eds.2023.36.
[40] S. Seydi, Y. Kanani-Sadat, M. Hasanlou, R. Sahraei, J. Chanussot, and M. Amani, "Comparison of Machine Learning Algorithms for Flood Susceptibility Mapping," *Remote Sens (Basel)*, vol. 15, p. 192, Dec. 2022, https://doi.org/10.3390/rs15010192.
[41] A. Shahin-Shamsabadi and J. Cappuccitti, "Proteomics and machine learning: Leveraging domain knowledge for feature selection in a skeletal muscle tissue meta-analysis," *Heliyon*, vol. 10, p. e40772, Jan. 2024, https://doi.org/10.1016/j.heliyon.2024.e40772.
[42] M. Büyükkeçeci and M. Okur, "A Comprehensive Review of Feature Selection and Feature Selection Stability in Machine Learning," *Gazi University Journal of Science*, vol. 36, Dec. 2022, https://doi.org/10.35378/gujs.993763.
[43] O. Salem, F. Liu, Y.-P. Chen, and X. Chen, "Ensemble Fuzzy Feature Selection Based on Relevancy, Redundancy, and Dependency Criteria," *Entropy*, vol. 22, p. 757, Dec. 2020, https://doi.org/10.3390/e22070757.
[44] H. Polat, O. Polat, and A. Çetin, "Detecting DDoS Attacks in Software-Defined Networks Through Feature Selection Methods and Machine Learning Models," *Sustainability*, vol. 12, no. 3, p/ 1035. 2020, https://doi.org/10.3390/su12031035.
[45] F. G. F. Niquini *et al.*, "Recursive Feature Elimination and Neural Networks Applied to the Forecast of Mass and Metallurgical Recoveries in A Brazilian Phosphate Mine," *Minerals*, vol. 13, no. 6, Jun. 2023, https://doi.org/10.3390/min13060748.
[46] F. Jiménez, G. Sánchez, J. Palma, L. Miralles-Pechuán, and J. A. Botía, "Multivariate Feature Ranking With High-Dimensional Data for Classification Tasks," *IEEE Access*, vol. 10, pp. 60421–60437, 2022, https://doi.org/10.1109/ACCESS.2022.3180773.
[47] T. Suryakanthi, "Evaluating the Impact of GINI Index and Information Gain on Classification using Decision Tree Classifier Algorithm*," *International Journal of Advanced Computer Science and Applications*, vol. 11, Jan. 2020, https://doi.org/10.14569/IJACSA.2020.0110277.
[48] T. Esaki, "Appropriate Evaluation Measurements for Regression Models," *Chem-Bio Informatics Journal*, vol. 21, pp. 59–69, 2021, https://doi.org/10.1273/cbij.21.59.

[49] N. Hassan, S. Sheikh Abdul Kadir, M. Husain, B. Satyanarayana, M. Ambak, and M. A.G., "Weight Prediction for Fishes in Setiu Wetland, Terengganu, using Machine Learning Regression Model," *BIO Web Conf*, vol. 73, Dec. 2023, https://doi.org/10.1051/bioconf/20237301007.
[50] Y. Fissha, H. Ikeda, H. Toriya, N. Owada, T. Adachi, and Y. Kawamura, "Evaluation and Prediction of Blast-Induced Ground Vibrations: A Gaussian Process Regression (GPR) Approach," *Mining*, vol. 3, no. 4, pp. 659–682, 2023, https://doi.org/10.3390/mining3040036.

BIOGRAPHY OF AUTHORS

I Made Darma Cahya Adyatma, is currently pursuing a bachelor's degree in computer science at Telkom University, Indonesia. Email, darmady@student.telkomuniversity.ac.id.

Erwin Budi Setiawan, is an Associate Professor at the School of Computing, Telkom University, Bandung, Indonesia. He has more than 10 years research and teaching experience in the field of Informatics, his academic interests include machine learning, people analytics, and social media analysis. Email, erwinbudisetiawan@telkomuniversity.ac.id.