Available online at http://www.iorajournal.org/index.php/ijgor/index

International Journal of Global Operations Research

e-ISSN: 2722-1016
p-ISSN: 2723-1739

Vol. 6, No. 2, pp. 124-129, 2025

Comparison of Grid Search and Random Search Effectiveness in Parameter Tuning on Electric Car Sentiment Analysis

Moch Panji Agung Saputra[1*], Muhammad Bintang Eighista Dwiputra[2]

[1]*Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Padjadjaran, Sumedang, Indonesia*
[2]*Department of Computer Science Education, Faculty of Mathematics and Natural Sciences Education, Universitas Pendidikan Indonesia, Bandung, Indonesia*

**Corresponding author email: moch16006@unpad.ac.id*

Abstract

The increasing use of electric cars in Indonesia has prompted many public discussions recorded on various digital platforms. This study aims to classify public sentiment towards the implementation of electric cars through comment analysis using the XGBoost Classifier model. The data used were obtained from the Kaggle platform, in the form of public comments that have gone through pre-processing stages, such as removing empty data, label encoding, and visualizing class distribution. Furthermore, the data was divided into training, validation, and test data using stratification techniques, and data imbalance was handled using the SMOTE method. Modeling was carried out using the XGBoost Classifier algorithm, then hyperparameter tuning was carried out using two approaches, namely Random Search and Grid Search. The parameters tested included *learning_rate, max_depth, n_estimators, subsample, colsample_bytree, gamma, alpha, and lambda*. The experimental results showed that the model without tuning produced an accuracy of 67%. After tuning, Random Search increased its accuracy to 68%, while Grid Search achieved the highest accuracy of 69%. Based on evaluation using precision, recall, f1-score, and accuracy metrics, tuning with Grid Search is proven to provide more optimal results compared to other methods. This study shows that systematic hyperparameter tuning can improve the performance of sentiment classification models.

Keywords: Electric car, sentiment analysis, xgboost classifier, random search, grid search, hyperparameter tuning.

1. Introduction

The rapid development of the electric vehicle industry in recent years has changed the paradigm of the automotive sector, presenting new challenges and opportunities for manufacturers and consumers. In addition to technical issues such as range, cost, and charging infrastructure, public perception of electric vehicles also plays a significant role in accelerating the adoption of this technology (Wijaya et al., 2024). Therefore, it is important for manufacturers and policymakers to understand public sentiment regarding electric vehicles in order to effectively respond to consumer needs and concerns.

With the increasing amount of data available on social media platforms, sentiment analysis has emerged as one of the most relevant techniques for assessing public opinion quickly and efficiently. Sentiment analysis is the process of processing and classifying text to determine the attitude or feeling contained in a statement, such as positive, negative, or neutral. In the context of electric vehicles, sentiment analysis can reveal how consumers respond to various aspects related to this product, including technology, price, government policies, and its impact on the environment (Aftab et al., 2023).

The application of sentiment analysis techniques to social media data faces a number of challenges, one of which is the class imbalance in the distribution of sentiment labels. The neutral sentiment class, for example, is often much smaller than the positive and negative sentiment classes. This imbalance can affect the performance of the sentiment analysis model, because the model tends to focus more on the majority class and ignores the minority class (Suandi et al., 2024). To overcome this problem, one technique that can be used is oversampling, which aims to balance the data distribution by increasing the number of samples from the minority class.

The application of machine learning in sentiment analysis requires the right hyperparameter settings to produce an optimal model. One way to find the ideal hyperparameter settings is through parameter tuning techniques, which aim

to find the best combination of model parameters that can improve prediction performance. Parameter tuning can be done using various methods, including Grid Search and Random Search (Jamaleddyn et al., 2023).

Grid Search is a method that performs a systematic parameter search, testing all combinations of a set of predetermined hyperparameters. This method tends to take longer because the search is carried out thoroughly. On the other hand, Random Search selects a random combination of parameters from a wider search space, which can often provide good results more efficiently in terms of time and resources (Belete and Huchaiah, 2022). Although both aim to optimize the model, the selection of the right tuning method is highly dependent on the characteristics of the data and the problem at hand.

This study aims to evaluate the comparison between Grid Search and Random Search in tuning the parameters of sentiment analysis models on electric car sentiment data. This study also examines how oversampling techniques can affect model performance in overcoming class imbalance, especially in the neutral class which has less data. The results of this study are expected to contribute to the selection of more efficient and effective tuning methods in the context of sentiment analysis on imbalanced data.

2. Methodology

2.1. Data Collection

The data used in this study were obtained from the *Kaggle* platform. The dataset used contains a collection of comments from Indonesian people regarding the implementation of electric car incentives. The comments were taken from various videos discussing the topic "Implementation of Electric Car Incentives in Indonesia" on social media platforms. Each entry in the dataset consists of two main components, namely the comment text that has gone through a text cleaning process and a sentiment label that indicates the sentiment category of the comment. Sentiment labels are classified into three classes, namely negative, neutral, and positive. The comments used have been selected to be relevant to the issue of electric cars and are worthy of further analysis in the context of public opinion on government policies.

2.2. Data Pre-processing

Data pre-processing is an important stage to ensure data quality before entering the modeling process. The pre-processing steps carried out in this study are as follows:

1) Checking and Deleting Missing Values
   The data obtained from Kaggle is first checked to detect the presence of empty values (missing values) in important columns, namely the *text_cleaning* and *sentiment* columns. Rows containing empty values are then deleted to maintain the consistency and quality of the data that will be used in the next stage.
2) Sentiment Label Mapping
   After the data is cleaned of missing values, the sentiment labels that were originally in text form, namely "negative", "neutral", and "positive", are converted into numeric format to facilitate the modeling process. The mappings used are: negative (0), neutral (1), and positive (2) (Pipin et al., 2023).

2.3. Tokenization and Vectorization

At this stage, the process of converting text data into a numeric representation that can be understood by the machine learning model is carried out. The steps taken are as follows:

1) Text Tokenization
   Comments in the *text_cleaning* column are processed using the tokenization technique. Each comment is converted into a sequence of tokens (words) based on a dictionary formed from the entire data. In this study, the Tokenizer from the TensorFlow library was used with a maximum number of words (*num_words*) of 10,000 and the use of special tokens for unknown words (*out-of-vocabulary tokens*).
2) Padding and Truncating
   The length of the token sequence is adjusted using the padding and truncating techniques. Padding is done at the end of the sequence (*post-padding*) so that all inputs have a uniform length, namely 100 tokens per comment.
3) Vectorization with TF-IDF
   For modeling using the XGBoost algorithm, text vectorization is carried out using the Term Frequency-Inverse Document Frequency (TF-IDF) technique. TF-IDF is used to convert text into numeric vectors based on word frequency in the document and the entire corpus (Havrlant and Kreinovich, 2017).

2.4. Data Splitting

After the data has gone through the cleaning and vectorization process, the next step is to divide the data into three subsets, namely training data, validation data, and test data. Training data consists of 70% of the total data and is used for the model training process. Furthermore, validation data is taken as much as 15% of the total data, which functions to evaluate and optimize model performance during the training process so that overfitting does not occur. The remaining 15% of the data is used as test data to measure the final model performance.

2.5. Handling Data Imbalance

Based on the initial analysis of the class distribution, an imbalance in the amount of data between classes was found, where the "neutral" class has much less data than the other classes. The total class distribution of each label will be shown in Figure 1.

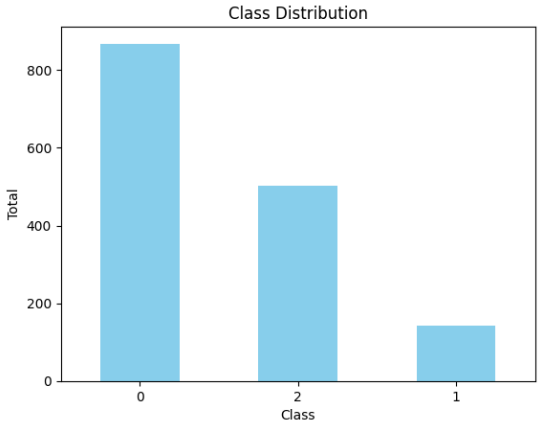

Figure 1: Class distribution

To overcome this problem, an oversampling technique is applied using the Synthetic Minority Over-sampling Technique (SMOTE) method. SMOTE works by generating new synthetic samples for the minority class based on interpolation from existing samples (Soltanzadeh and Hashemzadeh, 2021).. By implementing SMOTE, the amount of data in each class becomes more balanced, so that the model built is expected to be able to learn better and reduce bias towards the majority class.

2.6. Parameter Modeling and Tuning

In this study, the model used is the XGBoost Classifier with a multi-class classification scheme. After building the initial model, a hyperparameter tuning process is carried out to improve model performance. Tuning is done by comparing two approaches, namely Random Search and Grid Search. The purpose of this tuning is to evaluate the impact of each method on the final results of the model. In both methods, several XGBoost parameters are tested with various combinations of values. The scope of parameters tried in each approach is shown in Table 1.

Table 1: Hyperparameter search space in random search and grid search

Metode	Nama Parameter	Nilai yang Dicoba
Random Search	*learning_rate*	[0.01, 0.05, 0.1, 0.2]
	max_depth	[3, 5, 6, 7, 8]
	n_estimators	[100, 200, 300, 500]
	subsample	[0.6, 0.7, 0.8, 0.9, 1.0]
	colsample_bytree	[0.6, 0.7, 0.8, 0.9, 1.0]
	gamma	[0, 0.1, 0.2, 0.3]
	alpha	[0, 0.1, 0.2, 0.3]
	lambda	[0, 0.1, 0.5]
Grid Search	*learning_rate*	[0.01, 0.1]
	max_depth	[3, 6, 9]
	n_estimators	[50, 100, 150]
	subsample	[0.8, 1.0]
	colsample_bytree	[0.8, 1.0]

2.7. Model Evaluation

Model evaluation is conducted to measure the classification performance against test data. The metrics used in this evaluation include Accuracy, Precision, Recall and F1-Score with the formula (Obi, 2023):

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \times 100\% \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (3)$$

$$F1 - Score = \frac{TP}{TP + FN} \times 100\% \quad (4)$$

where: TP: True Positive; TN: True Negative; FP: False Positive; and FN: False Negative

3. Results and Discussion

3.1. Before Tuning

Before hyperparameter tuning was performed, the XGBoost Classifier model was first tested on the test data to obtain a performance baseline. Which will be displayed in Figure 2:

	precision	recall	f1-score	support
negatif	0.72	0.79	0.75	130
netral	0.29	0.18	0.22	22
positif	0.67	0.61	0.64	75
accuracy			0.67	227
macro avg	0.56	0.53	0.54	227
weighted avg	0.66	0.67	0.66	227

Figure 2: Evaluation results before tuning

Based on the evaluation results, the model achieved an accuracy of 67%. When viewed from each class, the negative class showed the best performance with a precision value of 0.72, a recall of 0.79, and an f1-score of 0.75. This shows that the model is quite good at identifying comments labeled negative.

The model's performance in the neutral class is relatively low, with a precision of only 0.29, a recall of 0.18, and an f1-score of 0.22. This indicates that the model has difficulty in recognizing neutral comments, possibly due to the smaller amount of neutral data compared to other classes, or because the characteristics of neutral text are less significantly different from other classes.

In the positive class, the model obtained a precision value of 0.67, a recall of 0.61, and an f1-score of 0.64, indicating moderate performance. When viewed from the macro average, the precision value is 0.56, recall is 0.53, and f1-score is 0.54. This value reflects the average performance of the model in all classes without taking into account the imbalance in the amount of data. While the weighted average, which takes into account the proportion of data for each class, shows a precision value of 0.66, recall is 0.67, and f1-score is 0.66.

3.2. Tuning Random Search

After the tuning process using the Random Search method with 100 parameter combinations and 3-fold cross-validation, the results will be displayed in Figure 3:

	precision	recall	f1-score	support
0	0.71	0.85	0.77	130
1	0.33	0.09	0.14	22
2	0.65	0.57	0.61	75
accuracy			0.68	227
macro avg	0.56	0.50	0.51	227
weighted avg	0.65	0.68	0.66	227

Figure 3: Random search tuning evaluation results

As seen in Figure 3, the results show that the best model configuration has an accuracy of 68.65%. This shows an improvement compared to the model before tuning which only achieved an accuracy of 67%. From the model evaluation results, the performance in the negative class showed an increase with a precision of 0.71, a recall of 0.85, and an f1-score of 0.77. This indicates that the model is getting better at detecting negative comments. In the neutral class, although there are still challenges, there was a slight improvement in the precision value to 0.33, although the recall value remained low (0.09). Meanwhile, for the positive class, the precision value reached 0.65 and the f1-score was 0.61, which was relatively stable compared to before tuning. Overall, the macro average value showed a precision of 0.56, a recall of 0.50, and an f1-score of 0.51. While for the weighted average, the model recorded a precision of 0.65, a recall of 0.68, and an f1-score of 0.66. The best parameters obtained from the results of Tuning Random Search:

Table 2: Optimal random search parameters

Parameter	Optimal value
subsample	0.9
n_estimators	100
max_depth	7
learning_rate	0.1
lambda	0.5
gamma	0.2
colsample_bytree	1.0
alpha	0.3

These results indicate that hyperparameter tuning using Random Search has a positive impact on improving model performance, especially in classifying negative and positive comments.

3.3. Tuning Grid Search

The next tuning process is carried out using the Grid Search method, which performs a systematic search on a combination of predetermined parameters which will be shown in Figure 4:

	precision	recall	f1-score	support
negatif	0.71	0.85	0.78	130
netral	0.29	0.09	0.14	22
positif	0.69	0.59	0.63	75
accuracy			0.69	227
macro avg	0.56	0.51	0.52	227
weighted avg	0.66	0.69	0.67	227

Figure 4: Grid search tuning evaluation results

The best results show that the model achieves an accuracy of 69% on the test data, slightly higher than the tuning results using Random Search. Based on the evaluation results, the model performance in the negative class increased with a precision of 0.71, a recall of 0.85, and an f1-score of 0.78. The model is quite effective in classifying negative comments. However, for the neutral class, the precision value is only 0.29 and the recall is 0.09, indicating that the model still has difficulty in detecting neutral comments accurately. In the positive class, the model managed to achieve a precision of 0.69 and an f1-score of 0.63, slightly better than Random Search.

Table 3: Optimal grid search parameters

Parameter	Optimal value
colsample_bytree	0.8
learning_rate	0.1
max_depth	9
n_estimators	100
subsample	0.8

Tuning using Grid Search has a positive impact on model performance, especially in identifying negative and positive comments better than the model without tuning.

3.4. Discussion

After conducting a series of experiments, it can be concluded that parameter tuning has an impact on improving the performance of the XGBoost Classifier model in classifying public comment sentiments related to the implementation of electric car incentives in Indonesia. The initial model without tuning produced an accuracy of 67%, with the highest f1-score in the negative class (0.75) and quite low performance in the neutral class (f1-score of 0.22). This imbalance in performance between classes indicates that the model still has difficulty in detecting neutral sentiment accurately.

After tuning using the Random Search method, there was a slight increase in accuracy to 68%. The optimal parameters found through Random Search helped the model improve recall in the negative class, but the f1-score for the neutral class remained low (0.14). Overall, the weighted average f1-score was recorded at 0.66, indicating a small improvement compared to the initial model. Meanwhile, tuning using Grid Search gave the best results with an accuracy of 69%. With the optimal parameters obtained, the model showed a higher f1-score for the negative and positive classes compared to the Random Search model. The F1-score in the negative class reaches 0.78, while in the positive class it reaches 0.63. Although the performance for the neutral class is still not satisfactory, the increase in the macro average and weighted average values shows that Grid Search is more effective in improving the performance balance between classes compared to Random Search.

4. Conclusion

The model without tuning produces an accuracy of 67%, with uneven performance especially in the neutral sentiment class. After hyperparameter tuning, there is an increase in performance. Tuning with the Random Search method increases accuracy to 68%, but has not provided significant improvements in neutral sentiment detection. Tuning using Grid Search produces the best performance with an accuracy of 69%. In addition, the f1-score values for the negative and positive classes are also higher than the results from Random Search. This shows that Grid Search not only improves overall accuracy but also helps balance performance between classes. Thus, it can be concluded that hyperparameter tuning using Grid Search produces the best model in this study, in terms of accuracy and balance of performance between classes. For further development, it is recommended to explore additional techniques, such as further data balancing or feature enrichment, to improve accuracy in the neutral sentiment class.

References

Aftab, F., Bazai, S. U., Marjan, S., Baloch, L., Aslam, S., Amphawan, A., & Neo, T. K. (2023). A comprehensive survey on sentiment analysis techniques. *International Journal of Technology*, *14*(6), 1288-1298.

Belete, D. M., & Huchaiah, M. D. (2022). Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results. *International Journal of Computers and Applications*, *44*(9), 875-886.

Havrlant, L., & Kreinovich, V. (2017). A simple probabilistic explanation of term frequency-inverse document frequency (tf-idf) heuristic (and variations motivated by this explanation). *International Journal of General Systems*, *46*(1), 27-36.

Jamaleddyn, I., El Ayachi, R., & Biniz, M. (2023). Optimization of Machine Learning Algorithms Through Hyper-parameter Tuning Applied for the classification of Arabic news.

Obi, J. C. (2023). A comparative study of several classification metrics and their performances on data. *World Journal of Advanced Engineering Technology and Sciences*, *8*(1), 308-314.

Pipin, S. J., Sinaga, F. M., Winardi, S., & Hakim, M. N. (2023). Sentiment Analysis Classification of ChatGPT on Twitter Big Data in Indonesia Using Fast R-CNN. *J. MEDIA Inform. BUDIDARMA*, *4*, 2137-2148.

Soltanzadeh, P., & Hashemzadeh, M. (2021). RCSMOTE: Range-Controlled synthetic minority over-sampling technique for handling the class imbalance problem. *Information Sciences*, *542*, 92-111.

Suandi, F., Anam, M. K., Firdaus, M. B., Fadli, S., Lathifah, L., Yumami, E., ... & Hasibuan, A. Z. (2024, December). Enhancing Sentiment Analysis Performance Using SMOTE and Majority Voting in Machine Learning Algorithms. In *7th International Conference on Applied Engineering (ICAE 2024)* (pp. 126-138). Atlantis Press.

Wijaya, H., Hostiadi, D. P., & Triandini, E. (2024). Optimization of XGBoost Algorithm Using Parameter Tunning in Retail Sales Prediction. *Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI*, *13*(3), 769-786.