Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943


# Hyperparameter Optimization for Convolutional Neural Network-Based Sentiment Analysis

Retno Kusumaningrum[1,*], Asyraf Humam Arafifin[2], Selvi Fitria Khoerunnisa[3], Priyo Sidik Sasongko[1], Panji Wisnu Wirawan[1], Muhammad Syafrudin[4]

[1] Department of Informatics, Faculty of Science and Mathematics, Universitas Diponegoro, Semarang Selatan, Semarang City, Central Java 50241, Indonesia
[2] SRIN - Samsung R&D Indonesia, Kota Jakarta Pusat, Daerah Khusus Ibukota Jakarta 10210, Indonesia
[3] School of Postgraduate Studies, Universitas Diponegoro, Semarang Selatan, Semarang City, Central Java 50241, Indonesia
[4] Department of Artificial Intelligence and Data Science, Sejong University, Gwangjin District, Seoul 05006, South Korea

**ARTICLE INFO**

***Article history:***
Received 3 October 2023
Received in revised form 23 August 2024
Accepted 27 August 2024
Available online 22 September 2024

***Keywords:***

Hyperparameter optimization; Bayesian optimization; Convolutional neural network; Sentiment analysis

**ABSTRACT**

The rapid development of digital technology brings about the importance of product or service opinions as a data source in business intelligence. Sentiment analysis is a business intelligence tool that can help gauge market trends by analysing the available opinions. Various studies have been developed to solve sentiment analysis problems, from selecting feature engineering techniques to selecting sentiment classification models based on classical or deep learning methods. However, these studies still select hyperparameters on a trial basis, which poses problems in terms of time and not optimal performance. Therefore, this research proposes the implementation of Bayesian Optimization to automate the selection of hyperparameters in sentiment analysis based on a Convolutional Neural Network (CNN). The results show that implementing the Bayesian Optimization method has the lowest computation time compared to Random and Grid Searches, which is only 175.746765 seconds, in the fifth trial. In addition, it made a reasonably large cut in terms of times compared to manual trial-based hyperparameter tuning that needs 540 trials. In the model assessment process, implementing Bayesian Optimization gives the highest values on Precision and Specificity, which are 0.8168 and 0.7176, respectively. Thus, implementing Bayesian Optimization is the right choice for sentiment analysis tasks since the precision of predicting negative sentiment classes is vital in hospitality business intelligence, especially the use of related information for product improvement or hotel service improvement. The implementation of Bayesian Optimization not only applied as a reliable hyperparameter selection technique for classification tasks but also regression prediction tasks.

## 1. Introduction

The rapid growth of technology causes an increase in the duration of internet usage by users. It has resulted in a change in the structure of many industries by giving rise to new rules and roles for

* *Corresponding author.*
*E-mail address: retno@live.undip.ac.id*

https://doi.org/10.37934/araset.53.1.4456

distributing information that influences people's purchases [1]. It is an ample opportunity for the e-commerce industry, especially Online Travel Agents (OTA). OTA is an e-commerce that allows ordering hotel, train or plane tickets and other facilities [2]. OTAs make it easy to plan and book trips online and help promote sustainable tourism by directing tourists to less visited areas with excellent tourism potential [3]. Thus, OTA becomes one of the critical forces in economic growth by expanding access to the tourism market and providing business opportunities for business actors engaged in the tourism sector. Reviews are used as a measure of customer satisfaction and a source of information for evaluating services that require improvement. Reviews can also be used as a reference for service changes according to developing market trends. For tourists, reviews are helpful as a source of hotel information according to their preferences.

Automating review analysis must be done to avoid human error in analysing and concluding reviews with large data volumes. The technology for automatically analysing reviews is usually referred to as sentiment analysis. Sentiment analysis collects and analyses product and service points of view expressed through reviews. The review data will then be collected to determine whether the polarity is positive or negative. *Sentiment analysis* is a business intelligence tool that can help gauge market trends by analysing prevailing opinions. It is also a practical tool for decision-makers to extract insights from large amounts of data [4]. Sentiment analysis will impact the development of OTA as brand monitoring, competitive intelligence, and identifying influencers. It makes research on sentiment analysis urgent in the tourism sector, especially in the hospitality sector.

Various studies have been developed to solve sentiment analysis problems, from selecting feature engineering techniques to selecting classification models based on classical or deep learning methods [5-9]. Previous research by Nawangsari *et al.,* [10] regards sentiment analysis of Indonesian-language hotels using the Convolutional Neural Network (CNN) as a classification algorithm and Word2Vec as a word embedding. The deep learning method is unable to analyse input data in strings. Therefore, word embedding is needed to convert text into vectors. Word2Vec is a word embedding that can represent characters as vectors. Then, as a classification method, CNN uses a convolution filter to capture the sequential pattern of the vector given by the word embedding. CNN does not depend on the structure of the language so that it can extract the semantic features attached to sentiment [11].

The research has produced good accuracy. However, this study still selects hyperparameters based on trials, which causes problems in terms of time and performance that could be more optimal, in addition to the high computational costs of manually searching for hyperparameters. *Hyperparameters* are parameters that can be set by the user whose values control the learning process [12]. Hyperparameter settings are critical in model optimization. The best hyperparameter combination can minimize the loss function and maximize model performance [13]. Hyperparameter optimization is selecting the optimal set of hyperparameters for the learning process. Hyperparameter optimization can solve the problem of time usage and optimize model performance in manual hyperparameter searches. This method will get the best model by reaching the optimum point without trying all possibilities. Therefore, this study proposes the implementation of hyperparameter optimization to automate the selection of hyperparameters in CNN-based sentiment analysis.

## 2. Methodology

This section describes detailed information about stages in the research methodology. Figure 1 presents the block diagram of the proposed study. A detailed description of each stage can be seen in the following sub-sections.

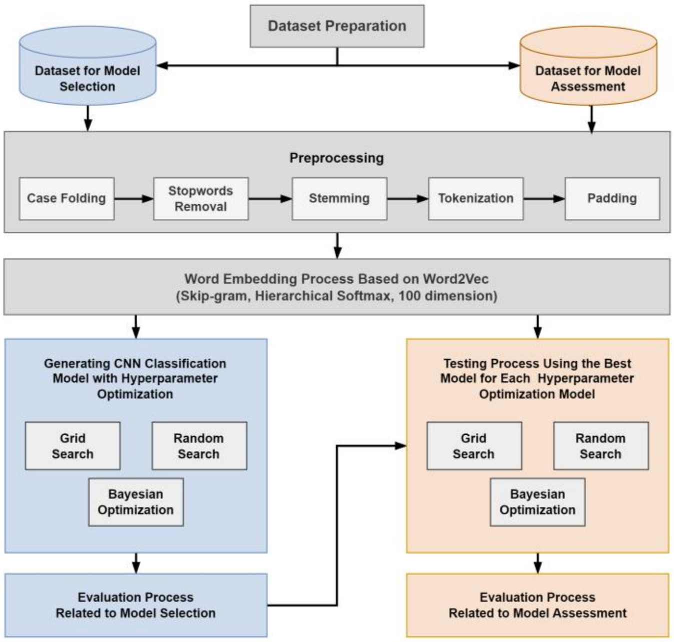


**Fig. 1.** Research methodology

### *2.1 Dataset Preparation*

Two datasets are employed in this study: the dataset for the model selection process and the dataset for the model assessment process. The first dataset was obtained from a previous study by Nawangsari *et al.*, [10]. The second dataset was obtained by manually crawling from an OTA. The crawling process was conducted towards 5 hotels and 50 reviews for each hotel. A detailed explanation of the description can be seen in section 3.

### *2.2 Preprocessing*

Five preprocessing steps are implemented in this study, including case folding, stopwords removal, stemming, tokenization, and padding. Preprocessing is implemented for both model selection and model assessment datasets. This process's output is clean data ready to be implemented in the model.

Case folding is a standardization of the form of text data. This process changes all letters into lowercase. Stopword removal is performed to remove common words and many appearances because they are considered meaningless. Next, stemming is changing words into their original form by removing affixes [14]. Tokenization divides sentence into words based on space, tabs, or enter. That process includes removing unnecessary punctuation, space, and characters. Lastly, padding is the process of equalizing the length of each document. Padding is necessary because deep learning requires the same input length [15].

### *2.3 Word Embedding*

*Word embedding* is a word mapping technique based on existing dictionaries for those containing real numbers. Neural networks cannot accept strings as input, so words are converted into vector space. This study employed Word2Vec as the word embedding method. The advantage of Word2Vec is that it can capture the syntax and semantic meaning of natural language. This method will teach the representation of the group of words and groups of similar words with the same vector [10]. The output of Word2Vec is a low-dimensional vector space representing the word's semantic meaning. The employed parameters in this study are skip-gram architecture, 100 dimensions of word vector, and hierarchical SoftMax as an evaluation method.

### *2.4 Hyperparameter Optimization*

Hyperparameter optimization was conducted to get the best model, in other words, to reach the optimum point without having to try all possibilities. It saves time, which would take a long time if done manually one by one. This study uses three methods for hyperparameter optimization: Grid Search, Random Search and Bayesian Optimization.

The actual process of finding the optimal hyperparameter is to do the training process iteratively to find the best combination of parameters. The data used is 1 of 10 folds from k-fold cross-validation, and the iteration process will be limited to n repetitions. The types of hyperparameters that will be optimized are filters, kernels, and activation functions in the convolutional layer, dropouts in the fully connected layer, activation functions used in the output layer, and learning rate.

Grid Search is the most common and simplest hyperparameter method. This method performs a hyperparameter search for each combination to determine the combination that produces the best performance. The Grid Search algorithm is a brute force algorithm, namely a search algorithm without information, meaning that the Grid Search does not learn from previous iterations [16]. The Grid Search method performs a complete search and ensures that every combination has been compared. Grid Search is effective when the compared parameters are known or have a small combination of parameters. A slight change in some parameters dramatically affects the performance evaluation results.

Random Search is an alternative hyperparameter tunning algorithm for a Grid Search. If the Grid Search tries every possible combination of hyperparameters, then the Random Search will only take a sample of the combinations to run [17]. Each sample randomly selects a hyperparameter from a list of possible parameter distributions. The user determines the number of samples to be tested. *Random Search* is an independent algorithm that does not learn from previous iterations like Grid Search. However, the benefit of having Random Search is that Random Search data optimizes the hyperparameter space by ignoring less influential hyperparameters to save computation time. Although this approach may result in the risk of missing the optimal performance value, it performs better with the same number of calculations.

Bayesian optimization builds a probability-based objective function model, which is advantageous in providing detailed information regarding uncertainties arising from targets [18]. The objective function model is subsequently used to select the most likely hyperparameter to evaluate the actual objective function. Figure 2 shows the process flow of Bayesian Optimization. The optimization process begins with selecting a set of hyperparameter combinations randomly to be evaluated. The evaluation results will be used in the selection of the next hyperparameter. Initialization begins by taking an initial sample from the search space, then the deep learning model, in this case, CNN, as a black box problem. Initialize the model to generate curated values from the selected sample. The next step is to plot the black box function's results into Gaussian process regression. It begins by forming a straight line as the average of the predicted values and a field describing the predicted results' possible values. They were followed by an approximation process that runs using several sampling points, where these points are selected using the acquisition function, which is the next stage of this plotting process. The selection of influential hyperparameters will be based on changes in accuracy values. Steps 2 to 4 will continue to be repeated for the number of attempts the user specifies. When the number of trials has been met, Bayesian Optimization will take the optimal point from the predicted average graph to get the most optimal combinations.

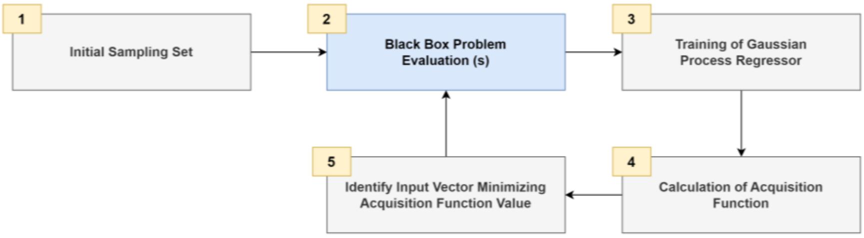


**Fig. 2.** Bayesian Optimization process flow

### *2.5 Generating CNN Classification Model with Hyperparameter Optimization*

CNN uses a convolutional filter to study the matching features in classifying text. For example, in the case of sentiment analysis, the convolutional filter will capture sequential patterns and extract semantic features attached to sentiment [11]. Text data input will be data with one dimension. Local text information is stored when the filter moves and features are extracted [19]. Figure 3 shows the representative proposed CNN architecture used in this study. The network starts from the input layer, embedding, one convolutional layer, flatten, dropout, fully connected layer, and output layer.

The model will be trained using CNN with three types of hyperparameter optimization: Grid Search, Random Search, and Bayesian Optimization. The training was performed using 10-fold cross-validation. The training process will be repeated ten times along with the validation process. The validation accuracy value of the model validation will be used to select the best classification model for each hyperparameter optimization method.

### *2.6 Testing Process using the Best Model for Each Hyperparameter Optimization Model*

Model assessment is the process of evaluating the predictive ability of the model. The model selection stage aims to find the best model representing the data and how well it generalizes it. The best model will be stored and used in the assessment process. Model assessment will see whether the model also has good performance if new data is included that has never been seen before. As previously explained, the assessment model employs new data, amounting to 250.

### *2.7 Evaluation Process for Both Model Selection and Model Assessment*

Evaluation for model selection will be based on accuracy and loss. Accuracy is the ratio of correctly predicted classes compared to the sum of all data calculation of accuracy using the help of the confusion matrix. The equation of accuracy can be seen in Eq. (1).

$$Accuracy = \frac{TP+TN}{TP+FN+FN+TN} \tag{1}$$

TP represents true positive, TN represents true negative, FP represents false positive, and FN represents false negative.

As for the assessment model section, this study does not only use accuracy metrics but also other metrics such as precision, recall or sensitivity, f1-score, and specificity. The formulas for these matrices can be seen in Eq. (2), Eq. (3), Eq. (4) and Eq. (5) respectively.

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

$$Recall = Sensitivity \ \frac{TP}{TP+FN} \quad (3)$$

$$F1 - Score = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (4)$$

$$Specificity = \frac{TN}{FP+TN} \quad (5)$$

## 3. Experimental Results and Analysis

### *3.1 Dataset and Experimental Environment*

The employed dataset has a balanced distribution for the positive sentiment class (1250 data) and the negative sentiment class (1250 data). The dataset was used for the model selection process. Hence, it is subsequently referred to as the model selection dataset. At the same time, the model testing process was performed based on a separate dataset of 10% from the previous dataset, namely 250 data. A Google Colab cloud platform with NVIDIA T4 Tensor Core GPUs is used in an experimental setting where the CNN was performed. Table 1 shows the example dataset for both datasets, i.e., model selection and model assessment datasets.

**Table 1**
Example Dataset

| Category | Reviews | Sentiment Polarity |
|---|---|---|
| Model Selection Dataset | *Tempatnya bersih, nyaman, sarapannya enak, viewnya bagus, pelayanannya ramah. Good place!* (The place is clean, comfortable, the breakfast is delicious, the view is good, the service is friendly. Good place!) | Positive |
| | *Ada beberapa perlu perbaikan, seperti contohnya AC yang mati, kran yang rusak dan pengharum ruangan yang perlu di maintance*. (There are a number of things that need improvement, such as, for example, a dead air conditioner, broken faucets and air freshener that need maintenance) | Negative |
| Model Assessment Dataset | *Resepsionis ramah, check in tidak lama, kamar nyaman dan bersih, makanannya enak.* (The receptionist is friendly, check-in is short, the room is comfortable and clean, the food is delicious.) | Positive |
| | *AC kurang dingin dan nunggu check in agak lama, kulkas mati tidak menyala* (The air conditioner is not cold enough and waiting for check-in is a bit long, the refrigerator is dead and doesn't turn on | Negative |

### *3.2 Experimental Scenarios*

A detailed illustration of the experimental scenario performed can be seen in Figure 3.

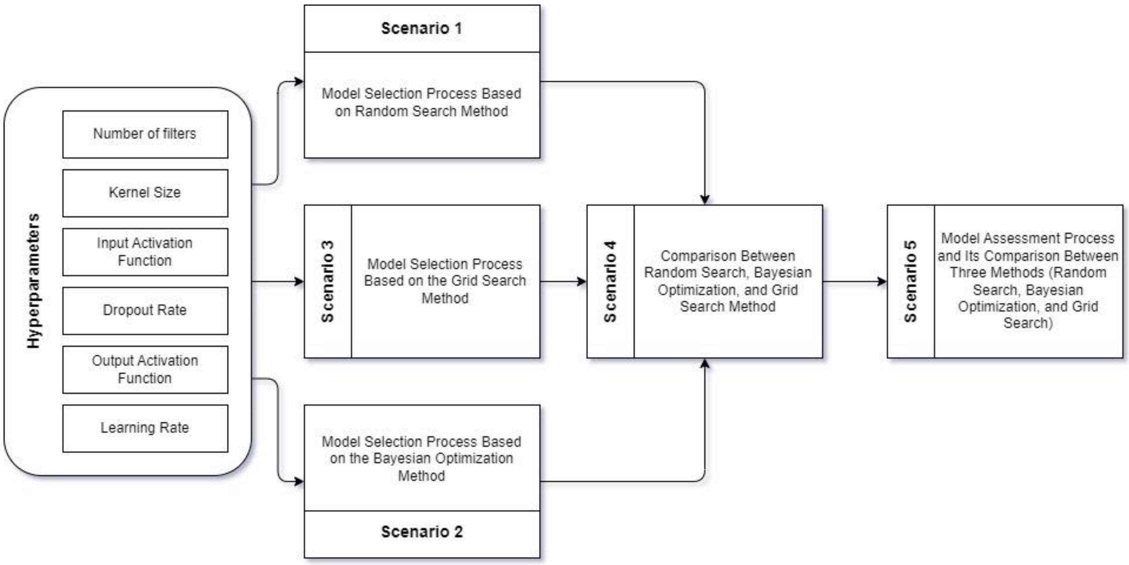


**Fig. 3.** Illustration of scenario experiments

As depicted in Figure 3, this study runs various experimental scenarios by applying three hyperparameter optimization methods for sentiment analysis based on CNN, including Grid Search, Random Search, and Bayesian Optimization. The Random Search and Bayesian Optimization methods were implemented by applying four trial values, namely 5, 10, 20, and 30. Meanwhile, the application of Grid Search consisted of 540 trials according to the CNN hyperparameter combination applied in this study. Details regarding the hyperparameters tested in this study can be seen in Table 2. The total combination of the six hyperparameters is 540

**Table 2**
Word2Vec parameters

| No | Parameter | Value |
|---|---|---|
| 1 | Number of filters | 100, 200, 300 |
| 2 | Kernel Szie | 3, 4, 5 |
| 3 | Input activation function | ReLU, Tanh |
| 4 | Dropout rate | 0.1, 0.2, 0.3, 0.4, 0.5 |
| 5 | Output activation function | Sigmoid, SoftMax |
| 6 | Learning rate | 1e-2, 1e-3, 1e-4 |

Based on the three factors, including method, type of hyperparameter and number of trials, several scenarios will be tested in this study. Experimental scenario 1 aims to determine the effect of the number of trials from implementing the Random Search method on the performance and computational time required in the model selection process. As with experimental scenario 1, experimental scenario two also aims to determine the performance and computational time required in the model selection process when the Bayesian Optimization method is applied. The decision-making process for the two scenarios is based on two evaluation metrics, namely validation accuracy and validation loss. Equal to the previous two scenarios, the third scenario aims to evaluate the performance and computational time required in the model selection process when applying the Grid Search method. In addition, those three scenarios also discussed the required execution time.

Based on the best results obtained from the three scenarios, a discussion will be performed regarding comparing the implementation of the three methods, namely Random Search, Bayesian Optimization and Grid Search in scenario 4. Furthermore, experiment 5 was performed for the model assessment process. This testing process was applied based on the best hyperparameter combinations obtained from experimental scenarios one, two, and three, described previously.

### *3.3 Result and Analysis*

As previously explained, scenario one performed a model selection process by applying the Random Search method to select the hyperparameters that perform best in forming a CNN-based sentiment classification model. There are four different numbers of trials, including 5, 10, 20, 30. The results can be seen in Table 3.

**Table 3**
Model selection results on Random Search Method for Scenario 1

| Number of Trials | Validation | | Time Elapsed (s) |
|---|---|---|---|
| | Loss | Accuracy | |
| 5 | 4.11317 | 0.876 | 121.66637 |
| 10 | 0.436079 | 0.876 | 325.178366 |
| 20 | 2.166620 | 0.880 | 328.235021 |
| 30 | <u>0.512365</u> | <u>0.884</u> | <u>999.369904</u> |

Table 3 shows that the best performance is obtained with an accuracy value of 0.884 while a loss value of 0.512365. Furthermore, because the best performance can only be obtained during the fourth trial with 30 trials, the time required reaches 999.369904 seconds.

Unlike the implementation of Random Search, the implementation of Bayesian Optimization can achieve the best performance with only five trials and produces the same accuracy performance as the implementation of the Random Search method of 0.884 and a smaller loss value of 0.435563. It can be seen in Table 4.

**Table 4**
Model selection results on Bayesian Optimization Method for Scenario 2

| Number of Trials | Validation | | Time Elapsed (s) |
|---|---|---|---|
| | Loss | Accuracy | |
| 5 | <u>0.435563</u> | <u>0.884</u> | <u>175.746765</u> |
| 10 | 2.615675 | 0.876 | 322.884202 |
| 20 | 2.310540 | 0.876 | 346.687378 |
| 30 | 1.428435 | 0.880 | 944.714824 |

Furthermore, implementing the Grid Search method got the best performance as the Random Search method in the 30th trial. The accuracy value that can be obtained in this trial is 0.892. Details of performance accuracy and time execution can be seen in Table 5.

**Table 5**
Model selection results on Grid Search Method for Scenario 3

| Number of Trials | Validation | | Time Elapsed (s) |
|---|---|---|---|
| | Loss | Accuracy | |
| 5 | 0.311137 | 0.872 | 201.224944 |
| 10 | 0.498905 | 0.876 | 374.511038 |
| 20 | 1.277708 | 0.872 | 708.818886 |
| 30 | <u>2.149796</u> | <u>0.892</u> | <u>1,163.573392</u> |

A comparison of the computational time required by the three hyperparameter optimization methods to get the best performance can be seen in Figure 4. Based on the figure, it can be seen that the implementation of the Bayesian Optimization method has the lowest computation time, which is only 175.746765 seconds or less than 3 minutes in the fifth trial. This lowest value was obtained since Bayesian Optimization stores information about the evaluation results of hyperparameter combinations that have been conducted previously to obtain a probability model that is capable of mapping hyperparameters to probability scores in the objective function as follows:

$$p(score|hyperparameters\_combination) \quad (6)$$

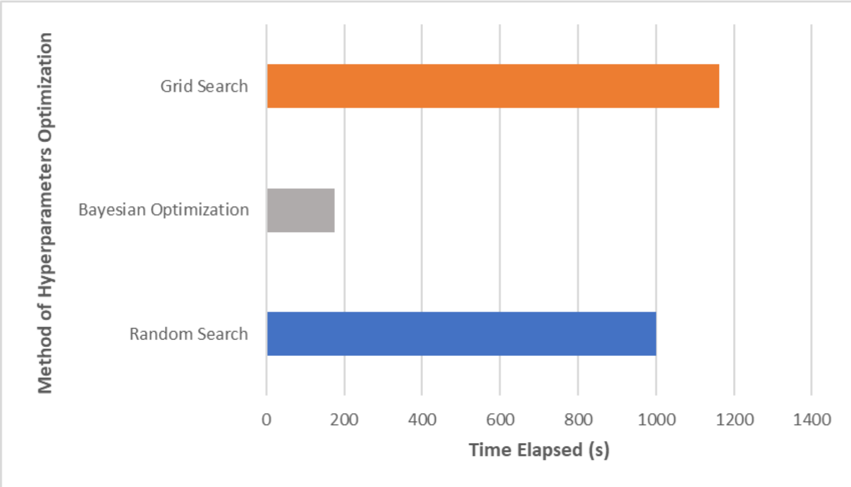


**Fig. 4.** Comparison of time computation for three methods for Scenario 4

This model is called a "surrogate" for the objective function. In this study, the objective function is to maximize the accuracy value as previously described. Optimizing the surrogate is easier than optimizing the objective function. On the contrary, the other two methods, namely Grid Search and Random Search, do not pay attention to information about the results obtained in previous trials. Both methods will continue running all hyperparameter combinations in the previously identified search space, even though the hyperparameter combinations tested are unlikely to optimize the objective function. The difference in decreasing execution time of the two methods (Random Search and Grid Search) to Bayesian Optimization can be seen in Figure 5. The difference between Grid Search and Bayesian Optimization is 987.83 seconds, while between Random Search and Bayesian Optimization is 823.62 seconds.

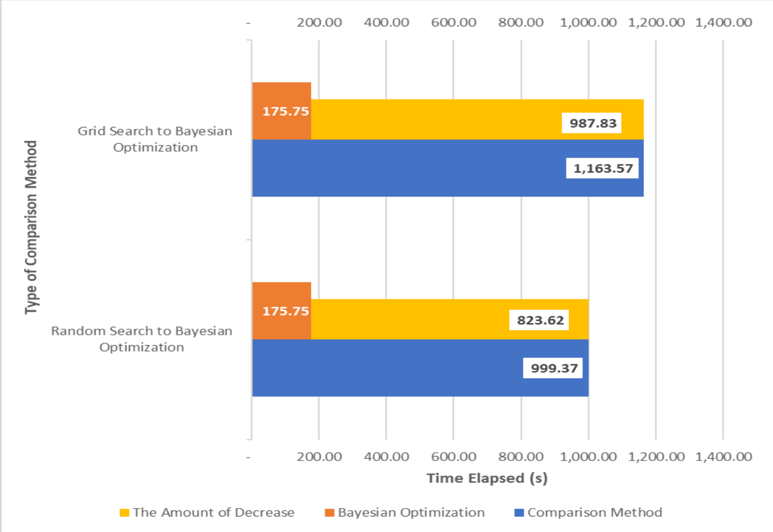


**Fig. 5.** Time comparison between three methods

Even though the Grid Search and Random Search methods have a longer execution time than the Bayesian Optimization method, these two methods are still better than manual tuning since manual tuning requires 540 trials according to the combination of all the employed hyperparameters in this study, as described in Table 5. The Random Search method can achieve the same accuracy as Bayesian Optimization after reaching 30 trials with 999.37 seconds. In comparison, the Grid Search method on trial 30, with a total time of 1,163.57 seconds, achieved better accuracy than the Random Search and Bayesian Optimization methods, namely 0.892 or a difference of 0.8%. The best hyperparameter combination for each method can be seen in Table 6.

**Table 6**
The Best Hyperparameters for each method

| No | Parameter | Grid Search | Random Search | Bayesian Optimization |
|---|---|---|---|---|
| 1 | Number of filters | 200 | 300 | 100 |
| 2 | Kernel Size | 5 | 5 | 3 |
| 3 | Input activation function | TanH | ReLU | ReLU |
| 4 | Dropout rate | 0.4 | 0.5 | 0.1 |
| 5 | Output activation function | Sigmoid | Softmax | Sigmoid |
| 6 | Learning rate | 0.01 | 0.01 | 0.01 |

The hyperparameter learning rate for the three methods is the same, namely 0.01, because a large learning rate will increase the loss error, and a small learning rate will overcome this problem but will increase the convergence time [20]. The dropout values in applying Bayesian Optimization and Random Search are large, 0.4 and 0.5, respectively. These values correlate with the number of filters (200 and 300, respectively) and the filter size (5 for both methods) in the two methods, so much noise representation results in a higher dropout value that must be applied. On the contrary, the number of filters in implementing Grid Search is 100, so the dropout value that should be applied is only 0.1. The input activation function in implementing the Random and Grid Search methods is ReLU because this activation function does not have the vanishing gradient problem [21], so that the weight updating process can be performed optimally. In implementing Bayesian Optimization, TanH is the best input activation function because TanH provides a stronger non-linearity than ReLU and has good stability [22]. The best output activation functions are Sigmoid and SoftMax because the classification task in this study is a binary classification.

Furthermore, as previously explained in this fifth scenario, a comparison of the best hyperparameter implementation of the three methods, including Bayesian Optimization, Random Search, and Grid Search, for the model assessment process is discussed and described in Figure 6.

Bayesian Optimization gives the highest values on Precision and Specificity. Since this study assigned the sentiment negative as the positive class, the high precision means the classifier performs well in predicting the negative sentiment as a negative class with the lowest error on identifying positive sentiment recognized as sentiment negative. Thus, implementing Bayesian Optimization is the right choice for sentiment analysis tasks since the precision of predicting negative sentiment classes is vital in hospitality business intelligence, especially the use of related information for product improvement or hotel service improvement. In addition, the high value on specificity means that the obtained classifier performs well in predicting the positive sentiment as a positive class, with a low error in predicting positive sentiment as a negative class. This condition also reinforces the previous statement that implementing Bayesian Optimization as a Hyperparameter Optimization method is appropriate for analysing hospitality sentiment. The precision value of the application of Bayesian Optimization reaches 0.8168, while the specificity value reaches 0.7176.

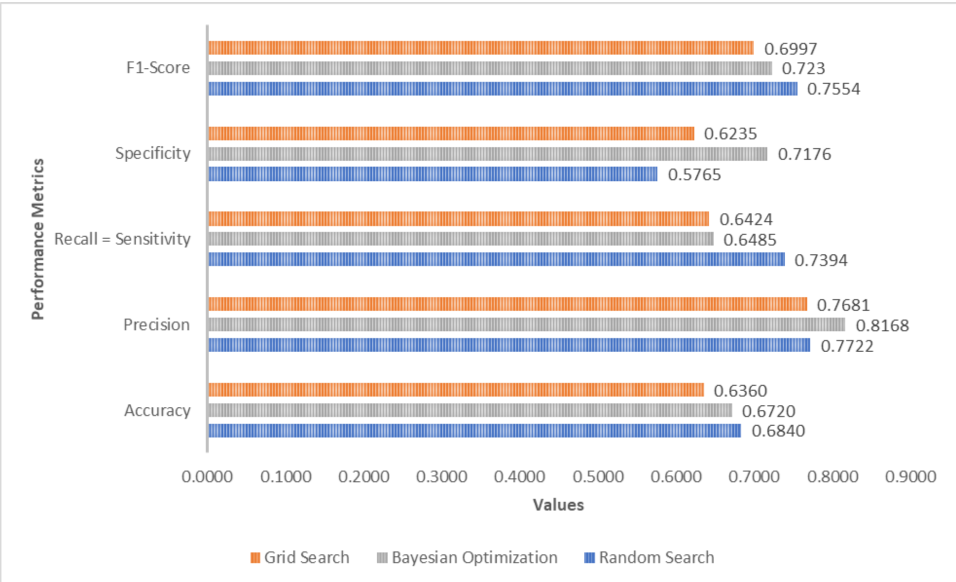


**Fig. 6.** Comparison of three methods in model assessment process for Scenario 5

Subsequently, the F1-Score metric and the accuracy of the classifier model obtained using Bayesian Optimization are lower than Random Search, with a difference of 3.24% and 1.2%, respectively. While the difference in the value of recall or sensitivity is quite significant, reaching 9.09%.

## 4. Conclusions

The experimental results show that the Bayesian Optimization method can reduce the computation time to 175.746765 seconds in the fifth trial. It has the lowest computation time compared to Random and Grid Searches. In addition, it made a reasonably large cut compared to manual trial-based hyperparameter tuning that needs 540 trials. In addition, implementing Bayesian Optimization results in the highest values on Precision and Specificity for the model assessment process, which are 0.8168 and 0.7176, respectively. Therefore, the implementation of Bayesian Optimization is appropriate for sentiment analysis tasks since the precision of predicting negative sentiment classes is vital in hospitality business intelligence, especially the use of related information for product improvement or hotel service improvement.

**Acknowledgement**
This research was funded by a grant from Directorate of Research, Technology and Community Services, Ministry of Education, Culture, Research and Technology, Indonesia (PFR Grant 449A-26/UN7.D2/PP/VI/2023, 20 June 2023).

**References**
[1] Raguseo, Elisabetta, Paolo Neirotti, and Emilio Paolucci. "How small hotels can drive value their way in infomediation. The case of 'Italian hotels vs. OTAs and TripAdvisor'." *Information & Management* 54, no. 6 (2017): 745-756. https://doi.org/10.1016/j.im.2016.12.002
[2] Muhammad, Putra Fissabil, Retno Kusumaningrum, and Adi Wibowo. "Sentiment analysis using Word2vec and long short-term memory (LSTM) for Indonesian hotel reviews." *Procedia Computer Science* 179 (2021): 728-735. https://doi.org/10.1016/j.procs.2021.01.061
[3] Liu, Xuerui, Fuad Mehraliyev, Chun Liu, and Markus Schuckert. "The roles of social media in tourists' choices of travel components." *Tourist studies* 20, no. 1 (2020): 27-48. https://doi.org/10.1177/1468797619873107
[4] Alqasemi, Fahd, Hatem Abdelkader, and Amira Abdelwahab. "Opinion Lexicon Automatic Construction on Arabic language." *Journal of Advanced Research in Applied Mechanics* 40, no. 1 (2017): 1-6.
[5] Kusumaningrum, Retno, Iffa Zainan Nisa, Rizka Putri Nawangsari, and Adi Wibowo. "Sentiment analysis of Indonesian hotel reviews: from classical machine learning to deep learning." (2021). https://doi.org/10.26555/ijain.v7i3.737
[6] Chakraborty, Partha, Farah Nawar, and Humayra Afrin Chowdhury. "Sentiment analysis of Bengali facebook data using classical and deep learning approaches." In *Innovation in Electrical Power Engineering, Communication, and Computing Technology: Proceedings of Second IEPCCT 2021*, pp. 209-218. Springer Singapore, 2022. https://doi.org/10.1007/978-981-16-7076-3_19
[7] Vateekul, Peerapon, and Thanabhat Koomsubha. "A study of sentiment analysis using deep learning techniques on Thai Twitter data." In *2016 13th International joint conference on computer science and software engineering (JCSSE)*, pp. 1-6. IEEE, 2016. https://doi.org/10.1109/JCSSE.2016.7748849
[8] Ghulam, Hussain, Feng Zeng, Wenjia Li, and Yutong Xiao. "Deep learning-based sentiment analysis for roman urdu text." *Procedia computer science* 147 (2019): 131-135. https://doi.org/10.1016/j.procs.2019.01.202
[9] Kanakaraddi, Suvarna G., Ashok K. Chikaraddi, Karuna C. Gull, and P. S. Hiremath. "Comparison study of sentiment analysis of tweets using various machine learning algorithms." In *2020 International conference on inventive computation technologies (ICICT)*, pp. 287-292. IEEE, 2020. https://doi.org/10.1109/ICICT48043.2020.9112546
[10] Nawangsari, Rizka Putri, Retno Kusumaningrum, and Adi Wibowo. "Word2vec for Indonesian sentiment analysis towards hotel reviews: An evaluation study." *Procedia Computer Science* 157 (2019): 360-366. https://doi.org/10.1016/j.procs.2019.08.178
[11] Kim, Hannah, and Young-Seob Jeong. "Sentiment classification using convolutional neural networks." *Applied Sciences* 9, no. 11 (2019): 2347. https://doi.org/10.3390/app9112347
[12] Yang, Li, and Abdallah Shami. "On hyperparameter optimization of machine learning algorithms: Theory and practice." *Neurocomputing* 415 (2020): 295-316. https://doi.org/10.1016/j.neucom.2020.07.061
[13] Wu, Jia, Xiu-Yun Chen, Hao Zhang, Li-Dong Xiong, Hang Lei, and Si-Hao Deng. "Hyperparameter optimization for machine learning models based on Bayesian optimization." *Journal of Electronic Science and Technology* 17, no. 1 (2019): 26-40.
[14] Riza, M. Alfa, and Novrido Charibaldi. "Emotion detection in Twitter social media using long short-term memory (LSTM) and fast text." *Int. J. Artif. Intell. Robot* 3, no. 1 (2021): 15-26. https://doi.org/10.25139/ijair.v3i1.3827
[15] Lopez-del Rio, Angela, Maria Martin, Alexandre Perera-Lluna, and Rabie Saidi. "Effect of sequence padding on the performance of deep learning models in archaeal protein functional prediction." *Scientific reports* 10, no. 1 (2020): 14634. https://doi.org/10.1038/s41598-020-71450-8
[16] Elgeldawi, Enas, Awny Sayed, Ahmed R. Galal, and Alaa M. Zaki. "Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis." In *Informatics*, vol. 8, no. 4, p. 79. MDPI, 2021. https://doi.org/10.3390/informatics8040079
[17] Liashchynskyi, Petro, and Pavlo Liashchynskyi. "Grid search, random search, genetic algorithm: a big comparison for NAS." *arXiv preprint arXiv:1912.06059* (2019).

[18] Sukri, Nur Raihana, Nurulakmar Abu Husain, Syarifah Zyurina Nordin, and Mohd Shahrir Mohd Sani. "Structural Damage Identification Using Model Updating Approach: A Review." *Journal of Advanced Research in Applied Mechanics* 84, no. 1 (2021): 1-12.
[19] Alzubaidi, Laith, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions." *Journal of big Data* 8 (2021): 1-74. https://doi.org/10.1186/s40537-021-00444-8
[20] Huang, Haibo, Xiaorong Huang, Weiping Ding, Siwen Zhang, and Jian Pang. "Optimization of electric vehicle sound package based on LSTM with an adaptive learning rate forest and multiple-level multiple-object method." *Mechanical Systems and Signal Processing* 187 (2023): 109932. https://doi.org/10.1016/j.ymssp.2022.109932
[21] Singh, Bharat, Suchit Patel, Ankit Vijayvargiya, and Rajesh Kumar. "Analyzing the impact of activation functions on the performance of the data-driven gait model." *Results in Engineering* 18 (2023): 101029. https://doi.org/10.1016/j.rineng.2023.101029
[22] Wang, Changyang, Kegen Yu, Fangyu Qu, Jinwei Bu, Shuai Han, and Kefei Zhang. "Spaceborne GNSS-R wind speed retrieval using machine learning methods." *Remote Sensing* 14, no. 14 (2022): 3507. https://doi.org/10.3390/rs14143507