JURNAL NASIONAL TEKNIK ELEKTRO DAN TEKNOLOGI INFORMASI
p-ISSN 2301–4156 | e-ISSN 2460–5719

Study and Analysis of Hyperparameter Tuning of IndoBERT in Fake News Detection

Anugerah Simanjuntak[1], Rosni Lumbantoruan[2], Kartika Sianipar[3], Rut Gultom[4], Mario Simaremare[5], Samuel Situmeang[6], Erwin Panggabean[7]

[1,2,3,4,5,6] Fakultas Informatika dan Teknik Elektro, Program Studi Sarjana Sistem Informasi, Institut Teknologi Del, Jl. Sisingamangaraja, Sitoluama, Kecamatan Laguboti, Toba Samosir, Sumatera Utara – Indonesia (*email : [1]iss19039@students.del.ac.id, [2]rosni@del.ac.id, [3]iss19014@students.del.ac.id, [4]iss19059@students.del.ac.id, [5]mario@del.ac.id, [6]samuel.situmeang@del.ac.id*)

[7] Sekolah Tinggi Manajemen Informatika dan Komputer, Program Studi Sarjana Teknik Informatika, Pelita Nusantara, Jl. Iskandar Muda No.1, Medan Baru, Sumatera Utara – Indonesia (email: *[7]erwinpanggabean9@gmail.com*)

[Received: dd Month 20xx, Revised: 21 October 2023]
Corresponding Author: Rosni Lumbantoruan

ABSTRACT — The rapid advancement of communication technology has transformed how information is shared, but it has also brought concerns about the proliferation of false information. A recent report by the Ministry of Communication and Informatics in Indonesia revealed that around 800,000 websites are involved in spreading false information, underscoring the seriousness of the problem. To combat this issue, researchers have focused on developing techniques to detect and combat fake news. This research centers on using IndoBERT-base-p1 for fake news detection and aims to enhance its performance through three methods to tune the hyperparameter value of the model namely: Bayesian optimization, grid search, and random search. After comparing the outcomes of the three hyperparameter tuning methods, Bayesian Optimization emerged as the most effective approach. Achieving a precision of 88.79%, recall of 94.5%, and F1-score of 91.56% for the "fake" label, Bayesian Optimization outperformed the other hyperparameter tuning methods as well as the model using the fine tuning hyperparameter value. These findings emphasize the importance of hyperparameter tuning in improving the accuracy of fake news detection models. Utilizing Bayesian Optimization and optimizing the specified hyperparameters, the model demonstrated superior performance in accurately identifying instances of fake news, providing a valuable tool in the ongoing battle against disinformation in the digital realm.

KEYWORDS — Fake News, BERT, IndoBERT, Hyperparameter Tuning, Natural Language Processing

I. INTRODUCTION

The internet has become an integral part of the lives of most of the world's population. Indonesia, in particular, is the third country in Asia with the highest number of internet users, totaling 212.35 million individuals [1] has transformed the Internet as an extremely useful source of information. Information technology in Indonesia has also progressed significantly, with the current number of internet users reaching 132.7 million or 51.6% of the country's population [2]. The accelaration of the spread of information as the result of communication technology advancement can also lead to misinformation or commonly referred to as "fake news", which is defined as deliberately fabricated news intended to deceive or mislead [3]. Similar to offensive language [4], common motives for spreading fake news include misleading readers, damaging reputations, or generating sensationalism. As other countries, Indonesia is also struggling with this mislead information, as indicated by a survey revealing that 38% to 61.1% of rural communities believe in fake news, while 45.3% to 79.6% of urban communities also fall victim to fake news [5]. Additionally, the Indonesian Ministry of Communication and Information Technology (Kominfo) states on its official website that there are 800,000 hoax-spreading websites circulating in Indonesia in 2017 [6]. These statistics clearly demonstrate the concerning prevalence of fake news dissemination in Indonesia.

This issue can be tackled by manually verifying the accuracy of news through alternative sources or through automated classification. However, these methods necessitate a significant amount of effort and time. Meanwhile, the Transformer model, a deep learning language model based on self-attention, has become extremely popular for automatically detecting fake news [7]. In recent years, transformers and their various modifications have achieved substantial performance improvements in various natural language processing tasks [7]. One prominent example of a pre-trained language representation model that has yielded exceptional performance across multiple specialized architecture tasks is BERT (Bidirectional Encoder Representations from Transformers) [8]. BERT is made to train deep text representations in both the left and right directions. [9].

Fake news may exist on both legal and illegal news sites. Even sites that predominantly share legitimate news are not immune to spreading fake news. Given this circumstance, in this situation, we used a collection of information where there are more real news articles than fake ones. This helps the model accurately find fake news in a big group of real news. In this study, researchers will take a dataset from previously conducted research with the name turnbackhoax.id dataset [10]. Where this researcher compares BERT, CNN, BiLSTM and Hybrid CNN-BiLSTM methods in detecting fake news in Indonesian. The findings indicated that BERT outperformed other methods. IndoBERT, a variant specifically developed for the Indonesian language, has been widely employed for various NLP tasks, including text classification, language modeling, and other natural language processing tasks. However, current IndoBERT research has mostly ignored the performance benefits of tweaking the model's hyperparameters. As a result, in our study, despite BERT's strong performance, we primarily concentrate on tuning the hyperparameter value for this model, particularly for tuning the hyperparameters of the model. Thus, in this research, we mainly focus on this tuning especially for the identification of fake news.

Despite BERT's strong performance in various natural language processing tasks, including fake news detection, ongoing research and model development consistently seek to improve performance. Hyperparameter tuning is an effective method to achieve such enhancements. Prior research has demonstrated that hyperparameter tuning can significantly enhance machine learning model performance, including those employed in fake news detection [11]–[13]. By enhancing fake news detection performance through hyperparameter tuning, it becomes possible to develop a more reliable and effective system for identifying the spread of fake news. As highlighted [14], selecting the optimal hyperparameters presents a challenge that must be addressed to attain better prediction results.

This research's contribution is comparing three of the most common approaches for tweaking hyperparameter values namely: bayesian optimization, random and grid search. Each technique is analyzed according to its performance and its fulfillment of the criteria for precision, recall, and F1-score. The method with the highest performance for all the evaluation metrics will be regarded as the tuning method for the case study of identifying fake news in Indonesian language.

II. STATE OF THE ART OF TEXT CLASSIFICATION

This section focuses on the latest advancements in classification methods, encompassing BERT as a general approach and a specific model designed for the Indonesian language, known as IndoBERT. Additionally, we explore popular techniques for hyperparameter tuning in the context of classification tasks namely Random Search, Grid Search and Bayesian Optimization.

A. BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS (BERT)

BERT (Bidirectional Encoder Representations from Transformers) is a language representation model designed as a bidirectional transformer that captures information from text by combining representations from both left and right context tokens across all layers. BERT understands word relationships in a bidirectional manner and generates representation vectors for each word based on their relationships within a sentence [9].

BERT utilizes a self-attention mechanism, where it combines multiple word vectors as input and includes cross-attention in both directions between two sentences [9]. BERT is used as a sentence encoder that accurately represents the context within a sentence. BERT excels in transfer learning, as the output of the BERT model provides a pre-trained model that can be adopted for text classification tasks, as shown in Figure 1 [9]. The authors use BERT as a sentence encoder, which accurately obtains contextual representations of a sentence [15]. There are two types of BERT models used for specific contexts [16]:

1) BERT BASE

BERT Base consists of an embedding layer and 12 encoder layers with 12 attention heads. It has 110 million parameters and hidden sizes of 768. This model is smaller in size and computationally affordable. However, it may not be appropriate for complex text mining operations.

2) BERT LARGE

BERT Large is a larger model with more computational demands. It has 24 layers, 16 attention heads, 340 million parameters, and hidden sizes of 1024. This model can handle larger text data and provide better results.

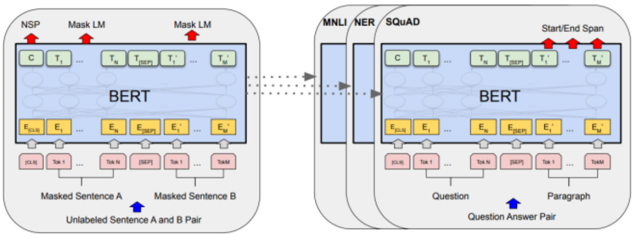


Figure 1. Pre-training and Fine-tunning BERT

Since BERT is a pretrained model that requires input data to be in a particular format, thus the following are required [9]:

1. Special token [SEP] is used to denote the end of a sentence or to separate two sentences.
2. Special token [CLS], is used for classification and is padded at the beginning of the text.
3. Tokens corresponding to the fixed vocabulary used in BERT.
4. Token IDs for tokens originating from the BERT tokenizer.
5. Mask IDs are made up of binary values; 1 means the model should pay attention to the tokens, while 0 indicates that it should not pay attention to the padding elements.
6. Segment IDs are used to differentiate between various sentences.
7. Positional embeddings are used to indicate the order of tokens in the sequence.

B. INDONESIAN BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS (INDOBERT)

IndoBERT is a transformer-based model inspired by BERT and trained on a large corpus of the Indonesian language. It incorporates a substantial vocabulary of over 220 million words sourced from reliable Indonesian language references such as online newspapers, the Indonesian Web Corpus, and other reputable sources. IndoBERT is a pretrained model developed with 2.4 million steps or 180 epochs, resulting in strong performance on various NLP tasks. The pretrained model is trained using masked language modeling (MLM) and next sentence prediction (NSP). It utilizes a Transformer architecture with 12 layers and 768 processing units. IndoBERT is trained on a WordPiece vocabulary of Indonesian consisting of 31,923 tokens. Several variants of IndoBERT have been developed, including IndoBERT-base, IndoBERT-large, and IndoBERT-lite. The base form contains 12 layers and roughly 125 million parameters, whereas the large variant has 24 layers and around 340 million parameters[17].

C. INDOBERT-BASE

IndoBERT-base is the fundamental model of IndoBERT, which was trained a 5.5 billion-word corpus that encompasses several forms of Indonesian text. This model can be used to perform a variety of natural language processing tasks. It has 12 transformer layers with 12 heads per layer and 110 million parameters. The following are examples of IndoBERT-based systems [17][18]:

1) INDOBERT-BASE-P1

Using transfer learning techniques, this model is trained on a huge dataset of texts in the Indonesian language from diverse sources, including news articles, Wikipedia, and social media. This model can also be used to do a variety of natural language processing tasks.

2) INDOBERT-BASE-P2

Compared to IndoBERT-base-p1, this model has been trained on a more complex dataset and can produce more accurate outputs. As a result, this model is appropriate for tasks requiring a deeper understanding of language, such as document classification and sentiment analysis

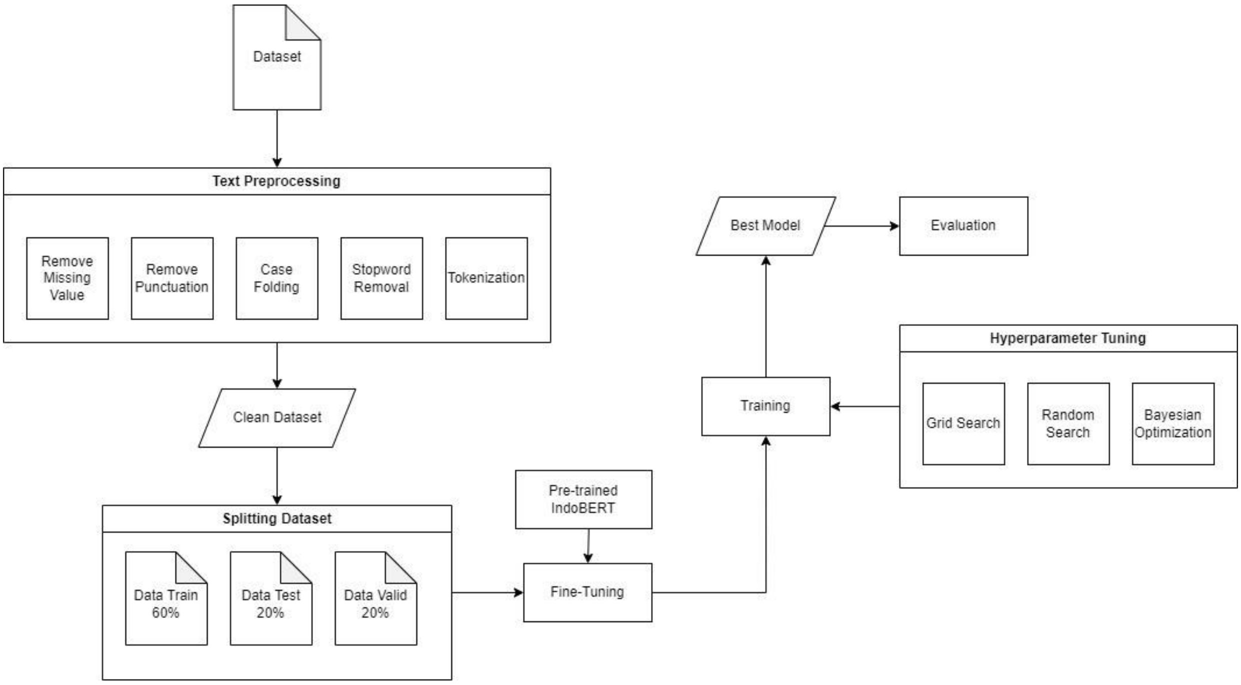


Figure 2. Research Methodology

D. HYPERPARAMETER TUNING

Hyperparameter tuning commonly involves techniques such as Grid Search, Random Search, and Bayesian Optimization [19]:

1) GRID SEARCH

Grid Search is a traditional method for optimizing hyperparameters by exhaustively searching a specified subset of the hyperparameter space of the training algorithm. The parameter space of the machine learning algorithm can include real or unbounded values for certain parameters, so it's necessary to define boundaries for applying grid search. Hyperparameters are determined using minimum (lower bound) and maximum (upper bound) values. Grid Search is an exponential time approach that is difficult to implement in high-dimensional areas. However, because hyperparameters normally operate independently of one another, it may be managed by parallelizing. Grid Search is an exponential time approach that is difficult to implement in high-dimensional areas. However, because hyperparameters normally operate independently of one another, it may be managed by parallelizing [19].

2) RANDOM SEARCH

Random Search is a method for identifying optimal sets of hyperparameter configurations by randomly attempting different hyperparameter combinations. It works by defining the hyperparameter search space and randomly selecting from it to generate a collection of hyperparameter combinations. There is no guarantee that this approach will combine the ideal hyperparameter values to attempt due to its randomness. Random Search is efficient and can handle large-dimensional data well. Instead of examining 100,000 samples, only 1,000 random samples from the hyperparameter set will be checked.

The advantages of Random Search include requiring less time and computation to obtain results compared to other optimization methods like Grid Search. Although it may not find the best set of hyperparameters, it can provide a model that approximates the ideal model's performance. On the other hand, Random Search has disadvantages such as requiring more time due to the random search in each iteration, which can be time-consuming and require many iterations to find the best hyperparameter combination.

3) BAYESIAN OPTIMIZATION

Bayesian Optimization is an informed search strategy, which means that previous iterations' findings were used as learning input for the next, and the outcome of one iteration influenced the next. A probabilistic surrogate model that captures our belief about the unknown objective function's behavior and an acquisition function that defines how optimal the query sequence is are the essential components of Bayesian Optimization. In practice, the acquisition function is frequently in the form of regret, either simple or cumulative [19].

Optimization in deep learning refers to finding efficient parameter values that maximize the output given numeric inputs. Selecting appropriate hyperparameters plays a crucial role in training BERT and significantly impacts the performance of the built model. Hyperparameters defined for the built model consist of Learning Rate, Batch size, and Number of Epochs. These hyperparameters play a significant role in optimizing the BERT model's performance during the training process.

E. MODEL EVALUATION

To measure the performance evaluation of the model using hyperparameter tuning in detecting fake news, we use accuracy, precision, recall, and F1 measurement metrics. The accuracy is calculated using Eq.1.

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{1}$$

The precision is calculated by calculating the ratio of positive correct predictions including the true prediction of positive class (TP) and true prediction of negative class (TN) compared to the overall positive checked results, including the previous correct predictions and false prediction of positive (FP) and negative class (FN) . The precision is calculated using Eq.2.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

Recall performs the calculation to get all documents that are considered relevant by the system. The recall is calculated using Eq.3.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

F1-Score is evaluated with a range of 0 being the worst value and 1 being the best value. The F1-Score is calculated using Eq.4.

$$\text{F1 Score} = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recall} \tag{4}$$

III. METHODOLOGY

In this section, we provide a detailed explanation of the data that will be used in this research, pre-processing, and the Bert model for classification, as illustrated in Figure 2 within the Research Methodology. The process starts with dataset collection, followed by data pre-processing and dataset splitting. Subsequently, the pre-trained IndoBERT stage is introduced, and the section concludes with the hyperparameter tuning stage. Each of these stages will be elaborated upon in the following subsections.

A. DATASET

In this study, we used a dataset by previous research, Turnbackhoax. The dataset contains data consisting of 1,116 lines in Indonesian. The number of fake news data is 433 data, and the number of real data is 683 data. There are three attributes in the dataset, namely "label" which consists of two labels, namely zero (0) for the real news category and label one (1) for the fake news category, "headline" contains the title of the news and "body" contains the text content of the news. After that, we evaluate the model's performance by first splitting the dataset into training, validation, and testing data, each of which contributes 60% to training, 20% to test, and 20% to validation.

B. DATA PREPROCESSING

The text preprocessing stage is the process of turning unstructured data formats into structured data in order to present the data in a clear word format. The data preprocessing stage will eliminate missing values, which means that any empty or null data in the dataset will be deleted at this point. Then we delete the punctuation, which attempts to remove properties such as "@% -"" " in order to not interfere with the calculating process in the algorithm's application. Then it will proceed with lowercasing all words to ensure that words like "yANG," "yang," and "Yang" all have the same meaning. Following that, the stop-word removal stage is the process of deleting terms from the list of unimportant words. A corpus from Sastrawi is used in this stop-word removal procedure to generate a list of stop-words in the Indonesian language. Stopwords in Sastrawi include "yang," "di," "ke," "adalah," "dan," and "dari," among others. The dataset had a total of 1,676,646 words before the stopword removal. The data was reduced to 1,672,798 words after the stopwords were removed, for a decrease percentage of 0.23%. This is done to reduce extraneous words from the statement while keeping their meaning intact. Tokenization is the final stage, in which the text is divided into individual words. This stage's goal is to find the tokens for each word in a sentence.

C. FINE TUNING

The fine-tuning process begins by initializing the IndoBERT pretrained model, in this research we use IndoBERT-base-p1. The model selection is based on the limited size of the dataset used in this research. IndoBERT base has fewer layers and parameters that make it faster to train the model [17]. In the fine-tuning process, the model architecture will be modified according to the specific task to be solved where in this research, the specific task to be solved is text classification. The fine-tuning process will be given a sequence of texts from the dataset as input to the IndoBERT model. The maximum sequence length value to be used is 512 [9]. If a text sequence exceeds the maximum length of 512 tokens, then the text needs to be truncated or divided into smaller parts to fit the constraint. In fine tuning, we set the hyperparameter value from previous research for these three hyperparameters: learning rate, batch size, and epoch. After going through a sufficient fine-tuning process, the IndoBERT model will have knowledge tailored to the specific task given. The text representation generated by the fine-tuned model can be used to perform inference on new data.

D. HYPERPARAMETER TUNING

Hyperparameter tuning in BERT is performed to find the optimal combination of hyperparameters to improve model performance. The hyperparameter tuning methods that will be implemented are the simplest and most common of these methods namely Grid Search, Random Search, and Bayesian Optimization [18]. The three methods were chosen because Grid Search is easy to implement and suitable for all types of models. Random Search is more efficient for large search spaces and many hyperparameters. While Bayesian Optimization provides global optimization for black-box functions, reducing the validation error evaluation required.

The framework used to automatically determine hyperparameter values is Optuna [20]. Optuna is built dynamically so that it is more likely to get the best parameters that may not be obtained by other hyperparameter methods [21].

The hyperparameters that will be used in this research are learning rate, batch size and epoch. Epoch, learning rate and batch size are the two most important hyperparameters in BERT [17]. Learning rate serves to control how fast or slow the model learns during the training process. If the learning rate is too high, the model may fail to converge, while if it is too low, the model may take too long to converge. Batch size serves to determine the number of samples used in each training iteration. A larger batch size can result in faster training time, but it can also lead to poor generalization ability and accuracy. In addition, epoch is used to determine how many times the model will be trained across the dataset.

E. EVALUATION

In this research, evaluation will be used as a foundation or benchmark to compare the performance of the three models after applying the hyperparameter tuning. The evaluation calculations and metrics used are accuracy, precision, recall, and F1-score.

The accuracy value is required in this study to evaluate the system's correctness in accurately classifying the data. However, accuracy alone will not demonstrate the model's capabilities. If all classified sentences are in the negative category, a classifier that always predicts negative sentences will have a very high accuracy. As a result, additional measurements like precision, recall, and F1-score are required. The precision value indicates whether the built model is correct or incorrect, whereas the recall value is required to evaluate the model's sensitivity. A high sensitivity model implies that the model's predictions are highly relevant, and vice versa. The F1-score is used to determine the comparative value of the weighted average of precision and recall, representing the overall performance of the system.

IV. EXPERIMENTS

We compare the performance of the three hyperparameter tuning methods in determining the optimal model hyperparameter value for learning rate, batch size, and number of epochs in this section. This experiment will utilize a pre-trained IndoBERT-base-p1, obtained from the Indo Benchmark repository on Hugging Face [19], with Adam as the optimizer.

A. FINE TUNING

The process will then proceed to fine-tuning, where the initialized IndoBERT model will be retrained with specific data. The hyperparameter values for learning rate, batch size and epoch that are used in the fine-tuning process are depicted in Table 1.

The hyperparameters will be trained together with the BERT model, and the validation accuracy will be calculated. The model will be evaluated, and from the evaluation, the combination of hyperparameters with the best performance will be determined.

TABLE 1
HYPERPARAMETER VALUE OF FINE TUNING

Hyperparameter	Value Range
Learning Rate	2e-5
Batch Size	16
Epochs	10

The result will include the best hyperparameters or the best model, which consists of the learning rate, number of epochs, batch size, and best validation loss.

B. HYPERPARAMETER TUNING

After fine-tuning, the hyperparameter tuning phase will follow, starting with selecting a range of values for each hyperparameter based on previous research. Three hyperparameter tuning techniques will be compared: Grid Search, Random Search, and Bayesian Optimization. The experiment will be conducted three times to test each method. The hyperparameter ranges used in the three types of hyperparameter tuning can be seen in Table 2.

TABLE 2
VALUE RANGE FOR HYPERPARAMETER TUNING

Hyperparameter	Value Range
Learning Rate	2e-5, 3e-5, 5e-5
Batch Size	16, 32
Epochs	10

In this research, we use Optuna as the framework for hyperparameter tuning. Optuna will try 36 different combinations of hyperparameters. If we look at all the possible hyperparameters given, we can see that the total combination is 3 (learning rate) * 6 (epochs) * 2 (batch size) = 36. In using Optuna with the 'minimize' direction, the goal is to find the hyperparameter that gives the lowest value to the objective function. Although local optima refer to the lowest point in a particular hyperparameter space and global optima is the lowest value in the entire space, Optuna uses various techniques to avoid getting stuck at local optima and approach global optima. So, the main goal in 'minimize' with Optuna is to approach or reach the global optima. To prevent overfitting, early stopping will be added. This technique involves monitoring the validation loss during training and will stop the training process when the validation loss does not improve. From the training results, the best model of the best hyperparameter value combination will be obtained from the specified range of values.

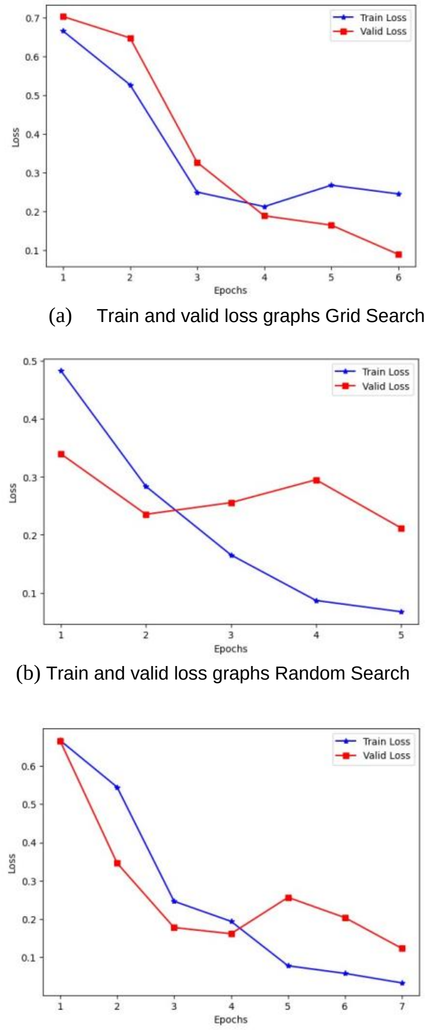
V. RESULT AND DISCUSSION

In this section, we compare the result of the three methods of hyperparameter tuning and then finally compare the performance to the one before they are tuned in terms of Precision, Recall, F1-Score and Accuracy.

A. TRAINING COMPARISON OF HYPERPAMATER TUNING METHODS

Figures 3 (a), (b), and (c) show the model performance during training in terms of training and validation loss, with the x-axis representing the number of epochs of the training model and the y-axis representing the associated loss rate at each epoch. The train loss graph initially has a high loss value, but as the number of epochs increases, both training and validation loss begin to decrease. We also used early stopping, where the training is terminated when overfitting is detected.

Figure 3(a) for Grid Search shows that training stops at epoch 6. The model still experiences a slight overfitting, where if observed the valid loss graph initially has a high loss value which then decreases as the number of epochs increases, but after a few epochs, the loss value starts to rise again.



(a) Train and valid loss graphs Grid Search

(b) Train and valid loss graphs Random Search

(c) Train and valid loss graphs Bayesian Optimization

Figure 3 A.Training comparison for different approaches

This indicates that the model is getting a better "fit" to the training data as the number of epochs increases but is not successfully generalizing what it has learned to the validation data.

Figure 3(b) for Random Search shows that training stops at epoch 6. The model still experiences a slight overfitting, where if the valid loss graph is observed, it initially has a high loss value and decreases in the first few epochs, but after a few epochs, the loss value in the validation data starts to increase again, while the loss value in the training data continues to decrease. This shows that the model is overfitting so that when applied to validation data, its performance decreases. The performance of this Random Search in terms of its loss is the worst performance amongst the three techniques, which shows a model that remains overfitting because the model cannot properly generalize patterns from training data to new data.

Figure 3(c) for Bayesian Optimization shows that the training early stops at epoch 7. This model does not exhibit overfitting, as evidenced by the valid loss graph, which initially has a high loss value but decreases as the number of epochs increases. This means that the model's performance on training and validation data is correlated, indicating that the model can learn patterns in training data and generalize well on validation data. As we can see, Bayesian optimization performs better and more consistently than the other two methods.

B. PERFORMANCE COMPARISON OF HYPERPAMATER TUNING METHODS

Previously, we have trained the model and found the optimal hyperparameter value for the three techniques used namely Grid Search, Random Search and Bayesian Optimization. The optimal hyperparameter values returned by each method are depicted in Table 3 below.

TABLE 3
OPTIMAL HYPERPARAMETER VALUE

Method	Learning Rate	Epoch	Batch size
Grid Search	5e-5	8	32
Random Search	5e-5	9	32
Bayesian Optimization	2e-5	8	16

Given the selected hyperparameter value in Table 3, then we compare the performance of these three hyperparameter tuning techniques in terms of Precision, Recall, and F1-Score respectively on Table 4, 5 and 6.

Table 4 depicts the performance comparison of these three techniques in terms of Precision. As can be seen, in terms of Precision, Bayesian Optimization outperforms the other two, Grid Search and Random Search in identifying the real and fake label of the news.

TABLE 4
PEROFRMANCE COMPARISONS OF DIFFERENT APRROACHES
IN TERMS OF PRECISION

Method	Precision	
	Label “Real”	Label “Fake”
Grid Search	0.9462	0.8661
Random Search	0.9541	0.8462
Bayesian Optimization	**0.9726**	**0.8879**

Table 5 depicts the performance comparison of the methods in terms of Recall. Bayesian Optimization outperforms Random Search and Grid Search in classifying the fake news data.

TABLE 5
PEROFRMANCE COMPARISONS OF DIFFERENT APRROACHES
IN TERMS OF RECALL

Method	Recall	
	Label “Real”	Label “Fake”
Grid Search	0.9336	0.8899
Random Search	0.9204	0.9083
Bayesian Optimization	**0.9425**	**0.9450**

Meanwhile, in Table 6 we can see the performance comparison of these three methods in terms of F1-Score. It shows that Bayesian Optimization constantly outperforms the other two methods overall evaluation metrics in classifying fake news data and followed by Grid Search and lastly the Random Search. Although Random Search performs well on the training data, this model is likely to have low evaluation scores on unobserved data.

TABLE 6
PEROFRMANCE COMPARISONS OF DIFFERENT APRROACHES
IN TERMS OF F1-SCORE

Method	F1-Score	
	Label “Real”	Label “Fake”
Grid Search	0.9399	0.8899
Random Search	0.9369	0.8761
Bayesian Optimization	**0.9573**	**0.9156**

C. PERFORMANCE COMPARISON OF MODEL BEFORE AND AFTER HYPERPARAMETER TUNING

In this section, we evaluate the performance of IndoBERT-base-p1 before and after applying the hyperparameter tuning in classifying the fake news, particularly in correctly classifying the news with label “fake”. We compare the performance of IndoBERT-base-p1 after the tuning with Grid Search, Random Search, Bayesian Optimization using the optimal hyperparameter value in Table 3 and the model by using the fine tuning hyperparameter value as depicted in Table 2.

Table 7 and 8 respectively illustrate the evaluation of the model before and after hyperparameter tuning in terms of Precision, Recall, F1-Score and Accuracy. The best performance is represented by the bold value, while the second-best performance is indicated by the underlined value.

TABLE 7
PERFORMANCE COMPARISON OF FINE TUNING VS. HYPERPARAMETER TUNING IN TERMS OF PRECISION, RECALL, AND F1-SCORE

Methods	Precision	Recall	F1-Score
Model with **Fine Tuning**	0.8632	0.9266	0.8938
Model with **Grid Search**	0.8661	0.8899	0.8899
Model with **Random Search**	0.8462	0.9083	0.8761
Model with **Bayesian Optimization**	**0.8879**	**0.9450**	**0.9156**

TABLE 8
FINE PERFORMANCE COMPARISON FINE TUNING VS. IN TERMS OF ACCURACY

Methods	Accuracy
Model with **Fine Tuning**	0.9313
Model with **Grid Search**	0.9194
Model with **Random Search**	0.9164
Model with **Bayesian Optimization**	**0.9432**

We can see that Model after the hyperparameter tuning, specifically Bayesian Optimization technique overpass the model with fine tuning for all evaluation metrics assessed. Bayesian Optimization is considered better than fine-tuning alone for IndoBERT-base-p1 because it effectively searches for the optimal hyperparameters for the model. Fine-tuning involves adjusting the weights of pre-trained models on specific tasks, but it often requires careful tuning of hyperparameters to achieve the best performance on the target task, such as fake news detection in this case.

Thus, it is proved that hyperparameter tuning in this research can increase the performance of the model in classifying the fake news. In addition to that, hyperparameter optimization can involve non-linear interactions between parameters, making it challenging for traditional methods like Grid Search or Random Search to effectively explore space. Bayesian Optimization, on the other hand, utilizes probabilistic modeling and Bayesian inference to better capture these non-linear relationships, making it more suitable for complex hyperparameter tuning tasks. The Grid Search method comes as the second best for the Precision and has a slightly different performance in terms of Recall and F1-Score compared to the model with fine tuning, yet, in terms of accuracy, fine tuning is the second best.

VI. CONCLUSION

Based on the results and discussion, the following conclusions can be drawn:

The Bayesian Optimization hyperparameter tuning method demonstrates superior performance on the IndoBERT-base-p1 model using the Turnbackhoax dataset compared to other methods, including the model without hyperparameter tuning. Specifically, it achieves the highest F1-Score in correctly identifying news data labeled as "fake," with optimal parameter values of 16 for batch size, 2e-5 for learning rate, and 8 for the number of epochs.

Based on the experimental comparisons between the fine-tuning method on the IndoBERT model and three other hyperparameter tuning methods, it becomes evident that Bayesian Optimization excels in optimizing the evaluation value during fine-tuning. This conclusion is supported by the fact that the evaluation value obtained through Bayesian Optimization hyperparameter tuning is higher when compared to the evaluation value achieved by IndoBERT-base-p1 fine-tuning on the Turnbackhoax dataset. On the contrary, the evaluation results from Grid Search and Random Search hyperparameter tuning do not lead to an improvement in the evaluation results during fine-tuning. This is due to the fact that the evaluation value generated after hyperparameter tuning using Grid Search and Random Search has decreased when compared to the evaluation value achieved during the initial fine-tuning process.

ACKNOWLEDGMENT

This research is fully funded by Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) of Institut Teknologi Del. Their commitment as the research division to promoting and facilitating research activities has been instrumental in the successful completion of this project.

REFERENCES

[1] V. B. Kusnandar, “Pengguna internet indonesia peringkat ke-3 terbanyak di asia,” 2021. [Online]. Available: https://databoks.katadata.co.id/datapublish/2021/10/14/pengguna-internet-indonesia-peringkat-ke-3-terbanyak-di-asia

[2] M. Rahmat, Indrabayu, and I. S. Areni, “Hoax Web Detection For News in Bahasa Using Support Vector Machine,” *2019 Int. Conf. Inf. Commun. Technol. ICOIACT*, 2019, doi: 10.1109/ICOIACT46704.2019.8938425.

[3] A. Thota, P. Tilak, S. Ahluwalia, and N. Lohia, “Fake News Detection: A Deep Learning Approach,” 2018.

[4] R. Lumbantoruan, R. U. Siregar, I. Manik, N. Tambunan, and H. Simanjuntak, “Analysis Comparison of FastText and Word2vec for Detecting Offensive Language,” *2022 IEEE Int. Conf. Comput. Sci. Inf. Technol. ICOSNIKOM*, 2022, doi: 10.1109/ICOSNIKOM56551.2022.10034886.

[5] I. Nadzir, S. Seftiani, and Y. S. Permana, “Hoax and misinformation in Indonesia: insights from a nationwide survey,” 2019.

[6] A. Yuliani, “Ada 800.000 situs penyebar hoax di indonesia,” 2017. [Online]. Available: https://www.kominfo.go.id/content/detail/12008/ada-800000-situs-penyebar-hoax-di-indonesia/0/sorotan_media

[7] Sultana, Raquiba, and T. Nishino, “Fake News Detection System: An implementation of BERT and Boosting Algorithm.,” 2021, doi: https://doi.org/10.1007/s11042-020-10183-2.

[8] Suadaa, L. Hulliyyatus, I. Santoso, B. Panjaitan, and A. Tabitha, “Transfer Learning of Pre-trained Transformers for Covid-19 Hoax Detection in Indonesian Language,” *Indones. J. Comput. Cybern. Syst. IJCCS*, 2021.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2018, doi: 10.18653/v1/n19-1423.

[10] J. Fawaid, A. Awalina, R. Y. Krisnabayu, and N. Yudistira, “Indonesia’s Fake News Detection using Transformer Network.,” *6th Int. Conf. Sustain. Inf. Eng. Technol.*, 2021.

[11] G. Maike and M. Aßenmacher, “Evaluating unsupervised representation learning for detecting stances of fake news,” *Proc. 28th Int. Conf. Comput. Linguist.*, 2020.

[12] R. R. Rajalaxmi, L. V. N. Prasad, B. Janakiramaiah, C. S. Pavankumar, N. Neelima, and V. E. Sathishkumar, “Optimizing hyperparameters and performance analysis of LSTM model in detecting fake news on social media,” *Assoc. Comput. Mach.*, 2022, doi: 10.1145/3511897.

[13] N. Kanagavalli, S. Priya, Baghavathi, and D. Jeyakumar, “Design of Hyperparameter Tuned Deep Learning based Automated Fake News Detection in Social Networking Data.,” *2022 6th Int. Conf. Comput. Methodol. Commun. ICCMC*, 2022, doi: 10.1109/ICCMC53470.2022.9753739.

[14] C. W. Kencana, E. B. Setiawan, and I. Kurniawan, “Hoax Detection System on Twitter using Feed-Forward and Back-Propagation Neural Networks Classification Method.,” 2020, doi: 10.29207/resti.v4i4.2038.

[15] M. E. Peters *et al.*, “Deep contextualized word representations,” 2018.

[16] R. K. Kaliyar, A. Goswami, and P. Narang, “FakeBERT: Fake news detection in social media with a BERT-based deep learning approach.,” 2021, doi: 10.1007/s11042-020-10183-2.

[17] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, “IndoLEM and IndoBERT: A benchmark dataset and pre-trained language model for Indonesian NLP.,” pp. 757--770, 2020, doi: 10.18653/v1/2020.coling-main.66.

[18] S. M. Isa, G. Nico, and M. Permana, “Indobert for Indonesian fake news detection,” *ICIC Express Lett.*, vol. 16, 2021.

[19] B. Bischl *et al.*, “Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges,” 2021.

[20] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A Next-Generation Hyperparameter Optimization Framework,” *Assoc. Comput. Mach.*, 2019, doi: 10.1145/3292500.3330701.

[21] W. S. Bhaya, “Review of data preprocessing techniques in data mining.,” 2017, doi: 10.3923/jeasci.2017.4102.4107.