DOI: 10.22146/jnteti.v13i1.8532

Research and Analysis of IndoBERT Hyperparameter Tuning in Fake News Detection

Anugerah Simanjuntak[1], Rosni Lumbantoruan[1], Kartika Sianipar[1], Rut Gultom[1], Mario Simaremare[1], Samuel Situmeang[1], Erwin Panggabean[2]

[1] Information System Study Program, Faculty of Informatics and Electrical Engineering, Institut Teknologi Del, Toba, Indonesia
[2] Program Studi Sarjana Teknik Informatika, Sekolah Tinggi Manajemen Informatika dan Komputer Pelita Nusantara, Medan, Indonesia

[Submitted: 10 Agustus 2023, Revised: 7 November 2023, Accepted: 12 December 2023]
Corresponding Author: Rosni Lumbantoruan (email: rosni@del.ac.id)

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 were 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, was the third country in Asia with the highest number of internet users, totaling 212.35 million individuals, in 2021 [1]. They regard the Internet as a very useful source of information. Another source stated that information technology in Indonesia also progressed significantly, with the number of Internet users in 2019 reaching 132.7 million or 51.6% of the country's population [2]. However, the rapid dissemination of information due to the advancement in communication technology can also lead to misinformation, commonly referred to as "fake news," which is defined as news that is intentionally fabricated to deceive or mislead the readers [3]. Similar to offensive language [4], common motives for spreading fake news include misleading readers, damaging reputations, or creating sensation. Indonesia, like other countries, is also struggling against this misleading 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]. In addition, the Ministry of Communication and Informatics of the Republic of Indonesia (Kominfo RI) stated on its official website that there were 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 gained significant popularity for its automatic detection of fake news [7]. In recent years, transformers and their various modifications have achieved substantial performance improvements in various natural language processing tasks (NLP) [7]. One prominent example of a pretrained language representation model that has yielded exceptional performance across multiple specialized architecture tasks is bidirectional encoder representations from transformers (BERT) [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 real news are are not exempt from spreading fake news. Given this circumstance, this research employed a collection of information that had more real news articles than fake news. It helps the model accurately find fake news in a big group of real news. This research utilized a dataset from previously conducted research, namely the TurnBackHoax.ID dataset [10]. This research compared BERT, convolutional neural network (CNN), bidirectional long short-term memory (BiLSTM), and combined CNN-BiLSTM methods to detect fake news in Indonesian. The findings indicated that BERT outperformed other methods. IndoBERT, a variant of BERT specifically developed for the Indonesian language, has been widely employed for various NLP tasks, including text classification, and language modeling. However, most of the current IndoBERT research overlooks the performance benefits of tuning the model's hyperparameters. As a result, despite the excellent performance of BERT, this research mainly concentrates on tuning the hyperparameter values for the model, especially to identify fake news.

Regardless of BERT's exceptional performance in various NLP tasks, including fake news detection, ongoing research and model development consistently seek to improve its performance. Hyperparameter tuning is an effective method to achieve such improve. Prior research has demonstrated that hyperparameter tuning can significantly enhance machine learning model performance, including those employed in fake news detection [11]–[13]. Improving the performance of fake news detection through hyperparameter tuning allows for the development of more reliable and effective systems to identify the dissemination of fake news. The selection of optimal hyperparameters poses a challenge and must be addressed to attain better prediction results [14].

This research makes contribution by comparing the three most commonly used approaches for tuning hyperparameter values: Bayesian optimization, random search, and grid search. Each technique was analyzed based on its performance and fulfillment of precision, recall, and F1-score criteria. The method with the highest performance for all evaluation metrics is regarded as the tuning method for the case research of Indonesian fake news identification.

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, known as IndoBERT. Additionally, it explores popular techniques for hyperparameter tuning in the context of classification tasks namely Bayesian optimization, random search, and grid search.

A. BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS (BERT)

BERT 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. It 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]. As a sentence encoder, BERT accurately represents the context of a sentence. BERT excels in transfer learning because it provides a pretrained model that can be adopted for text classification tasks, as depicted in Figure 1 [9]. Researchers use BERT as a sentence encoder, which can accurately derive the contextual representation 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 suitable for complex text mining operations.

2) BERT LARGE

BERT large is a larger model with more computational requirements. 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.

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

- Special token [SEP] is used to denote the end of a sentence or to separate two sentences.
- Special token [CLS] is used for classification and is padded at the beginning of the text.

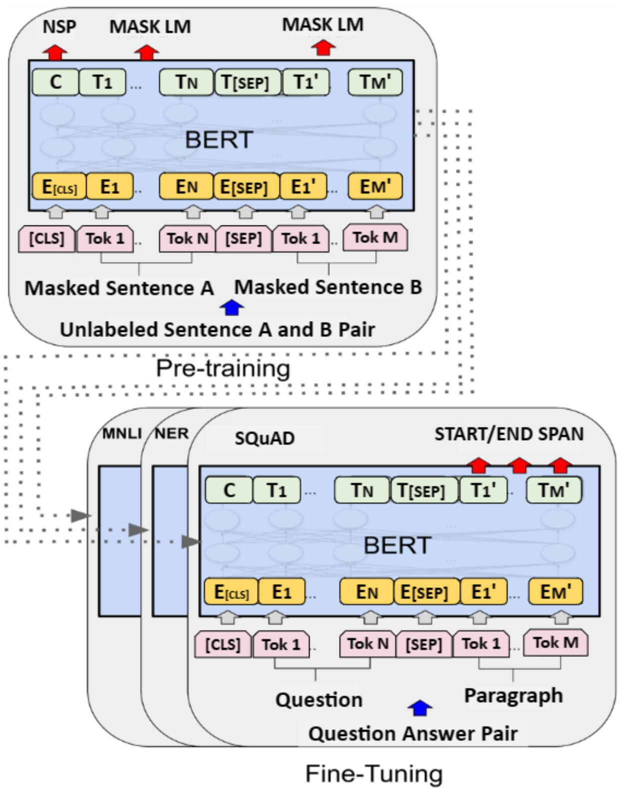

Figure 1. Pretraining and fine-tunning of BERT.

- Tokens corresponding to the fixed vocabulary used in BERT.
- Token IDs for tokens originating from the BERT tokenizer.
- 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.
- Segment IDs are used to differentiate between various sentences.
- Positional embeddings are used to indicate the order of tokens in the sequence.

B. INDOBERT

IndoBERT is a transformer-based model inspired by BERT and was 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 and the Indonesian Web Corpus. IndoBERT is a pretrained model developed with 2.4 million steps or 180 epochs, resulting in robust performance on various NLP tasks. The pretrained model was trained using masked language modeling (MLM) and next sentence prediction (NSP). It utilized a transformer architecture with 12 layers and 768 processing units. IndoBERT was trained on an Indonesian WordPiece vocabulary 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. trained with a corpus of 5.5 billion words encompassing various forms of Indonesian texts. This model is applicable for a variety of NLP tasks. It comprises 12 transformer layers with 12 heads per layer and 110 million parameters. The following are examples of IndoBERT-base [17], [18]:

1) INDOBERT-BASE-P1

The model was trained using transfer learning techniques on an extensive dataset of Indonesian texts from various sources, such as news articles, Wikipedia, and social media. In addition, this model is capable of performing various NLP tasks.

2) INDOBERT-BASE-P2

This model, in contrast to IndoBERT-base-p1, was trained on more complex datasets and can generate more accurate output. As a result, it is well-suited for tasks that necessitate deeper language understanding, such as document classification and sentiment analysis.

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 conventional 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. Therefore, it is necessary to determine the boundaries in order to implement the 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, hyperparameters usually operate independently of each other, allowing management parallelization [19].

2) RANDOM SEARCH

Random search is a method for identifying optimal sets of hyperparameter configurations by randomly attempting different hyperparameter combinations. This method operates by defining a hyperparameter search space and performing random selection to generate a set of hyperparameter combinations. Due to its randomness, there is no guarantee that this approach will combine the ideal hyperparameter values to implement. Random search is efficient and can handle large-dimensional data well. Instead of 100,000 samples, only 1,000 random samples from the hyperparameter set were evaluated.

Random search has the advantage of generating results with less time and computation, unlike other optimization methods such as grid search. Although it may not find the best set of hyperparameters, it can provide a model that approximates the ideal model's performance. Random search, on the other hand, has disadvantages such as requiring more time due to random search in each iteration, which can be time-consuming and require numerous iterations to find the best combination of hyperparameters.

3) BAYESIAN OPTIMIZATION

Bayesian optimization is an information-based search algorithm, meaning that the outcomes of the previous iteration are used as learning input for the next iteration, and the outcomes of one iteration have an impact on the subsequent iteration. In Bayesian optimization, a probabilistic surrogate model that captures opinions about the behavior of an unknown objective function and an acquisition function that defines the optimal level of the query sequence are essential components. 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 from given numeric inputs. Appropriate selection of 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

Recalling the nature of the model as a classification model that identifies whether news is fake, this research measured the model's performance using the widely used metrics for classification problems such as accuracy, precision, and recall. In addition to that, an F1-score that balances the needs for both precision and recall was used. Accuracy is calculated using (1).

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)}. \qquad (1)$$

Accuracy 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). Accuracy is calculated using (2).

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

Recall performs the calculation to get all documents that are considered relevant by the system. Recall is calculated using (3).

$$Recall = \frac{TP}{TP+FN}. \qquad (3)$$

F1-score is rated on a scale of 0 to 1, with 0 being the worst and 1 being the greatest. F1-Score is formulated in (4).

$$F1Score = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recall}. \qquad (4)$$

III. METHODOLOGY

This section presents a detailed explanation of data used in this research, preprocessing, and the BERT model for classification, as illustrated in Figure 2. The process commenced with the dataset collection, followed by data preprocessing and dataset splitting. Following that was the IndoBERT pretrained stage and concluded with the hyperparameter tuning stage.

A. DATASET

This research utilized a dataset from prior research, namely TurnBackHoax.ID. The dataset contained data consisting of 1,116 lines in Indonesian. The total "fake" news data were 433, and the total "real" news data were 683. There were three attributes in the dataset, namely "label," which comprised of two labels: zero (0) for the "real" news category and one (1) for the "fake" news; "headline" contained the news title; and "body" contained the news content text. When constructing the model, it is essential to divide the dataset in order to evaluate the machine learning model's performance. The model's performance was then evaluated by first dividing the dataset

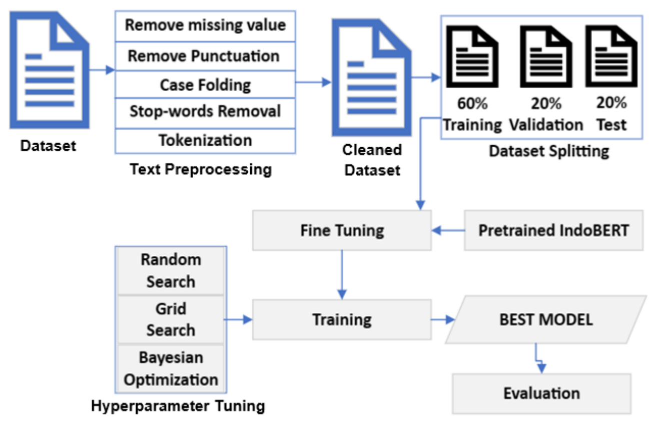

Figure 2. Research methodology.

into training, validation, and testing data, each of which contributed 60% to training, 20% to test, and 20% to validation.

B. DATA PREPROCESSING

The text preprocessing stage is the process of transforming unstructured data formats into structured data to present the data in a clear word format. The data preprocessing involves removing missing values, which means that any empty or null data in the data set will be removed at this stage. After that, punctuation marks are removed to eliminate properties such as "@% -" " so as not to interfere with the calculation process in the algorithm application. Then, it will proceed with converting all capital words to lowercase to ensure that words like "yANG," "yang," and "Yang" all have the same meaning.

The stopword removal stage is the process of removing terms from the list of insignificant words. The Sastrawi corpus was utilized in this stopword removal procedure to generate a list of stopwords in Indonesian. In Sastrawi, words such as "yang," "di," "ke," "adalah," "dan," and "dari" are instances of stopwords. This dataset had a total of 1,676,646 words before the stopword removal. The data decreased to 1,672,798 words after the stopwords were removed, resulting in a reduction of 0.23%. It was done to reduce extraneous words from the statements while maintaining their meaning intact. Tokenization is the final stage in data preprocessing, where texts are divided into individual words. This stage aims to find tokens for each word in a sentence.

C. FINE-TUNING

The fine-tuning process began by initializing the pretrained IndoBERT model. The IndoBERT model variation used in this research was IndoBERT-base-p1. The model selection was based on the limited size of the dataset employed in this research. IndoBERT-base has fewer layers and parameters, allowing it to train the model faster [17].

In the fine-tuning process, the model architecture is modified according to the specific task to be accomplished. In this research, this specific task was text classification. In the fine-tuning process, text sequences from the dataset were used as input to the IndoBERT model. The maximum sequence length value used was 512 [9]. If a text sequence exceeds the maximum length of 512 tokens, it needs to be truncated or divided into smaller parts to fit the length limit.

In this fine-tuning process, the hyperparameter value of learning rate, batch size, and epoch were set based on the value of previous research [18]. After the IndoBERT model underwent an adequate fine-tuning process, it had knowledge tuned 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

In prior research, the resulting model did not consider the need to perform hyperparameter tuning; thereby, the parameter values used were static [18]. In this research, the model in the previous research [18] was compared with the model after hyperparameter tuning. Hyperparameter tuning in BERT was performed to find the optimal combination of hyperparameters to improve model performance. The simplest and most commonly used methods, namely grid search, random search, and Bayesian optimization, were selected to compare models before and after hyperparameter tuning [18]. Grid Search was chosen as it is suitable for all types of models and is fast enough to obtain decent result. Random Search, on the other hand, is faster than grid search, which makes it more efficient for large search spaces and many hyperparameters. Meanwhile, Bayesian optimization provides global optimization for black-box functions, reducing the validation error evaluation required. The framework used to automatically determine hyperparameter values was Optuna [20]. Optuna is built dynamically, so it is more likely to acquire the best parameters that may not be obtained using other hyperparameter methods [21].

The hyperparameters used in this research encompassed the learning rate, batch size, and number of epochs. These three parameters are the most essential hyperparameters in BERT [17]. Learning rate serves to control the speed at which the model learns during the training process. If the learning rate is too high, the model may fail to converge. Conversely, 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, the number of epochs is used to determine the amount of training performed on the model across the entire dataset.

E. EVALUATION

This research used evaluation as a foundation or benchmark to compare the performance of the three models after applying the hyperparameter tuning. The calculations and evaluation metrics used were accuracy, precision, recall, and F1-score. In this research, the accuracy value was used to evaluate whether the system was able to classify data accurately. However, accuracy alone is not enough to show the model's capability. If all classified sentences are in the negative category, a classifier that always predicts sentences as negative will yield an extremely high accuracy. As a result, additional metrics such as precision, recall, and F1-score are required. The precision value indicates whether the model built had a high or low level of correctness, while the recall value is required to evaluate the model's sensitivity. A model with high sensitivity indicates that the model's predictions are highly relevant, and vice versa. F1-score is used to determine the comparative value of the weighted average of precision and recall, representing the system's overall performance.

IV. EXPERIMENTS

This section presents the performance comparison of the three hyperparameter tuning methods in determining the

TABLE I
HYPERPARAMETER VALUE OF FINE TUNING

Hyperparameter	Value Range
Learning Rate	2×10^{-5}
Batch Size	16
Epochs	10

TABLE II
VALUE RANGE FOR HYPERPARAMETER TUNING

Hyperparameter	Value Range
Learning Rate	2×10^{-5}, 3×10^{-5}, 5×10^{-5}
Batch Size	16, 32
Epochs	10

optimal model hyperparameter value for learning rate, batch size, and number of. This experiment utilized a pretrained IndoBERT-base-p1, acquired from the Indo Benchmark repository on Hugging Face [19], with Adam as the optimizer. The hardware specification used for developing and conducting experimentation was processor with Intel(R) Xeon(R) CPU @ 2.20 GHz and 8GB of RAM.

A. FINE-TUNING

The subsequent process is fine-tuning, where the initialized IndoBERT in [18] model was retrained with specific data. In this research, the model [18] was denoted as the model with fine-tuning. The hyperparameter values for learning rate, batch size, and number of epochs used in the fine-tuning process are presented in Table I.

The hyperparameters were trained together with the BERT model, and the validation accuracy was calculated. Afterwards, the model was evaluated These evaluation results determined the hyperparameter combination that had the best performance.

The results included the best hyperparameters or the best model, consisting of the best learning rate, batch size, number of epochs, and validation loss.

B. HYPERPARAMETER TUNING

After the fine-tuning stage was completed, hyperparameter tuning was performed. This stage was begun by selecting a range of values for each hyperparameter based on previous research. Three hyperparameter tuning techniques were compared: grid search, random search, and Bayesian optimization. The experiment was conducted three times to test each method. The hyperparameter ranges used in the three types of hyperparameter tuning can be seen in Table II.

This research employed the Optuna as the framework for hyperparameter tuning, which tried 36 different hyperparameter combinations. Given all the possible hyperparameters, the total combination $3 \times 6 \times 2 = 36$ (learning rate × number of epochs × batch size). Optuna was used with the "minimize" direction to find the hyperparameter that yielded 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 employs various techniques to avoid getting stuck at local optima and approach global optima. Early stopping was added to prevent overfitting. This technique involves monitoring the validation loss during training and stops the training process when the validation loss does not improve. From the training results, the best model of the best hyperparameter value combination was obtained from the specified range of values.

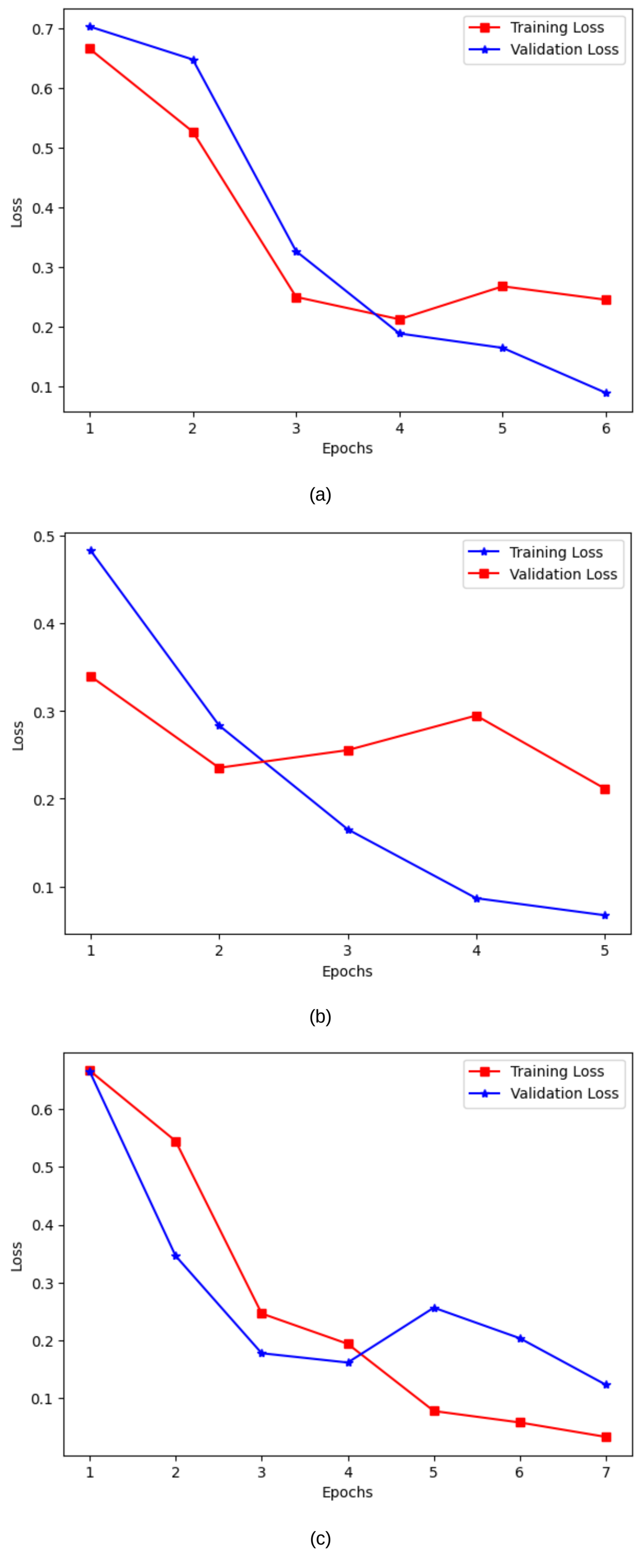

Figure 3. Comparison of training loss and validation loss for multiple approaches, (a) grid search, (b) random search, (c) Bayesian optimization.

V. RESULTS AND DISCUSSION

This section presents the comparison of the three hyperparameter tuning methods. After that, their performances are compared with their performance before tuning. These performances include precision, recall, F1-score, and accuracy.

A. TRAINING COMPARISON OF HYPERPAMATER TUNING METHODS

Figure 3 show the model performance during training in terms of training and validation loss, with the x-axis

TABLE III
OPTIMAL HYPERPARAMETER VALUE

Method	Learning Rate	Epoch	Batch size
Grid search	5×10^{-5}	8	32
Random search	5×10^{-5}	9	32
Bayesian optimization	2×10^{-5}	8	16

representing the number of epochs of the training model and the y-axis representing the associated loss rate at each epoch. The training loss graph initially has a high loss value. Nevertheless, training and validation loss exhibits a decrease with the increase of epoch number. An early stopping was also employed to terminate the training in case of overfitting.

Figure 3(a) shows that training at the 6th epoch for the grid search approach. The model still experienced a little overfitting, i.e., the validation loss graph initially had a high loss value, which then decreased as the number of epochs increased. However, after a few epochs, the loss value rose again. It 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) demonstrates that for the random search approach, training stopped at the 6th epoch. If observed, the validation loss graph initially had a high loss value and decreased in the first few epochs. After a few epochs, the loss value on the validation data began to increase again, while the loss value on the training data continued to decrease. It shows that the model is overfitting, so its performance decreases when applied to validation data. In terms of loss, the random search had the worst performance among the three techniques, suggesting that the model remains overfitting because the model cannot generalize the patterns from the training data to the 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 validation 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. It can be seen that Bayesian optimization performs better and more consistently than the other two methods.

B. PERFORMANCE COMPARISON OF HYPERPAMATER TUNING METHODS

Previously, the model was trained and the optimal hyperparameter values were identified 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 III.

Given the chosen hyperparameter values in Table III, the performance of these three hyperparameter tuning techniques was compared in terms of precision, recall, and F1-score, as shown in Table IV until Table VI. Table IV depicts the performance comparison of these three techniques in terms of precision. As can be seen, in terms of Precision, Bayesian optimization outperformed the other two techniques in identifying real and fake labels on a news story.

Table V depicts the performance comparison of the methods in terms of recall. Bayesian optimization outperformed random search and grid searches in classifying fake news data.

TABLE IV
PERFORMANCE COMPARISONS OF DIFFERENT APPROACHES IN TERMS OF PRECISION

Method	Precision			
	Label "Real"	Improvement	Label "Fake"	Improvement
Grid Search	0.9462	2.50%	0.8661	1.89%
Random Search	0.9541	1.77%	0.8462	3.53%
Bayesian Optimization	0.9726	-	0.8879	-

TABLE V
PERFORMANCE COMPARISONS OF DIFFERENT APPROACHES IN TERMS OF RECALL

Method	Recall			
	Label "Real"	Improvement	Label "Fake"	Improvement
Grid Search	0.9336	0.83%	0.8899	4.90%
Random Search	0.9204	2.03%	0.9083	3.33%
Bayesian Optimization	0.9425	-	0.9450	-

TABLE VI
PERFORMANCE COMPARISONS OF DIFFERENT APPROACHES IN TERMS OF F1-SCORE

Method	F1-Score			
	Label "Real"	Improvement	Label "Fake"	Improvement
Grid Search	0.9399	1.64%	0.8899	2.29%
Random Search	0.9369	1.91%	0.8761	3.46%
Bayesian Optimization	0.9573	-	0.9156	-

Meanwhile, Table VI presents the performance comparison of these three methods in terms of F1-score. It shows that that Bayesian optimization constantly outperformed the other two methods in the overall evaluation metrics in classifying fake news data, followed by grid and random search. Despite performing well on training data, random search was likely to yield low evaluation scores on unobserved data.

C. PERFORMANCE COMPARISON OF MODEL BEFORE AND AFTER HYPERPARAMETER TUNING

In this section, the performance of IndoBERT-base-p1 is evaluated before and after applying the hyperparameter tuning in classifying the fake news, particularly in correctly classifying the news with label "fake." The performance of IndoBERT-base-p1 after the tuning process with grid search, random search, and Bayesian optimization using the optimal hyperparameter values shown in Table III was compared. Table VII until Table X respectively illustrate the evaluation of the model before and after hyperparameter tuning in terms of precision, recall, F1-score, and accuracy and each hyperparameter tuning method improvement in terms of percentage to the method with fine tuning.

It is clear that the model after the hyperparameter tuning, specifically Bayesian Optimization technique, outperformed the model with fine-tuning in all the evaluation metrics assessed. Bayesian optimization is regarded 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 pretrained models on

JURNAL NASIONAL TEKNIK ELEKTRO DAN TEKNOLOGI INFORMASI
Volume 13 Number 1 February 2024

TABLE VII
PERFORMANCE COMPARISON OF FINE-TUNING VS HYPERPARAMETER TUNING IN TERMS OF PRECISION

Methods	Precision	Improvement
Model with Fine-Tuning	0.8632	-
Model with Grid Search	0.8661	0.33%
Model with Random Search	0.8462	-2.01%
Model with Bayesian Optimization	0.8879	2.78%

TABLE VIII
PERFORMANCE COMPARISON OF FINE-TUNING VS HYPERPARAMETER TUNING IN TERMS OF RECALL

Methods	Recall	Improvement
Model with Fine-Tuning	0.9266	-
Model with Grid Search	0.8899	-4.12%
Model with Random Search	0.9083	1.95%
Model with Bayesian Optimization	0.9450	-

TABLE IX
PERFORMANCE COMPARISON OF FINE-TUNING VS HYPERPARAMETER TUNING IN TERMS OF F1-SCORE

Methods	F1-Score	Improvement
Model with Fine-Tuning	0.8938	-
Model with Grid Search	0.8899	-0.44%
Model with Random Search	0.8761	-2.02%
Model with Bayesian Optimization	0.9156	2.38%

TABLE X
PERFORMANCE COMPARISON OF FINE-TUNING VS HYPERPARAMETER TUNING IN TERMS OF ACCURACY

Methods	Accuracy	Improvement
Model with Fine-Tuning	0.9313	-
Model with Grid Search	0.9194	-1.29%
Model with Random Search	0.9164	-1.63%
Model with Bayesian Optimization	0.9432	1.26%

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 evident that hyperparameter tuning in this research could enhance the model's performance in classifying fake news. In addition to that, hyperparameter optimization can involve non-linear interactions between parameters, which poses a challenge to conventional methods such as grid search or random search to explore the space effectively. Bayesian optimization, on the other hand, utilizes probabilistic modeling and Bayesian inference to better capture these nonlinear relationships, making it more suitable for complex hyperparameter tuning tasks. The grid search method comes as the second best for the precision. In comparison to the model with fine-tuning, it has a slightly different performance in terms of recall and F1-score. However, in terms of accuracy, fine-tuning ranks second best.

D. TIME-COST COMPARISON OF MODEL BEFORE AND AFTER HYPERPARAMETER TUNING

Table XI depicts the efficiency comparison of IndoBERT-base-p1 with fine tuning method and after performing hyperparameter tuning using grid search, random search, and Bayesian optimization based on the time required to obtain the optimal hyperparameter value to complete the classification.

TABLE XI
TIME-COST COMPARISON OF FINE-TUNING VS HYPERPARAMETER TUNING

Methods	Time Cost (minutes)	Additional Time (%)
Model with Fine Tuning	70	-
Model with Grid Search	88	25.71
Model with Random Search	77	10.00
Model with Bayesian Optimization	84	20.00

When comparing the model with fine-tuning to the model with hyperparameter tuning, it is obvious that the model with hyperparameter tuning required additional time, which was used to find the optimal hyperparameter values. Nevertheless, this additional time was only required once, after which it can be used again for the classification tasks. Therefore, this additional time is tolerable due to the enhanced performance of the hyperparameter tuning model.

VI. CONCLUSION AND SUGGESTION

Based on the results and discussion, it can be seen that the Bayesian Optimization hyperparameter tuning method demonstrates superior performance on the IndoBERT-base-p1 model using the TurnBackHoax.ID dataset compared to other methods, including the model without hyperparameter tuning. Specifically, it achieved the highest F1-score in correctly identifying news data labeled as "fake," with optimal parameter values of 16 for batch size, 2 × 10-5for learning rate, and 8 for the number of epochs.

Based on the experimental comparison between the fine-tuning method of the IndoBERT model and the other three hyperparameter tuning methods, it can be concluded 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.ID dataset. On the contrary, the evaluation results from grid search and random search hyperparameter tuning did not demonstrate any improvements in the evaluation results during fine-tuning. It is due to the fact that the evaluation value generated after hyperparameter tuning using grid search and random search decreased when compared to the evaluation values achieved during the initial fine-tuning process.

Regardless of the improvement in terms of effectiveness, Bayesian optimization needs additional time to determine the optimal values of hyperparameter. Therefore, future research should consider a comparative evaluation of the efficiency and effectiveness of evolutionary algorithms, including gradient-based optimization, which operates by gradually updating the hyperparameter values until an optimal solution is attained.

AUTHORS' CONTRIBUTIONS

Conceptualization, methodology and model, Rosni Lumbantoruan, Anugerah Simanjuntak, Kartika Sianipar, Rut Gultom; validation, Rosni Lumbantoruan, Samuel Situmeang, Mario Simaremare, Erwin Panggabean; formal analysis, Rosni Lumbantoruan, Anugerah Simanjuntak; investigation, Kartika Sianipar, Rut Gultom; resources; Anugerah Simanjuntak, Kartika Sianipar, Rut Gultom; data curation, Anugerah Simanjuntak, Kartika Sianipar, Rut Gultom; writing—original draft preparation, Rosni Lumbantoruan, Anugerah

Simanjuntak, Rut Gultom, Kartika Sianipar; writing—review and editing, Rosni Lumbantoruan, Samuel Situmeang; visualization, Anugerah Simanjuntak; supervision, Rosni Lumbantoruan; project administration, Rosni Lumbantoruan.

ACKNOWLEDGMENT

This research is fully funcded by the Institute of Research and Community Service (Lembaga Penelitian dan Pengabdian kepada 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 (2021) "Pengguna internet Indonesia peringkat ke-3 terbanyak di Asia," [Online], https://databoks.katadata.co.id/datapublish/2021/10/14/pengguna-internet-indonesia-peringkat-ke-3-terbanyak-di-asia, access date: 10-Jan-2023.

[2] M.A. Rahmat and I.S. Areni, "Hoax web detection for news in Bahasa using support vector machine," *2019 Int. Conf. Inf. Commun. Technol. (ICOIACT)*, 2019, pp. 332–336, doi: 10.1109/ICOIACT46704.2019.8938425.

[3] A. Thota, P. Tilak, S. Ahluwalia, and N. Lohia, "Fake news detection: A deep learning approach," *SMU Data Sci. Rev.*, vol. 1, no. 3, pp. 1–20, 2018.

[4] R. Lumbantoruan *et al*., "Analysis comparison of FastText and Word2vec for detecting offensive language," *2022 IEEE Int. Conf. Comput. Sci. Inf. Technol. (ICOSNIKOM)*, 2022, pp. 1–8, 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," *ISEAS-Yusof Ishak Inst.*, vol. 2019, pp. 1–12, Nov. 2019.

[6] A. Yuliani (2017) "Ada 800.000 situs penyebar hoax di Indonesia," [Online], https://www.kominfo.go.id/content/detail/12008/ada-800000-situs-penyebar-hoax-di-indonesia/0/sorotan_media, access date: 10-Jan-2023.

[7] R. Sultana and T. Nishino, "Fake news detection system: An implementation of BERT and boosting algorithm," *Proc. 38th Int. Conf. Comput. Their Appl.*, 2023, pp. 124–137, doi: 10.29007/d931.

[8] L.H. Suadaa, I. Santoso, and A.T.B. Panjaitan, "Transfer learning of pre-trained transformers for COVID-19 hoax detection in Indonesian language," *Indones. J. Comput. Cybern. Syst. (IJCCS)*, vol. 15, no. 3, pp. 317–326, Jul. 2021, doi: 10.22146/ijccs.66205.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *Proc. 2019 Conf. N. Am. Chapter Assoc. Comput. Linguist., Hum. Lang. Technol.*, 2019, pp. 4171–4186, 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," *Proc. 6th Int. Conf. Sustain. Inf. Eng. Technol.*, 2021, pp. 247–251, doi: 10.1145/3479645.3479666.

[11] M. Guderlei and M. Aßenmacher, "Evaluating unsupervised representation learning for detecting stances of fake news," *Proc. 28th Int. Conf. Comput. Linguist.*, 2020, pp. 6339–6349, doi: 10.18653/v1/2020.coling-main.558.

[12] R.R. Rajalaxmi *et al*., "Optimizing hyperparameters and performance analysis of LSTM model in detecting fake news on social media," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, to be published, doi: 10.1145/3511897.

[13] N. Kanagavalli, S.B. Priya, 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, pp. 958–963, 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," *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 4, no. 4, pp. 655–663, Aug. 2020, doi: 10.29207/resti.v4i4.2038.

[15] M.E. Peters *et al*., "Deep contextualized word representations," *Proc. 2018 Conf. N. Am. Chapter Assoc. Comput. Linguist., Hum. Lang. Technol.*, 2018, pp. 2227–2237, doi: 10.18653/v1/N18-1202.

[16] R.K. Kaliyar, A. Goswami, and P. Narang, "FakeBERT: Fake news detection in social media with a BERT-based deep learning approach," *Multimed. Tools Appl.*, vol. 80, no. 8, pp. 11765–11788, Mar. 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," *Proc. 28th Int. Conf. Comput. Linguist.*, 2020, pp. 757–770, 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, no. 3, pp. 289–297, Mar. 2022, doi: 10.24507/icicel.16.03.289.

[19] B. Bischl *et al*., "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *WIREs Data Min. Knowl. Discov.*, vol. 13, no. 2, pp. 1–43, Mar./Apr. 2023, doi: 10.1002/widm.1484.

[20] T. Akiba *et al*., "Optuna: A next-generation hyperparameter optimization framework," *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2019, pp. 26232631, doi: 10.1145/3292500.3330701.

[21] S.A. Alasadi and W.S. Bhaya, "Review of data preprocessing techniques in data mining," *J. Eng. Appl. Sci.*, vol. 12, no. 16, pp. 4102–4107, Sep. 2017, doi: 10.3923/jeasci.2017.4102.4107.