Scientific Journal of Informatics
Vol. 10, No. 3, Aug 2023

p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

Performance Improvement of Fake News Detection Models Using Long Short-Term Memory Hyperparameter Optimization

Lindawati[1*], Muhammad Fadli Ramadhan[2], Sopian Soim[3], Nabila Rizqia Novianda[4]
[1,2,3]Department of Electrical Engineering, Politeknik Negeri Sriwijaya, Indonesia
[4]Department of Electronics Engineering, National Chin-yi University of Technology, Taiwan

Abstract.
Purpose: The proposed model was developed based on prior research that distinguished between fake and real news using a deep learning-based methodology and an LSTM neural network, with a model accuracy of 99.88%. This study uses hyperparameter tuning techniques on a Long Short-Term Long Memory (LSTM) neural network architecture to improve the accuracy of a fake news detection model.
Methods: To improve the accuracy of the fake news detection model and optimize the model from previous research, this study uses the hyperparameter tuning technique on models with Long Short-Term Memory (LSTM) neural network architecture. For this technique, three different types of experiments, hyperparameter tuning on the LSTM layer, Dense layer, and Optimizer, were conducted to obtain the best hyperparameters in each layer of the model architecture and the model parameters proposed. The fake and real news dataset, which has also been used in earlier studies, was used in this study.
Results: The proposed model could detect fake news with a high accuracy of 99.97%, surpassing the previous research models with an accuracy of 99.88%.
Novelty: The novelty of this study was hyperparameter tuning technique on different layers of the LSTM neural network to optimize the fake news detection model. The research aims to improve upon previous approaches and increase the accuracy of the model.

Keywords: Fake news detection, Long short-term memory, Hyperparameter optimization, Performance improvement, Machine learning models.

Received June 2023 / **Revised** July 2023 / **Accepted** August 2023

INTRODUCTION

Many emerging technologies help understand human behavior in this technological world. Emerging technologies can assist governments in developing safety policies. One of the unique technologies developed by John McCarthy in 1955 is artificial intelligence. Afterward, neural networks, machine learning, deep learning, natural language processing, and predictive analytics were developed [1]. The emergence of new technologies has significantly advanced in every field of life [2].

One of the emerging technologies that is altering how we handle business issues is artificial intelligence [3], [4]. Machine learning and advanced data analytics are being used by more and more companies to solve problems. Natural Language Processing (NLP), which has improved during the artificial intelligence era, provides much potential for companies wishing to interpret human behavior using existing data [5].

All forms of communication in social and natural settings, including audio, video, and text, can be used with NLP. Text mining has been helping identify numerous relevant patterns and trends in the textual collection. By strategically utilizing NLP in today's market environments, organizations can gain an edge over rivals. The vast amounts of unstructured data in various fields, as well as public opinion in the government sectors, in areas including healthcare, education, fake news, economic sectors, security, and trust, can be fought with artificial intelligence and natural language processing. The use of natural language processing improves communication between humans and robots, which improves decision-making and increases overall company productivity [6].

*Corresponding author.
Email addresses: lindawati@polsri.ac.id (Lindawati)
DOI: 10.15294/sji.v10i3.45420

Unrecognized fake news tends to spread more quickly. Fake news is currently more popular on social media platforms, such as Facebook News Feed, than in the past when it was prevalent in print. The rise of fake news has been connected to post-truth politics and political extremism. To better comprehend how fake news spreads, the authors [7] examined a dataset of Twitter rumor chains spanning from 2006 to 2017. The study aimed to understand and evaluate the influence of fake news on society. The authors observed the wide-scale distribution of inaccurate information, with around 3 million people spreading approximately 126,000 instances of fake news. Therefore, for the benefit of society and government backing, bogus news must be removed. Recognizing fake news before being spread is meaningful and valuable because of the fast-growing social media database and technological advances.

The importance of identifying fake news has been highlighted. A method has been devised to detect fake news, such as neural network method used in research [2], [8]–[12], which is very commonly used for classification cases. Apart from deep learning techniques, several traditional learning methods are also used for classification, such as Support vector machines (SVM) [13] and decision trees [14]. However, using deep learning methods such as neural networks is superior to these two methods because they are more scalable, and higher accuracy can be achieved by increasing the network size or training data set [15]. In addition, [16] shows that traditional decision trees and Support Vector Machine learning methods are inefficient for many modern applications. This means that traditional learning methods require a large number of observations to achieve generalization and impose significant manpower to determine prior knowledge in models. The neural network method can solve the problem of binary text classification to identify fake and real news. However, the neural network method has weaknesses, as it is difficult to select the optimal hyperparameter layers of the neural network model and requires much experimentation to determine a parameter. In addition, it is optimally used for a particular dataset.

Previous studies regarding the detection of fake news, [2], [8]–[12], used various optimization techniques and methods to enhance the performance of the neural network model. Studies [8]–[12] were chosen since they were compared in [2], so it is related to the comparison of the performance of the detection model in this research. The datasets used in these six studies differ, but in essence, the accuracy of the models proposed was assessed in the six studies, comparing the techniques used to create models for the fake news identification [17]. Kaliyar in [9] used the DeepFake multi-layer deep neural network method with a model accuracy of 88.64%. In addition to the DeepFake method, Goswami in [10] used the Echo FakeD method to optimize the coupled matrix-tensor factorization approach with the obtained model accuracy of 92.30%. Raj [12] used optimization of the neural network model using the coupled ConvNet method or the CNN framework for detecting fake news with a model accuracy of 93.56%. The BERT (Bidirectional Encoder Representations from Transformers) method can also be used to optimize deep learning techniques like CNN and LSTM to deal with the ambiguity of natural language understanding by models [18]. The method used by Narang in [8] achieved a model accuracy of 98.90%. Ozbay in [11] also used the neural network model optimization method using the improved Salp Swarm Optimization (SSO) method to detect fake news on social media, with a model accuracy of up to 99.50%. The effectiveness of the neural network model in the five previous studies was surpassed with the LSTM method by Chauhan [2], where the parameters used can achieve a model accuracy of 99.88%.

Referring to the background, a better neural network model to use for performing NLP tasks, especially for detecting fake news, is the LSTM neural network model. However, higher accuracy can be achieved by increasing the network size or training data set. This indicates that by choosing efficient and optimal hyperparameters for the fake news detection model and further research, the accuracy of neural network model can be increased [1], [19]. Therefore, we use the hyperparameter tuning method on the LSTM neural network model to improve the performance of the model for fake news detection. In this paper, the authors would optimize the hyperparameters in the LSTM neural network architectural model using the Hyperparameter tuning method to get better model accuracy performance achieved by [2]. Thus, the model proposed could help researchers and machine learning developers in the development of neural network models to prevent the spread of fake news on social media, such as Twitter.

METHODS

A system block diagram of the entire system is created as the first step in the research design process. In the research design process, flowchart is essential to give a general overview of how the research suite functions. This flowchart helps understand the steps in the research methodology. Consequently, a thorough research block diagram aids in developing a system that can work effectively. The steps taken in conducting the research method are shown in Figure 1.

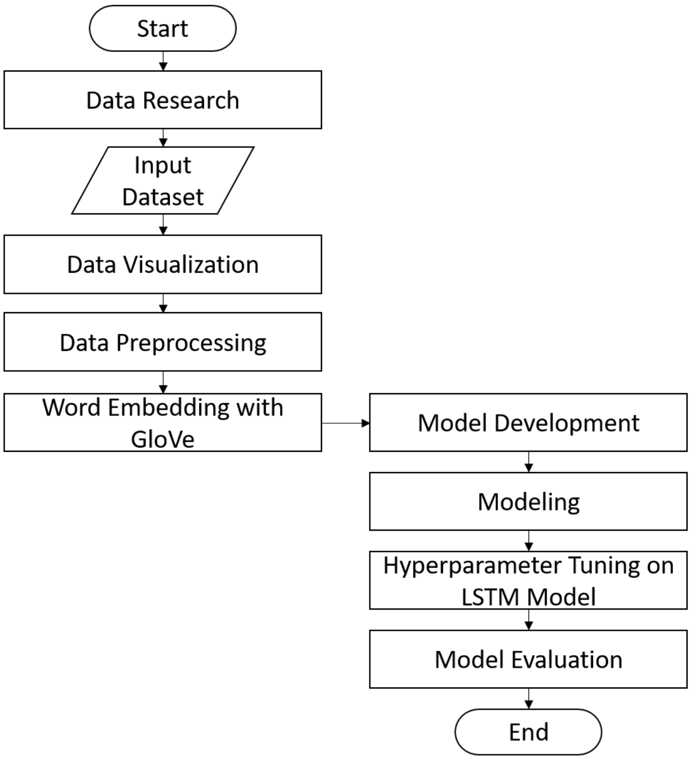

Figure 1. Flowchart of research methodology

Data Research

Data research examines the model used in previous research to get a better model evaluation value. Data research uses the Code feature on Kaggle in Jupyter Notebook and Python. The stages and detailed explanations of the data research are presented in the following sections.

Datasets

The public dataset[2] used is obtained from the Kaggle website, which has been tested in [20] and [2]. The dataset contains around 40,000 articles covering fake and real news. The fake and real news data are separated into two sets of around 20,000 articles. The dataset contains four labels, namely title, text, subject, and date. Glove Twitter[3] is also used with pre-trained word embedding data available on the Kaggle website, which has been identified in [21]. The word embedding data has specifications of 2B tweets, 27B tokens, 1.2M vocabs, and 100-dimensional tweet vector word embedding. This information is useful for the input layer that embeds the model.

Data Visualization

Data visualization simplifies the understanding of comparative data by visually representing it using graphs or maps. This enhances the ability to identify trends and insights in large datasets, leveraging the human mind's natural analytical capabilities. The dataset is divided into two categories: real news (class '1') and fake news (class '0').

---

[2] https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset
[3] Glove Twitter Source: https://nlp.stanford.edu/projects/glove/

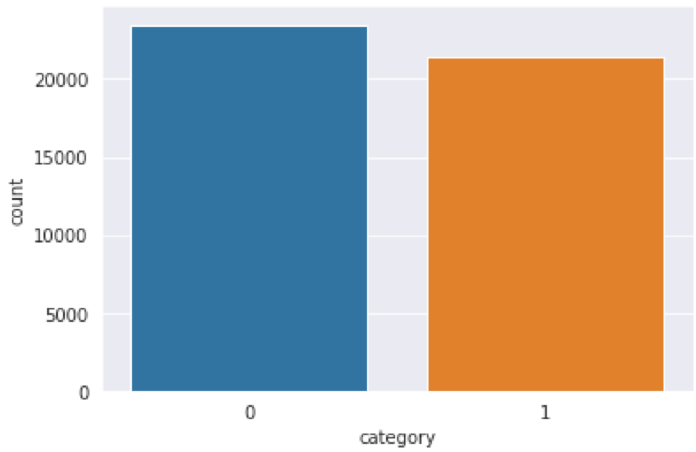

Figure 2. Dataset visualization

Figure 2 shows an article on the corresponding fake and real class labels. There are not many differences between the two datasets, so they look balanced. In addition, class '0' (blue) represents the wrong news category while '1' (orange) represents the real news category. After observing the graph, the number of datasets representing the original news category is 21,417 data, and the fake news category is 23,481, making it 44,898 total data. The corresponding subject, title, and date of each story can be omitted, leaving only the main text for further processing because the content in the subject section differs in the two categories.

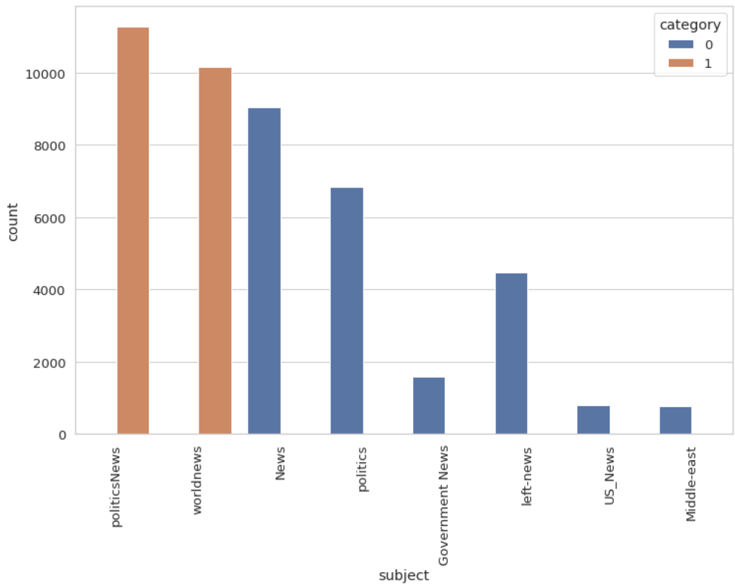

Figure 3. Topics on fake and real news categories

Figure 3 describes the various subjects that make up the dataset. The quantity of pertinent topics reflects how widely news is spread throughout the populace. The blue bar displays unreliable or false information, while the orange bar displays trustworthy and accurate news. Real news includes, among its topics, coverage of politics and world news. On the other hand, fake news discusses politics, the government, the left news, the United States, and Middle Eastern news.

Figures 4 and 5 display the keywords found in the real and fake news datasets, respectively. A word cloud is generated for each category, including up to 2000 words. Word clouds visualize groups of words by representing and emphasizing them in various sizes and lengths.

Figure 4. Word cloud representation for the real news dataset

Figure 4 shows the louder and bigger words observed are Us, Said, United, State, Donald, and Trump, contributing to the real news dataset.

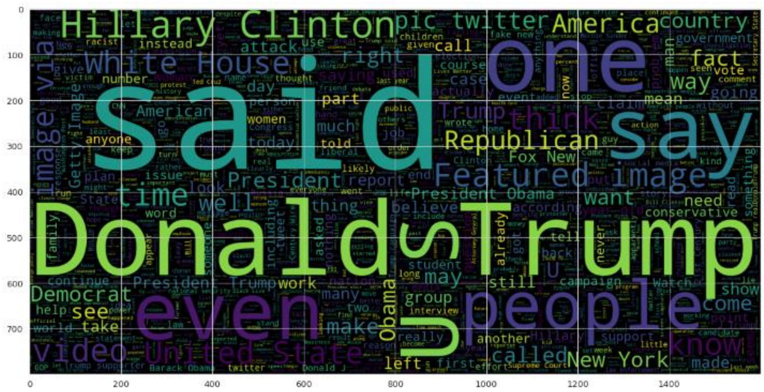

Figure 5. Word cloud representation for the fake news dataset

In Figure 5, the bold and prominent words such as "one," "said," "Donald," "Trump," "Hillary," and "Clinton" indicate their contribution to the fake news dataset. The size and boldness of a term in the word cloud represent its frequency and significance within the document. The words are extracted from separate categories, one for fake news and the other for real news, and individual word cloud is created based on each category.

Data Preprocessing

At this stage, data are manipulated from the fake news dataset used before being executed by the LSTM model network, which involves data cleansing and transformation techniques. The data cleaning stage aims to process unwanted noise in text data, so it is easier for the dataset to input LSTM model data, such as removing stop words. Stop words are common English words that have minimal impact on the overall meaning of a sentence. They can be disregarded without altering the sentence’s significance. Examples of stop words include "the," "he," and "have." [22]. Data transformation is the second stage of data preprocessing. Tokenization is used in data transformation to make the processed data easier for the input model to understand. The process of turning each word into a number is known as tokenization. Tokenization divides a given corpus into units called tokens [23]. Moreover, tokenization parses texts to eliminate specific words (tokenization) so textual data can be used for predictive modeling. Feature extraction is the process of converting words into integers or floating-point numbers for use as input in machine-learning techniques.

Word Embeddings: Glove

In this step, word vector representations are generated to capture meaningful relationships within the word vector space. This is achieved through training on a statistical corpus that measures global word-by-word co-occurrence [24]. This study uses GloVe Twitter, a pre-trained word embedding obtained on a dataset provider site for data scientists and machine learning developers, namely Kaggle[4]. Glove Twitter has a Massive data scale and Relevance to natural language processing. Twitter is a very popular social media platform, with millions of users sending millions of tweets every day. The data available from Twitter is huge, covers a wide variety of topics, and reflects the diversity of languages and writing styles that exist on the internet. This data is used for data input at the embedding layer of the neural network architecture. The model demonstrates how co-occurrence probability can be applied to the corpus to extract particular underlying meanings [24]. Consider the words a and b from the corpus to clarify this idea. The ratio of these two words' co-occurrence probabilities with probe words, *c*, can be used to confirm the relationship between them. For *c* related to *a* but not *b*, the ratio $\frac{Pa|c}{Pb|c} > 1$. Similarly, for *c* related to *b* but not *a*, the ratio $\frac{Pa|c}{Pb|c} < 1$. If *c* is either related to both *a* and *b* or not related to both *a* and *b*, the ratio $\frac{Pa|c}{Pb|c}$ is close to 1. Therefore, it implies that determining the ratio of the probabilities of word co-occurrence is the first step in learning word vectors. Eq.1 shows the model's general form.

$$F\ (w_a,\ w_b,\ w^{\#}_c) = \frac{Pa|c}{Pb|c} \tag{1}$$

where $w \in \mathbb{R}^d$ are word vectors and $w^{\#} \in \mathbb{R}^d$ are probe word vectors in the corpus.

Model Development

The research model is developed to create a fake news detection model with greater model accuracy than what earlier studies proposed. After conducting research on the processed dataset, this stage is attained. The process of creating a model, or modeling, and developing the model using Hyperparameter Tuning optimization are both included in the model development stage.

Modeling

After transforming the data at the data preprocessing stage, the next step is building a neural network model. The LSTM model has the advantage of solving sequential data problems. The challenge lies in dealing with sequences of different lengths, diverse vocabulary of input symbols, and the need for models to grasp long-term context and relationships within the input sequence of symbols. The Long Short-Term Memory (LSTM) model is employed for fake news detection.

The following is the LSTM and Hyperparameter model layer architecture in [2] which was used as an initial model for the hyperparameter tuning technique.

Table 1. LSTM layer architecture in previous research [2]

Layer (type)	Output size	Number of Paramet
Embedding_1 (embedding)	300 x 100	1.000.000
Lstm_1 (LSTM)	300 x 128	117,248
Lstm_2 (LSTM)	128	49,408
Dense_1 (Dense)	64	2080
Dense_2	1	33

Table 2. Hyperparameters in previous research [2]

Parameter	Value
Embedding layer	1
LSTM layer	2
Dense layer	2
Loss Function	Binary cross entropy
Activation function	ReLu
Optimizer	Adam
Learning_rate	0.01
Number of epochs	10
Embedding size	100
Batch size	256

[4] https://www.kaggle.com/datasets/bertcarremans/glovetwitter27b100dtxt

Hyperparameter Tuning on the LSTM Model

In this research, we used the Hyperparameter Tuning method on the LSTM model for the fake news detection model. This method contributes to improving the accuracy of the detection model in [2].

In the context of the LSTM (Long Short-Term Memory) model, hyperparameter tuning optimizes the hyperparameter values used to construct the LSTM model. Hyperparameters are parameters that the model does not directly learn, but they have an impact on the model's functionality, rate of convergence, and capacity. Finding the right mix of hyperparameters to produce the model with the best or optimal performance is the goal of this hyperparameter tuning. Model performance can be evaluated using metrics like accuracy, precision, recall, F1-score, and mean squared error (MSE), depending on the type of problem being solved or other pertinent evaluation metrics. Model accuracy is the evaluation metric used in this research.

To find the ideal combination that results in the best model performance, hyperparameter tuning involves experimenting with different combinations of these hyperparameter values. Finding the ideal hyperparameter combination can be done manually by repeatedly trying different values, or it can be done using optimization techniques like grid search, random search, or other heuristic methods [25]. The Hyperparameter Tuning process in this study is the LSTM model hyperparameter tuning. It can be done manually, and the hyperparameter combinations are used in the LSTM model architecture, as in Table 1 and Table 2. In this research, the Hyperparameters in the LSTM model optimized in this study are as follows:

1. The LSTM Layer includes the number of layers, the number of LSTM units, and dropouts. For hyperparameter tuning experiments on the LSTM layer, 4 hyperparameter combination experiments are conducted.
2. Layer Dense includes the number of layers, number of dense units, and dropouts. For hyperparameter tuning experiments on the LSTM layer, 8 hyperparameter combination experiments are conducted.
3. Optimizer includes the learning rate. For hyperparameter tuning experiments on the LSTM layer, 3 hyperparameter combination experiments are conducted.

The hyperparameters in this LSTM model are selected based on the results of the best model accuracy of each combination in the Hyperparameter Tuning experiment at each layer. Moreover, they are used as the proposed LSTM model architecture for comparison with the models in [2], which used the same fake news datasets and detection models.

The LSTM model architecture that has been optimized using the hyperparameter tuning technique is expected to be able to build a fake news detection model with better model accuracy performance than the detection model in [1].

Model Evaluation

Testing the system's capacity to recognize fake news is the main goal of system evaluation. Measuring the degree to which the LSTM model can identify and differentiate between real and fake news is a key component of evaluating the accuracy of the LSTM model on the fake news detection model. The model evaluation metric used in this research is model accuracy.

Table 3. Confusion matrix

	FALSE POSITIVE (FP)	FALSE NEGATIVE (FN)
TRUE POSITIVE	TP	FN
TRUE NEGATIVE	FP	TN

The effectiveness of the classification model is assessed using a confusion matrix. It provides a visual summary of the model's predicted results by comparing the actual predictions with those predicted by the model.

In the context of fake news detection, the confusion matrix can be used to measure how well the model can distinguish between fake news and true news. For example, the confusion matrix can show how much fake news is successfully detected correctly (TP), how much correct news is detected correctly (TN), how much correct news is incorrectly detected as fake (FP), and how many false news is detected as true (FN).

Accuracy in the confusion matrix is one of the evaluation metrics calculated based on its entries [26]. Accuracy measures the proportion of accurate predictions among all predictions made:

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (2)$$

Based on entries in the confusion matrix, accuracy shows a general idea of how well the model can distinguish between false and real news. It should be understood, though, that accuracy can provide inaccurate information if the data is class unbalanced. The accuracy of the model under consideration in this research is expressed as accuracy metrics for the LSTM model's performance in identifying fake news from the dataset under consideration by the detection model.

RESULTS AND DISCUSSIONS

The LSTM model employed in the previous studies obtained the accuracy value of the detection model for predicting fake news datasets at 99.88%. The hyperparameter tuning optimization method, on the other hand, could enhance model accuracy by carrying out experiments to obtain the best LSTM model hyperparameter selection.

The results of this study are expected to improve the fake news detection model in [2] and increase the accuracy of the model in detecting fake and real news on social media by optimizing the hyperparameter setting of the LSTM Model.

The initial plan for the proposed LSTM optimization model architecture shows the hyperparameters for each layer in the LSTM neural network (Table 1). This model architecture was the experimental basis for better LSTM model optimization by modifying the hyperparameters for each layer in the LSTM model.

Real and fake news are distinguished by the LSTM model. The modeling processes were conducted in 15 experiments to determine the best hyperparameter to be used as the architecture of the LSTM model.

The following sections describe the experiments conducted in optimizing the LSTM model architecture Hyperparameters on the LSTM input layer, Dense layer, and Optimizer to get the Hyperparameter optimization results for each layer in the Proposed LSTM model.

Results of the Hyperparameter Optimization for Each Layer in the Proposed LSTM Model

The hyperparameters with the highest accuracy in each LSTM model layer as the LSTM model proposed are as follows:

1. **Results of the Hyperparameter Tuning Experiments on the LSTM Layer**
   The LSTM layer, which serves to remember and process input sequences, is the fundamental layer in the LSTM model architecture. The results of hyperparameter optimization experiments on the LSTM layer are shown in the following table.

Table 4. Results of the hyperparameter tuning experiments on the LSTM layer

Test	Hyperparameter Tuning				Model
	Number of Layers	Layer 1 Units	Layer 2 units	Dropout	Accuracy
1	2	64	32	0.2	0.9932
2	2	128	64	0.2	0.9980
3	2	256	128	0.2	0.9962
4	2	128	64	0.5	0.9962

   Table 4 shows four experiments conducted to obtain the LSTM layer hyperparameters with the best model accuracy. The values for the hyperparameter in the LSTM layer were obtained through two stages: determining the LSTM memory unit and the Dropout value. The experiment results show that the second experiment obtained the best model accuracy among the other experiments, with a model accuracy of 0.9980.

2. **Results of the Hyperparameter Tuning Experiments on the Dense Layer**
   The Dense layer in the LSTM model is used as an output layer to classify whether a text is fake news. The results of hyperparameter optimization experiments on the dense layer are shown in the following table.

Table 5. Results of the hyperparameter tuning experiment on the dense layer

Test	Hyperparameter Tuning				Model Accuracy
	Number of Layers	Number of Neurons 1	Number of Neurons 2	Dropout	
1	1	16	-	0.2	0.9969
2	1	32	-	0.2	0.9922
3	1	64	-	0.2	0.9953
4	1	128	-	0.2	0.9979
5	2	32	16	0.2	0.9936
6	2	64	32	0.2	0.9918
7	2	128	64	0.2	0.9981
8	2	128	64	0.5	0.9970

Table 5 shows eight experiments with three different selections of hyperparameters for the dense layer, including the number of layers, the number of neuron units, and the dropout value. the experiment results show that the 7th experiment obtained the highest dense layer hyperparameter, with a model accuracy of 0.9981.

3. **Results of the Hyperparameter Tuning Experiment on the Optimizer**

In this process, a model was arranged to be ready for training. The number of weighted inputs and biases in neural networks was determined using the activation function [27]. The results of the hyperparameter optimization experiment on the optimizer output layer are shown in the following table.

Table 6. Results of the hyperparameter tuning experiment on the optimizer

Test	Hyperparameter Tuning Learning Rate	Model Accuracy
1	0.001	0.9798
2	0.01	0.9965
3	0.1	0.5234

Table 6 shows that the hyperparameters on the Optimizer are the existing parameters for the model training process, meaning that the Optimizer is configured before the modeling process. In the hyperparameter optimization on the optimizer, the function of optimizer and activation of loss are the same, namely Optimizer Adam and loss Binary Cross Entropy. At the same time, the Learning Rate is only configured to be three tries. As a result, the second experiment with a 0.01 learning rate obtained a model accuracy of 0.9965.

The Proposed Model

The LSTM neural network model is proposed in this study. Tables 4, 5, and 6 show that the architecture of this model has been optimized using a hyperparameter selection technique or hyperparameter tuning for each layer in the LSTM model. This proposed model was trained and tested to show the performance of the model optimized using this technique.

Table 7. The proposed model of LSTM model architecture

Layer (type)	*Output size*	*Number of Paramet*
Embedding_25 (embedding)	300 x 100	1.000.000
Lstm_50 (LSTM)	300 x 128	117,248
Lstm_51 (LSTM)	64	49,408
Dense_57 (Dense)	128	8320
Dense_58 (Dense)	64	8256
Dropout_11 (Dropout)	64	0
Dense_59 (Dense)	1	65

The proposed model of LSTM network architecture is shown in Table 7. The proposed model includes details on the model architecture's layer type, output size, and parameter count. The output size in LSTM depends on how many units or output dimensions the LSTM layer generates. At every step, the LSTM layer's individual units generate output. The output size of the LSTM layer depends on how many units are present there. The weights and biases in the LSTM layer are included in the number of parameters in the LSTM model architecture. The input size, output size, and configuration of each LSTM layer affect the number of parameters in that layer.

Table 8. Hyperparameters in the proposed model

Parameter	*Value*
Embedding layer	1
LSTM layer	2
Dense layer	3
Dropuout	0.2
Loss Function	Binary cross entropy
Activation function	ReLu
Optimizer	Adam
Learning_rate	0.01
Number of epochs	10
Embedding size	100
Batch size	256

Table 8 shows the hyperparameters in the model proposed in this study. Compared with the hyperparameters in [1] as in Table 2, the difference is only in the number of dense layers, with 3 layers, and dropouts. The LSTM model performe better due to the larger number of memory units giving the model more ability to capture complex patterns in the input data. However, using too many memory units may result in overfitting or raise model complexity. There is an additional hyperparameter (retaining probability) introduced by the dropout function [28]. Dropout hyperparameters were employed to counteract this overfitting and stop it from occurring in the model. In a single LSTM layer, dropout during training disabled several units at random. Thus, the risk of overfitting was reduced, and the units did not overly reliant on one another. Dropouts improved model generalizability and lowered reliance on particular features in the training data.

The Performance of the Proposed Model

Prior to the modeling process, the dataset in this study was divided into train and test subsets with a ratio of 75:25. The proposed model was trained using a train subset and evaluated using a test subset. Model performance can be measured using model evaluation metrics of accuracy. The accuracy evaluation results for the proposed model are shown in Figure 6. Meanwhile, the Confusion Matrix Model is shown in Figure 7.

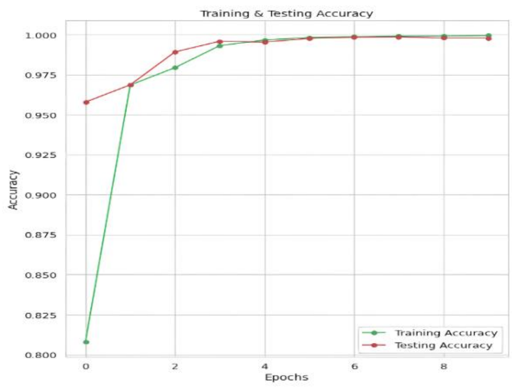

Figure 6. Accuracy of the proposed LSTM model

Figure 6 shows two indicators in the graph, namely Training accuracy (green line) and Testing accuracy (red line). Model performance on training and test datasets was evaluated using two metrics, training and test accuracy.

The term “training accuracy” describes how well the model can predict the training dataset. Calculated is the ratio of the number of accurate predictions to the total sample size of the training dataset. The model’s training accuracy describes how well it recognizes patterns in the training data and how well it can feature the data. Calculated accuracy measures the model’s capacity for accurate prediction on a test dataset unused for training. The test dataset’s sample size is divided by the percentage of correct predictions, which is calculated.

The accuracy test shows how well the model is able to generalize to new data that has never been seen before. The test accuracy is used to objectively measure model performance on independent data. If the test accuracy is not much different from the training accuracy, the model can generalize well and not overfit the training data.

In the model accuracy graph above, the training accuracy value in 10 epochs is 99.9703049659729%, while the test accuracy value is 99.7861921787262%. The accuracy value of this model shows that the test and training accuracy values are not significantly different, allowing for proper generalization and preventing overfitting of the training data. This is also influenced by the dropout layer used in the model architecture so the model accuracy is not too high or overfitting.

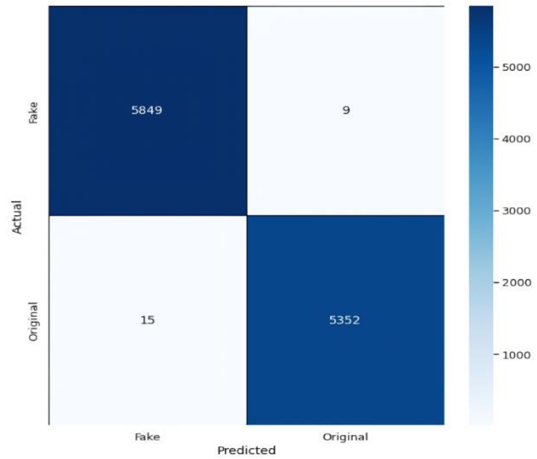

Figure 7. Confusion matrix on the proposed model

Figure 7 shows the confusion matrix table in the proposed model taken from the training data subset. The Confusion Matrix shows that the data tested/evaluated was 11225 data because the dataset division for the test data subset in this study was 25% of the total dataset. In this test data, it appears that the model can recognize fake news data and real news well. From 11225 test data, the model can recognize 5849 fake news data and 5352 real news data. The real news detected as fake news are 15 data, while the fake news detected as the real news are 9 data. Accuracy is determined by calculating the percentage of correct positive and negative predictions compared to the total data [29]. The result is an accuracy rate of 99.78619 percent.
Accuracy = (TP + TN) / (TP + TN + FP + FN) x 100%
Accuracy = (5352 + 5849) / (5352 + 5849 + 9 + 15) x 100%
Accuracy = 99.78619% of test accuracy.

Comparison of the Model Evaluation Results

After completing the model evaluation and obtaining information about the accuracy metrics of the proposed model, the next step is to compare the performance of the fake news detection model between the model proposed in this study and in [2]. This study compares the effectiveness of fake news detection models using model training accuracy metrics describing how well the model learns patterns in the training data and measures the extent to which the model can predict accurately on the training dataset, rather than giving a general overview of how well the model is able to generalize to new data that has never been seen before. This is done because this study uses the same dataset as in [2], To improve the performance of the fake news detection model compared with [2] , certain factors must be taken into account when developing the proposed model[2]. Table 9 and Figure 8 show the conclusions of this comparison.

Table 9. Comparison of detection model accuracy

Author	Method	Model Accuracy	Year
Lindawati, Muhammad Fadli Ramadhan, Sopian Soim, Nabila Rizqia Novianda	The LSTM neural network model with Hyperparameter Tuning	**99.97%**	**2023**
Tavishee Chauhan, ME, Hemant Palivela, PhD	The LSTM neural network model	**99.88%**	**2021**

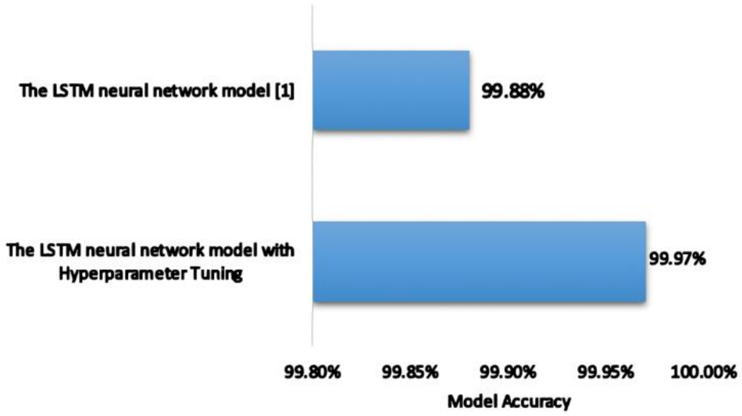

Figure 8. Comparative analysis of the proposed model with previous models

Table 9 shows a comparison of model accuracy performance of the Hyperparameter tuning method in this study with the models in [2]. To improve the performance of the fake news detection model, this study uses the hyperparameter tuning technique to determine the most effective hyperparameters, whereas [2] did not use optimization techniques in its model. This comparison allows us to assess how well each detection model performed when it came to identifying fake news in the dataset. A comparison of the proposed model's accuracy performance with models from earlier studies is shown in Figure 8. The comparison chart help better understand the discrepancies between the proposed model's accuracy and that of earlier research methods. In particular, the neural network-based model uses the hyperparameter tuning method for each layer of the LSTM model to determine how accurate the model is in differentiating between fake and real news data based on the training dataset. In addition, the comparison helps better understand how to increase the accuracy performance of the model.

It can be concluded that the hyperparameter tuning technique on the LSTM model to detect fake news training datasets was more accurate than the previous research models with an accuracy value of 99.97% and outperformed [2] with an accuracy of 99.88%. These results indicate that the LSTM model optimized using the Hyperparameter tuning technique on the model is able to improve the performance of the detection model. It was more accurate in detecting fake news datasets compared to the performance of models in previous studies even though the difference is only slightly, namely 0.09%, considering that the accuracy of the model in the prevuous study was very high and close to 100% accuracy. However, the results of this study contribute to enhancing the fake news detection model's performance, which is more accurate than earlier models. It is expected that these findings are able to assist other authors in enhancing the LSTM model's performance for identifying fake news with the methodology used in this study and be applied as fake news detection systems to assist people in dealing with the spread of fake news.

CONCLUSION

This study uses the Hyperparameter Tuning method to optimize the LSTM (Long Short-Term Memory) model to detect fake news more accurately than previous model. Despite using the same deep learning algorithm as earlier model, choosing an accurate hyperparameter can have an impact on how well the fake news detection model performs. The number of layers, memory units, and cells in each layer, the learning rate value on the pre-trained parameters, the pre-trained word embedding that serves as the input layer embedding, i.e., Glove, the number of epochs in the model training process, the type of activation function utilized, and the Dropout function to overcome overfitting are just a few of the variables that have an impact on the accuracy of this model. The results of this research can be a reference and consideration for researchers when optimizing the performance of the LSTM model in NLP (Natural Language Processing) cases using the Hyperparameter Tuning method. In the future, it is expected that there will be other neural network methods that can be used to improve the model's accuracy in detecting fake news and a project to implement a fake news detection model to assist the public in dealing with the spread of fake news.

REFERENCES

[1] J. I.-Z. Chen, H. Wang, and K.-L. Du, *Machine Learning and Autonomous Systems: Proceedings of ICMLAS 2021*. Springer Science and Business Media LLC, 2022.

[2] T. Chauhan and H. Palivela, "Optimization and improvement of fake news detection using deep learning approaches for societal benefit," *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 2, p. 100051, 2021, doi: 10.1016/j.jjimei.2021.100051.

[3] V. R. Palanivelu and B. Vasanthi, "Role of artificial intelligence in business transformation," *Int. J. Adv. Sci. Technol.*, vol. 29, no. 4 Special Issue, pp. 392–400, 2020.

[4] J. L. Ruiz-Real, J. Uribe-Toril, J. A. Torres, and J. D. E. Pablo, "Artificial intelligence in business and economics research: Trends and future," *J. Bus. Econ. Manag.*, vol. 22, no. 1, pp. 98–117, 2021, doi: 10.3846/jbem.2020.13641.

[5] Y. Kang, Z. Cai, C. W. Tan, Q. Huang, and H. Liu, "Natural language processing (NLP) in management research: A literature review," *J. Manag. Anal.*, vol. 7, no. 2, pp. 139–172, 2020, doi: 10.1080/23270012.2020.1756939.

[6] M. Bahja, "Natural Language Processing Applications in Business," *E-bus. - High. Educ. Intell. Appl.*, 2021, doi: 10.5772/intechopen.92203.

[7] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science (80-.).*, vol. 359, no. 6380, pp. 1146–1151, 2018, doi: 10.1126/science.aap9559.

[8] 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, 2021, doi: 10.1007/s11042-020-10183-2.

[9] R. K. Kaliyar, A. Goswami, and P. Narang, "DeepFakE: improving fake news detection using tensor decomposition-based deep neural network," *J. Supercomput.*, vol. 77, no. 2, pp. 1015–1037, 2021, doi: 10.1007/s11227-020-03294-y.

[10] R. K. Kaliyar, A. Goswami, and P. Narang, "EchoFakeD: improving fake news detection in social media with an efficient deep neural network," *Neural Comput. Appl.*, vol. 33, no. 14, pp. 8597–8613, 2021, doi: 10.1007/s00521-020-05611-1.

[11] F. A. Ozbay and B. Alatas, "Adaptive Salp swarm optimization algorithms with inertia weights for novel fake news detection model in online social media," *Multimed. Tools Appl.*, vol. 80, no. 26–27, pp. 34333–34357, 2021, doi: 10.1007/s11042-021-11006-8.

[12] C. Raj and P. Meel, "ConvNet frameworks for multi-modal fake news detection," *Appl. Intell.*, vol. 51, no. 11, pp. 8132–8148, 2021, doi: 10.1007/s10489-021-02345-y.

[13] W. F. Abror and M. Aziz, "Journal of Information System Bankruptcy Prediction Using Genetic Algorithm-Support Vector Machine (GA-SVM) Feature Selection and Stacking," vol. 1, no. 2, pp. 103–108, 2023.

[14] L. Alfaris, R. C. Siagian, A. C. Muhammad, and U. I. Nyuswantoro, "Classification of Spiral and Non-Spiral Galaxies using Decision Tree Analysis and Random Forest Model : An Investigation of the Zoo Galaxy Dataset," vol. 10, no. 2, pp. 139–150, 2023, doi: 10.15294/sji.v10i2.44027.

[15] H. Mostafa and X. Wang, "Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization," *36th Int. Conf. Mach. Learn. ICML 2019*, vol. 2019-June, pp. 8163–8172, 2019.

[16] M. Loni, S. Sinaei, A. Zoljodi, M. Daneshtalab, and M. Sjödin, "DeepMaker: A multi-objective optimization framework for deep neural networks in embedded systems," *Microprocess. Microsyst.*, vol. 73, Mar. 2020, doi: 10.1016/j.micpro.2020.102989.

[17] N. Guimarães, Á. Figueira, and L. Torgo, "Can fake news detection models maintain the performance through time? A longitudinal evaluation of twitter publications," *Mathematics*, vol. 9, no. 22, 2021, doi: 10.3390/math9222988.

[18] M. Madani, H. Motameni, and R. Roshani, "Fake News Detection Using Feature Extraction, Natural Language Processing, Curriculum Learning, and Deep Learning," *Int. J. Inf. Technol. Decis. Mak.*, pp. 1–36, Apr. 2023, doi: 10.1142/S0219622023500347.

[19] Sulistiana and M. A. Muslim, "Support Vector Machine (SVM) Optimization Using Grid Search and Unigram to Improve E-Commerce Review Accuracy," *J. Soft Comput. Explor.*, vol. 1, no. 1, pp. 8–15, 2020.

[20] H. Ahmed, I. Traore, and S. Saad, "Detecting opinion spams and fake news using text classification," *Secur. Priv.*, vol. 1, no. 1, p. e9, 2018, doi: 10.1002/spy2.9.

[21] J. Pennington, R. Socher, and C. Manning, "Glove: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, vol. 19, no. 5, pp. 1532–1543. doi: 10.3115/v1/D14-1162.

[22] Y. Nezu and M. Takao, *Extracting stopwords on social network service*. Information Modelling and Knowledge Bases XXXI 321, 2020.

[23] S. Choo and W. Kim, "A study on the evaluation of tokenizer performance in natural language

processing," *Appl. Artif. Intell.*, vol. 37, no. 1, 2023, doi: 10.1080/08839514.2023.2175112.

[24] J. Ravi and S. Kulkarni, "Text embedding techniques for efficient clustering of twitter data," *Evol. Intell.*, no. 0123456789, 2023, doi: 10.1007/s12065-023-00825-3.

[25] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020, doi: 10.1016/j.neucom.2020.07.061.

[26] M. A. Mahjoubi, S. Hamida, O. El Gannour, B. Cherradi, A. El Abbassi, and A. Raihani, "Improved Multiclass Brain Tumor Detection using Convolutional Neural Networks and Magnetic Resonance Imaging," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 3, pp. 406–414, 2023, doi: 10.14569/IJACSA.2023.0140346.

[27] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning," *Math. Comput. Sci.*, pp. 1–20, 2018.

[28] B. Ait Skourt, A. El Hassani, and A. Majda, "Mixed-pooling-dropout for convolutional neural network regularization," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 8, pp. 4756–4762, 2022, doi: 10.1016/j.jksuci.2021.05.001.

[29] W. Budiawan Zulfikar, A. Rialdy Atmadja, and S. F. Pratama, "Sentiment Analysis on Social Media Against Public Policy Using Multinomial Naive Bayes," *Sci. J. Informatics*, vol. 10, no. 1, pp. 25–34, 2023, doi: 10.15294/sji.v10i1.39952.