DOI: 10.22146/jnteti.v14i4.21069

Optimization of Parallel Neural Network Layer Configuration in English Text Sentiment Analysis

Agung Nugroho[1], Arief Setyanto[2]

[1] Magister of Informatics, Universitas AMIKOM Yogyakarta, Sleman, Yogyakarta, Indonesia
[2] Informatics, Faculty of Computer Science, Universitas AMIKOM Yogyakarta, Sleman, Yogyakarta, Indonesia

[Submitted: 12 June 2025, Revised: 12 August 2025, Accepted: 24 September 2025]
Corresponding Author: Agung Nugroho (email: agung4b08@students.amikom.ac.id)

ABSTRACT — Accuracy in analyst sentiment classification is very important so that the trained model can be implemented well to make business decisions. Researchers proposed a method for configuring neural network models arranged in parallel to improve classification accuracy. The results of the first stage, a bidirectional long short-term memory (Bi-LSTM) algorithm with Keras embedding with a sequential layer configuration, produced the best accuracy of 80.20%. The results of this first stage served as the baseline to be used as a reference for the combination in the second stage of the experiment. In the second stage of the experiment, a combination of the Bi-LSTM algorithm with other algorithms was carried out in parallel, such as gated recurrent unit (GRU), recurrent neural network (RNN), and Simple RNN with Keras embedding. It was found that the combination of three parallel layers of GRU-BiLSTM-RNN with Keras Embedding produced the highest accuracy for sentiment analysis of three classes, with a value of 88%. A statistical test of the t-test method was carried out with a critical p-value of 0.05 to prove the accuracy that has been produced between the sequential and the parallel configuration. The results of the t-test between the sequential configuration and the parallel configuration obtained a p-value of 0.5e-9 which is much smaller than the critical p-value of 0.05 so that in statistical testing the average accuracy produced from the two configurations is significantly different.

KEYWORDS — Sentiment Analysis, Parallel Layer, Bi-LSTM, GRU, Keras.

I. INTRODUCTION

Customer satisfaction is an important indicator in the business world, as it reflects whether the products or services offered are well-received by consumers and can still compete with business competitors [1], [2]. The number of active internet users in the world in 2021 was 4.13 billion [3], which continues to increase. Social media microblogs can provide the latest information regarding consumer opinions on the products and services provided, so that the sentiment obtained from social media can be used to evaluate the products and services provided [4].

In 2013, Google introduced Word2Vec [5], which had been trained on six billion words and became a popular method in the natural language processing (NLP) process. A year later, in 2014, global vectors for word representation (GloVe) were introduced [6], which combined global matrix factorization and local context window in the word embedding model. In 2017, FastText was introduced [7], which used a new approach method based on skip-grams and each word was represented in an n-gram character. Keras embedding is a layer that functions to represent words in vectors to be input for neural networks [8]. In prior research, bidirectional encoder representations from transformers (BERT) were introduced [9], which was designed to train two-dimensional representations of words. To improve BERT's capabilities, the robustly optimized BERT approach (RoBERTa) was introduced by changing the main hyperparameters and using mini-batches and byte pair encoding (BPE) [10]. Furthermore, decoding enhanced BERT with disentangled attention (DeBERTA) was introduced, which utilized disentangled attention.

To carry out the sentiment analysis process, it is necessary to apply an algorithm, either machine learning-based or deep learning-based, as mentioned in [11]. Sentiment analysis is a process that requires a sequence of data, so the algorithm used must be able to handle time series problems. Recurrent neural network (RNN), as mentioned in [12]. The weakness of RNN is that it is only able to see signals that have fewer than 10 steps, so it is susceptible to vanishing gradients.

Based on these problems, researchers tried to find a solution. According to [13], long short-term memory (LSTM) was introduced by modifying the RNN network. LSTM consists of input, forget, and output gates, which have the advantage of learning the previous 1,000 steps. Furthermore, bidirectional LSTM (BiLSTM), which was developed from LSTM, enables the analysis of the steps before and after the, thereby providing more complete information [14]. In addition, in 2014, the gated recurrent unit (GRU) was introduced [15]. The advantage of GRU is that it can store important information and simultaneously eliminate unimportant information. The main purpose of GRU is to simplify LSTM.

Research has been conducted on sentiment analysis research for hotel review services in three classes using the LSTM-GRU algorithm, resulting in 91% accuracy [16]. The dataset exhibited a fairly large imbalance, with positive sentiment comprising 92.3% of the total data. Hence, the dataset was not adequately representative of the negative and neutral classes. Researchers provide suggestions to overcome this problem by means of data augmentation, including creating synthetic samples for minority classes with the text paraphrasing method and creating text with neutral sentiment. In addition, researchers suggest utilizing a hybrid approach, namely the ensemble learning method, which combines several algorithms to get the advantages of each algorithm used.

Other research has conducted sentiment analysis research using the Bi-LSTM algorithm for the Amazon product review dataset in two classes, resulting in 91.40% accuracy [17]. The dataset consisted of 104,975 records of Amazon product reviews that were labeled as positive and negative. This study outperformed the sentiment analysis results conducted by previous researchers, which produced an accuracy of 95.74%.

In other research, sentiment analysis classification was conducted for the Amazon product review dataset using the GRU algorithm, resulting in 87% accuracy [18]. In addition to the GRU algorithm, the study also used the LSTM, k-nearest neighbor (KNN), random forest, naïve Bayes, and support vector machines (SVM) algorithms. The results indicated that the deep learning-based algorithms, namely GRU and LSTM, outperformed other machine learning-based algorithms. Among the machine learning models, SVM and logistic regression achieved the highest accuracy of 84%. This study suggests conducting a combination of experiments using the GRU and LSTM algorithms, which are commonly referred to as ensemble learning.

In addition, in the experiments conducted by researchers using deep learning-based algorithms, such as LSTM and BERT, the study showed that deep learning-based classification produced better accuracy. The BERT-based algorithm produced an accuracy of 83% and was still superior to previous results using RoBERTA [19], which produced an accuracy of 80.8%.

Prior research has stated that deep learning is one of the best techniques in the sentiment analysis classification process because it provides easy automation of feature extraction in text [20]. Researchers compared the accuracy of several contemporary deep learning-based algorithms in the classification process. The researchers suggest that other researchers conduct experiments using ensemble network algorithms, where several types of algorithms are combined in the sentiment analysis classification process to improve generalization in the feature extraction process.

Research [21] reinforces [20], suggesting that the application of ensemble learning in the sentiment analysis classification can increase the average accuracy by 5.53%. In addition, subsequent research further supports the theory that ensemble learning improves machine learning performance over a single model [22]. According to researchers, the stacking technique that combines different algorithms in the ensemble learning process is the most ideal because it can accommodate bias and variance at once. Moreover, ensemble learning configurations can be arranged in parallel and sequentially.

This research aims to improve the accuracy of sentiment analysis classification using the ensemble learning method of several algorithms arranged in parallel. This research can contribute to knowledge, including the influence of parallel neural network configuration on sentiment classification accuracy, learning rate hyperparameter tuning on accuracy, the influence of synthetic minority over-sampling technique (SMOTE) implementation on classification accuracy, the influence of word embedding used in sentiment classification, and the influence of k-fold cross validation implementation on the sentiment analysis classification process.

II. RELATED WORKS

In this section, the author presents several similar studies that have been conducted by previous researchers. In the study using the American Airlines Company dataset, including a study on sentiment analysis by [23]. The results of the experiments showed that the random forest classifier algorithm produced the highest accuracy with a value of 81.35%, followed by the adaptive boosting (AdaBoost) classifier algorithm, which produced an accuracy of 78.55%, and then followed by the decision tree classifier algorithm with an accuracy value of 75.88%.

Research [24] conducted a study using several algorithms. The results showed that SVM achieved an accuracy of 83.31%, followed by logistic regression at 81.81%, random forest at 78.55%, eXtreme Gradient Boosting (XGBoost) algorithm at 75.93%, naïve Bayes algorithm at 73%, and decision tree algorithm at 70.55%.

In [25], the proposed algorithm was linear regression combined with stochastic gradient descent classifier. The combination of the feature extraction term frequency (TF) method obtained an accuracy of 0.791, and the feature extraction term frequency–inverse document frequency (TF-IDF) method produced an accuracy of 0.792. Researchers also conducted experiments with the LSTM algorithm, but the results only achieved an accuracy of 0.68.

A study was conducted using seven classification algorithms for American Airlines Company sentiment analysis data [26]. Decision tree achieved an accuracy of 64.5%, random forest 86.5%, SVM 84.8%, Gaussian naïve Bayes 64.6%, AdaBoost 83.5%, logistic regression 81.9%, and k-nearest neighbor (KNN) 59.3%.

Prior research classified sentiment analysis of American Airlines Company data into three classes: positive, negative, and neutral [27]. The algorithms used were decision tree, naïve bayes, random forest, KNN, iterative dichotomiser (ID3), and random tree. In the first research setup with the unbalanced dataset, the highest accuracy was obtained by the naïve Bayes and ID3 algorithms, with an average accuracy of 58.89%. In the second scheme, namely the balance dataset, the naïve Bayes algorithm produced the highest accuracy with an average accuracy value of 76.10%. A prior study also classified American Airlines Company dataset into three classes and used random forest classifier as a baseline for machine learning-based experiments [28]. LSTM, Roberta, and electra-based models were also used to compare the accuracy of machine learning-based algorithms and deep learning-based algorithms. The BERT algorithm produced an accuracy of 83% and remained superior to prior research [19] conducted using RoBERTA, which produced an accuracy of 80.8%.

Based on previous related work using the same dataset, this research proposed a new method with a parallel neural network configuration based on BiLSTM, LSTM, GRU, and RNN to improve classification accuracy. The highest accuracy achieved in the previous study was still 84.50%. At the end of the study, 88% accuracy was obtained using the proposed method offered.

III. METHOD

Based on previous research references, the author identified a gap or potential for conducting further research. The literature indicates that the BERT, Bi-LSTM, LSTM, and GRU algorithms are the state of the art in the sentiment classification process. In addition, based on previous research and similar research, the research was conducted to compare the accuracy results of several different classification algorithms and was carried out in one stage with a sequential layer neural network configuration. At this stage, the author identified the potential to conduct further research with two schemes, namely the sequential layer process and the parallel layer process.

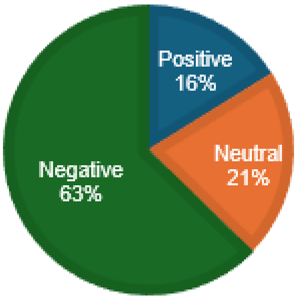


Figure 1. Dataset class distribution before applying SMOTE.

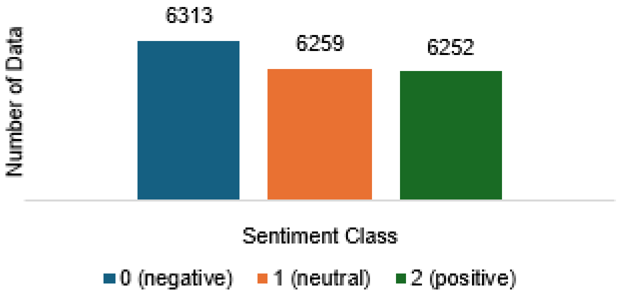


Figure 2. Dataset class distribution after applying SMOTE.

A. DATASET

The American Airlines Company dataset was obtained from www.kaggle.com and consisted of three sentiment classes: neutral, positive, and negative. Figure 1 depicts the composition of the dataset class used in this study.

This dataset contained 14,641 records related to customer opinions of six airlines in the United States, taken from the Twitter or X crawling process. On the Kaggle website, this dataset has a usability factor of 8.24 and has a CC BY-NC-SA 4.0 License. The sentiment distribution was 62.69% negative, 21.16% neutral, and 16.14% positive. This distribution is by the tendency of customers to express disappointment with a company’s service on social media, whereas if a service has met customer expectations, it tends to be silent and pass as something that usually happens.

To overcome the imbalance of dataset classes in this study, the SMOTE technique was used so that, in the training process, a more balanced amount of data was obtained in each class. After carrying out the SMOTE process, a more balanced dataset was obtained, with 6,313 data in the negative data class label (0), 6,259 data in the neutral data class label (1), and 6,252 data in the positive class (2). Balanced training data were visualized in Figure 2.

B. RESEARCH FLOW

Research began by downloading the American Airlines Company dataset from the Kaggle website. The dataset was then stored in a Google Drive folder to facilitate access using Google Colab Pro. In this study, the paid platform Google Colab Pro was used with system RAM specifications of 12.7 GB, GPU RAM of 15.0 GB, and disk capacity of 112.6 GB.

The next process was text preprocessing using a series of methods, such as case folding, filtering, tokenizing, and stemming. This process aimed to process the data to make them more structured. The case folding or lowercase process aims to standardize word characters into all lowercase letters. Filtering or stopword removal is the process of cleansing unnecessary words in the form of conjunctions, punctuation, characters, numbers, and white spaces. The stopword removal process is carried out to reduce the index size and processing time. This process needs to be carried out carefully as it can worsen the performance of sentiment analysis classification; some researchers suggest skipping this process. Stemming is the process of returning words to their basic form without prefixes and suffixes. Tokenizing is the process of breaking down words, so that they can be easily distinguished. This stage makes it easier to distinguish important words from other punctuation for the cleansing process.

The dataset that had been preprocessed was then divided into train and test data. The train test split process was carried out with a percentage of 70% data for training and 30% for testing, with a random state value during the splitting process of 22. To overcome the unbalanced amount of data in each sentiment class in the next process, the training data underwent the SMOTE process to achieve more balanced class distribution. During the data synthesis process for classes with a small number of used parameters, a sampling strategy used was set to auto and a random state was set to 42.

The built model process was carried out to compile the neural network configuration with the algorithm used and the combination of word embeddings used. This process is the most important stage in this research, as the combination of algorithms used was arranged into two schemes: the sequential layer scheme and the parallel layer scheme. Each neural network configuration scheme was also combined with several word embeddings for experimentation.

The research process was conducted using the first scheme, namely the sequential neural network configuration. At this stage, experiments were conducted using the LSTM, BiLSTM, and GRU algorithms combined with word embedding. The results of this experiment were taken as the highest accuracy results, which served as the baseline for the subsequent stage of the experiment. The subsequent experimental process was based on the first experimental scheme, carried out with a neural network model design arranged in parallel. In this study, the algorithms used were LSTM, BiLSTM, GRU, and Simple RNN, arranged in parallel with several combination schemes. The results of this experimental scheme yielding the highest accuracy value were used as the baseline for the selected model.

In each experimental process, the k-fold cross-validation method with 5 splits was applied to ensure the accuracy obtained was valid. The epoch value used was 100, the batch size was 32, the early callback parameter was set by monitoring val_loss with a patience value of 3 and restore_best_weights was set to true. Meanwhile, the learning rate value used varied between 1 and 0.0000001. Adam optimizer and softmax activation were used. In the training process, each learning rate value used was processed at a 5-fold cross-validation, and the results of the total training iterations carried out were processed to calculate the mean accuracy and standard deviation obtained.

Each experimental configuration was validated using the testing data and displayed in a classification report containing precision, recall, F1 score, and accuracy values. A confusion matrix was used to visualize the accuracy of the test results and was complemented by a receiver-operating characteristic curve (ROC) graph. In the evaluation stage, each experimental result was compared against its precision, recall, F1 score, and accuracy values. The experimental results with the highest accuracy were then statistically tested using the t-test method to determine whether a significant difference was present in the average accuracy of each experimental scheme.

TABLE I
SEQUENTIAL LAYER CONFIGURATION

Layer (Type)	Output Shape	Param #
embedding (Embedding)	?	0 (unbuilt)
spatial_dropout1d (SpatialDropout1D)	?	0
bidirectional (Bidirectional)	?	0 (unbuilt)
dense (dense)	?	0 (unbuilt)

Finally, to determine the impact of the parallel neural network configuration on classification accuracy, an ablation study was conducted. The results of this process were compared across accuracy metrics, the classification report, the confusion matrix, one-way analysis of variance (ANOVA), and the t-test.

C. SEQUENTIAL LAYER SCHEME

At this stage, a sequential layer configuration, in which one output layer became the input for the next layer, was examined. This first scheme was carried out using several algorithms that represent the state of the art of the sentiment analysis classification process, combined with various word embedding. The algorithms used in this first process were LSTM, GRU, and BiLSTM. The combination of word embedding used included Keras, Word2Vec Google News 300, GloVe Twitter 200, RoBERTA, BERT Large Uncased, FastText Subword 300, Deberta V3 fine-tuned, and Word2Vec self-train corpus dataset. The results of this first stage were used as the baseline for further research on the second stage scheme, namely the parallel layer configuration

The sequential layer configuration is presented in Table I. Sequential layer is a configuration where each layer will be the input for the next layer in sequence, and one layer will only be one source for the next layer. The training data were processed as input for the algorithm used, and combined with word embedding, the next process in the dense layer was the class decision process of sentiment analysis classification. The training data became input, then the embedding process was carried out on the neural network. Then, the output of the embedding process entered the dropout layer, the output of this layer then entered the LSTM layer. Finally, the output entered the dense layer for the classification process.

D. PARALLEL LAYER SCHEME

This stage is a continuation of the results of the first stage, in which the best results in the sentiment analysis classification process with sequential configuration were modified and combined with several algorithms arranged in parallel layers. The results of this process were compared for their accuracy, and the research results with the highest accuracy value were taken.

The results of the experiment on the first configuration served as a reference for adding algorithms arranged in parallel, in which a single output layer could act as the input for multiple subsequent layers. This scheme is illustrated in Table II. The train data became input for several algorithms that were arranged in parallel, and then the feature extraction process from each algorithm was combined before the classification process was carried out on the dense layer. The training data in the first stage became input and then the embedding process was carried out. As shown in Figure 3, in the parallel scheme, the output layer of the embedding layer becomes the input for the next three layers arranged in parallel, then the results of each layer are processed and flattened to be further processed and concatenated. From the concatenated layer, the classification is then processed on the dense layer. This configuration results in the highest classification accuracy in this study.

TABLE II
PARALLEL LAYER CONFIGURATION

Layer (Type)	Output Shape	Param #	Connected To
input_layer (InputLayer)	(None, 47)	0	-
embedding (Embedding)	(None, 47, 32)	359,968	input_layer [0][0]
bidirectional (Bidirectional)	(None, 47, 256)	164,864	embedding [0][0]
gru (GRU)	(None, 47, 128)	62,208	embedding [0][0]
flatten_1 (Flatten)	(None, 1504)	0	embedding [0][0]
flatten_2 (Flatten)	(None, 12032)	0	bidirectional
flatten_3 (Flatten)	(None, 6016)	0	gru
concatenate (Concatenate)	(None, 22368)	0	flatten_1 [0][0] flatten_2 [0][0] flatten_3 [0][0]
dense (Dense)	(None, 3)	58,659	concatenate

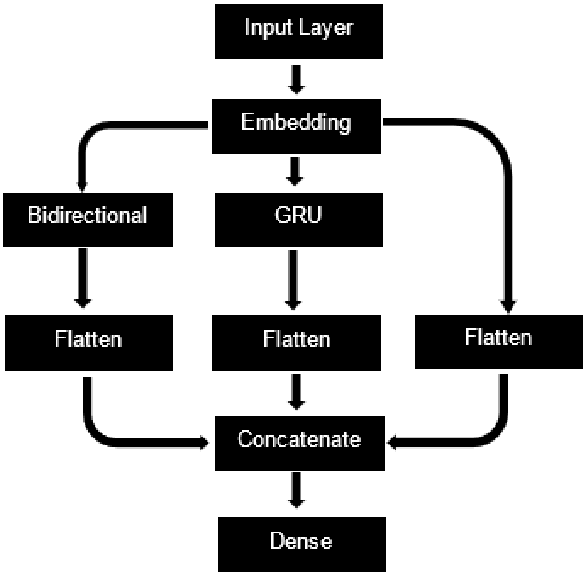


Figure 3. Parallel layer scheme.

E. EQUIPMENT USED

The research was conducted using Google Colab Pro with the runtime T4 GPU high RAM engine with Google Drive storage media. The Google Colab Pro platform was used because the initial experiment used the free version and found several limitations, including the limited compute units available, limited storage, limited memory, and limited time for the training process, so that the training process often stopped in the middle of the experiment process. After using the Google Colab Pro, the research could be continued more smoothly.

IV. RESULTS & DISCUSSION

This section discusses the results of the research that has been carried out. Table III exhibits the result of the experiment with a sequential layer scheme, while Table IV presents the

TABLE III
RECAP OF SEQUENTIAL LAYER EXPERIMENT RESULTS

No	Model Configuration	Layer Type	Embedding	Accuracy
1	BiLSTM	Sequential	Keras	80.20%
2	GRU	Sequential	Keras	79.30%
3	LSTM	Sequential	Keras	79.90%
4	LSTM	Sequential	Word2Vec	76.90%
5	LSTM	Sequential	GloVe	76.70%
6	LSTM	Sequential	RoBERTA	72.90%
7	LSTM	Sequential	BERT	72.70%
8	LSTM	Sequential	FastText	71.60%
9	LSTM	Sequential	Deberta	71.40%
10	LSTM	Sequential	Word2Vec self-train Corpus Dataset	68.20%

TABLE IV
RECAP OF PARALLEL LAYER EXPERIMENT RESULTS

No	Model Configuration	Layer Type	Embedding	Acc
1	2 Parallel GRU-RNN	Parallel	Keras	87.50%
2	2 Parallel GRU-RNN	Parallel	BERT	85.40%
3	3 Parallel GRU-BiLSTM-CNN	Parallel	Keras	87.50%
4	3 Parallel GRU-BiLSTM-RNN	Parallel	Keras	88.00%
5	3 Parallel GRU-SimpleRNN-CNN	Parallel	BERT	86.20%
6	3 Parallel GRU-SimpleRNN-RNN	Parallel	BERT	85.50%
7	4 Parallel GRU-BiLSTM-SImpleRNN-CNN	Parallel	Keras	86.10%
8	4 Parallel GRU-BiLSTM-SimpleRNN-RNN	Parallel	Keras	87.80%
9	4 Parallel GRU-LSTM-SimpleRNN-CNN	Parallel	Deberta	85.70%
10	4 Parallel GRU-LSTM-SImpleRNN-CNN	Parallel	RoBERTA	85.50%
11	4 Parallel GRU-LSTM-SImpleRNN-CNN	Parallel	BERT	85.30%
12	4 Parallel GRU-LSTM-SImpleRNN-CNN	Parallel	DeepSeek	80.80%

research scheme with a parallel layer configuration. In the first configuration scheme, the Bi-LSTM algorithm with Keras Embedding produced the highest accuracy of 80.20%, which outperformed the GRU algorithm with 79.30% and LSTM algorithm with 79.90%.

Table III also presents the results of the LSTM algorithm combined with several types of word embedding. Keras embedding produced the highest accuracy of 79.90%, followed by Word2Vec Google News 300 with 76.90%, and GloVe Twitter 200 with 76.70%. The lowest accuracy was obtained using the Word2Vec self-train corpus dataset embedding, which yielded an accuracy of 68.20%. This lower accuracy is likely due to the limited database of words used, especially when compared to other word embeddings that had been trained on a significantly larger number of words.

The Bi-LSTM and Keras embedding algorithms produced the highest accuracy in the first scheme. Hence, this algorithm and word embedding served as a reference or baseline for the second scheme, namely the parallel scheme, by combining other algorithms.

In the parallel layer research scheme, the best results in the sequential layer configuration were the benchmark for conducting experiments by adding other algorithms and arranging them in parallel. Table IV presents research findings, showing that the combination of the GRU-Bi-LSTM-RNN algorithm with Keras embedding, arranged in three parallel layers, produced the best accuracy of 88%. This was followed by the four-parallel-layer configuration of GRU-BiLSTM-SimpleRNN-RNN with Keras embedding, which achieved an accuracy of 87.80%. The third-best accuracy was the three-parallel-layer configuration of GRU-BiLSTM-CNN with Keras embedding, which produced an accuracy of 87.50%. Finally, the three-parallel-layer configuration of GRU-SimpleRNN-CNN with BERT large, uncased embedding achieved an accuracy of 86.20%. From the comparison of the word embedding used, the top three accuracies were achieved with Keras embedding, while the best BERT embedding accuracy was obtained from the three-parallel-layer configuration of GRU-SimpleRNN-CNN with an accuracy of 86.20%.

To measure improvements in classification accuracy, a t-test statistical method was used for each configuration and experimental stage. In the first stage of the t-test, a sequential configuration with the BiLSTM algorithm was compared with a parallel configuration for the BiLSTM-GRU-RNN algorithm. In this process, the experiment was carried out using the 5-fold cross-validation method to ensure more valid accuracy. The Epoch value used was 100, the batch size was 32, the early callback parameter was set by monitoring val_loss with a patience value of 3, and restore_best_weights was set to true. The learning rate value used varied between 1 and 0.0000001. In the training process, each learning rate value used was processed using 5-fold cross-validation, and the results of the total training iterations carried out were then processed to calculate the mean accuracy and standard deviation obtained.

Figure 4 presents the results of the t-test of sequential and parallel configurations. The results showed that the accuracy of the parallel layer configuration was better with low variances, while the sequential configuration produced lower accuracy with greater variances.

Table V exhibits the comparison of the average accuracy, standard deviation, t-test value, and p-value for two data groups: BiLSTM (sequential) and BiLSTM-GRU-RNN. In the t-test experiment of parallel and sequential configurations, the sequential configuration achieved an average accuracy of 0.7328 with a standard deviation of 0.06293. Meanwhile, the parallel configuration of the BiLSTM-GRU-RNN algorithm achieved an accuracy of 0.9924 with a standard deviation of 0.00285. In the calculation process, the t-value was 26.42, while the p-value was 0.000000005. From these results, since the p-value is smaller than 0.05, the null hypothesis can be rejected, indicating a fairly large difference in results between the two data groups.

The next statistical test was the influence level of the layer addition on the neural network model using the ablation study method. This test was conducted by training on the BiLSTM-

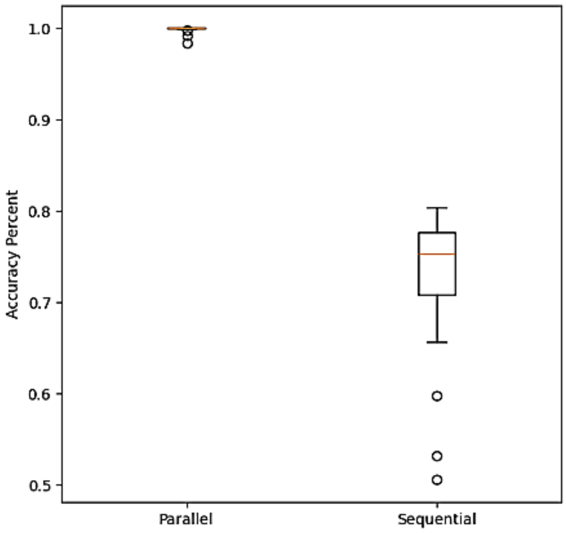


Figure 4. Comparison t-test of sequential and parallel configuration.

TABLE V
T-TEST SEQUENTIAL-PARALLEL

No	Model Configuration	Mean Accuracy	STD Accuracy
1	BiLSTM (sequential)	0.7328	0.06293
2	BiLSTM-GRU-RNN	0.9998	0.00285
	t-test value	26.42	
	p-value	0.000000005	

TABLE VI
T-TEST PARALLEL

No	Model Configuration	Mean Accuracy	STD Accuracy
1	BiLSTM-GRU	0.9998	0.00374
2	BiLSTM-GRU-RNN	0.9998	0.00285
	t-test	0.2249	
	p-value	0.8225	

GRU-RNN configuration. After that, the RNN layer was removed, and training was carried out. The final stage was training for the BiLSTM layer only. The configuration in this test used the 5-fold cross-validation method to ensure greater accuracy. The epoch value used was 100, the batch size was 32, the early callback parameter was set by monitoring val_loss with a patience value of 3 and restore_best_weights was true. The learning rate value used varied between 1 and 0.0000001. In the training process, each learning rate value used was processed using a 5-fold cross-validation. The results of the total training iterations carried out were processed to calculate the mean accuracy and standard deviation obtained.

As presented in Table VI, the ablation study method used to determine the effect of the neural network component on the trained model yielded t-value of 0.2249, indicating that there was no significant difference in the accuracy results of the models being compared. The p-value is above 0.05, indicating that the resulting accuracy data was not significantly different. Figure 5 shows that there is no significant difference in accuracy between the BiLSTM-GRU RNN and BiLSTM-GRU configurations. The accuracy did not increase significantly even though the ablation study principle was applied. However, the data showed that increasing the number of parallel layers reduced the variance of the accuracy data. The implementation of SMOTE on the training data to achieve a balance in the number of training data classes did not result in a significant change in accuracy; however, the implementation of SMOTE could reduce the variance of the training accuracy data (Figure 6).

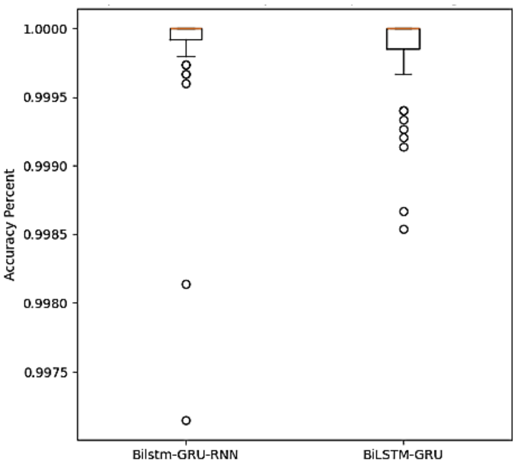


Figure 5. Comparison t-test BiLSTM-GRU-RNN vs BiLSTM-GRU.

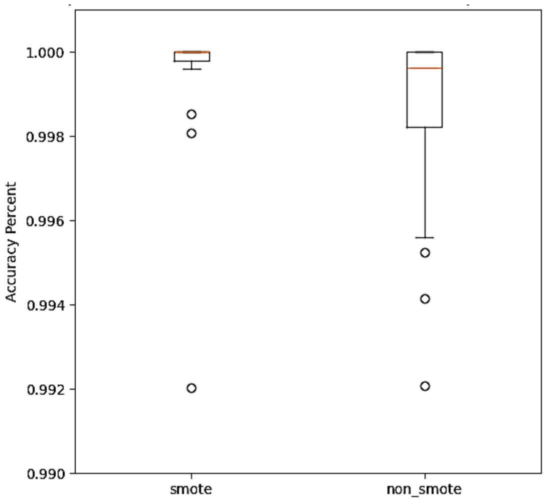


Figure 6. Comparison of t-test with SMOTE and without SMOTE.

TABLE VII
T-TEST SMOTE AND NON-SMOTE

No	Model Configuration	Mean Accuracy	STD Accuracy
1	SMOTE	0.99924	0.00285
2	Non-SMOTE	0.99813	0.00369
	t-test	1.4881	
	p-value	0.1407	

Table VII summarizes the results of the experiment to measure the effect of SMOTE on classification accuracy. The classification accuracy with the SMOTE implementation was slightly better than that without SMOTE. This was evidenced by the small t-test value of 1.4881 and the p-value greater than 0.05, namely 0.1407. This result is likely due to the use of k-fold cross-validation method during training, which helps overcome the effects of class imbalance in the dataset.

An additional experiment was conducted to determine the effect of the learning rate on classification accuracy. The BiLSTM-GRU-RNN algorithm was used in this experiment. One-way ANOVA was used to compare the accuracy statistics of each learning rate, ranging from 1 to 0.00000001. This experiment found that higher learning rates resulted in lower accuracy with greater data variance. As the learning rate value decreased, accuracy increased and variance decreased, as visualized in Figure 7.

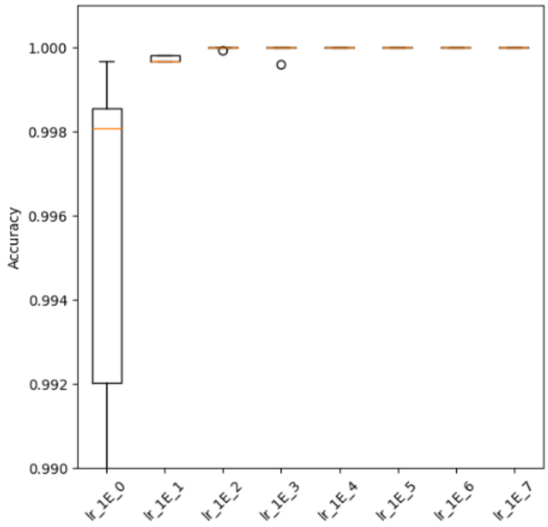
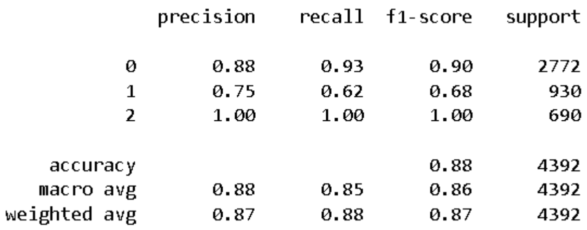


Figure 7. F-statistic learning rates comparison.



```
              precision    recall  f1-score   support

           0       0.88      0.93      0.90      2772
           1       0.75      0.62      0.68       930
           2       1.00      1.00      1.00       690

    accuracy                           0.88      4392
   macro avg       0.88      0.85      0.86      4392
weighted avg       0.87      0.88      0.87      4392
```

Figure 8. Classification report validation.

The experimental results for the BiLSTM-GRU-RNN configuration produced the highest accuracy, followed by validation testing. At this stage, accuracy validation was carried out using accuracy, precision, recall, F1 score, confusion matrix, and ROC curve metrics.

From the classification report visualized in Figure 8, the validation of the test data obtained an accuracy of 0.88. F1 score for the negative class (0) was 0.9, while for the neutral class (1) was 0.68, and for the positive class (2) was 1.00. These results showed that the classification for the positive and negative classes was well validated, while the neutral class had the smallest F1 score value.

The model validation results are also displayed in a confusion matrix in Figure 9. The negative prediction and the actual negative class had the largest value of 2,578. The positive class had 690 correct predictions, while neutral class had 577. The largest prediction error was found in 353 negative predictions but actually belonging to neutral class.

The validation process is also shown in the ROC curve of Figure 10. The true positive rate produced a curve above the random guess line limit approaching the number 1, so that it could be concluded that the accuracy of the resulting model had good performance.

Table VIII shows the comparison result from this research. The results obtained an accuracy of 88%, which outperformed the results of previous research. This followed by [26], which used the AdaBoost algorithm and achieved an accuracy of 84.5%. Research [24] used SVM algorithm and achieved an accuracy of 83.81%, while the use of BERT algorithm in [28] produced an accuracy of 83.00%.

When comparing sequential layer and parallel layer configurations, the parallel layer configuration outperformed all sequential layer. In the sequential layer configuration, the best accuracy was 80.20%, while in the parallel layer, the lowest accuracy was 80.80%, which was still superior to the best results in the sequential layer configuration.

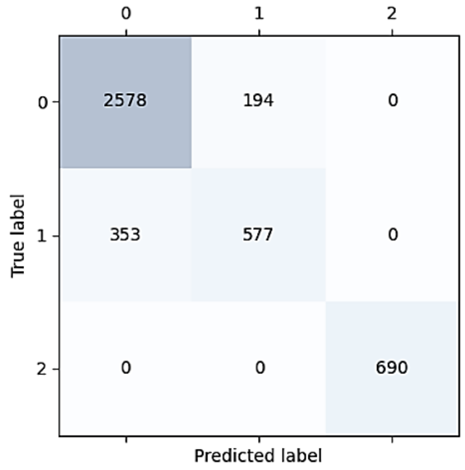


Figure 9. Confusion matric validation.

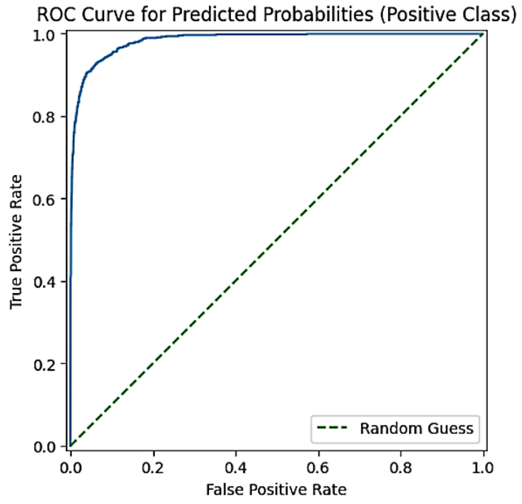


Figure 10. ROC curve validation.

TABLE VIII
COMPARISON OF RESEARCH RESULTS WITH PREVIOUS RESEARCH

No	Research	Algorithm	Accuracy
1	[28]	BERT	83.00%
2	[23]	Random forest classifier	81.35%
3	[24]	Support vector machine	83.81%
4	[25]	Linear regression + stochastic gradient descent classifier	79.40%
5	[26]	AdaBoost	84.50%
6	[27]	Naïve Bayes	76.10%
7	This research	3 Paralel layer GRU-BiLSTM-RNN	88.00%

The results of this study are in accordance with findings of prior research on ensemble learning, which reported that combining several machine learning algorithms or deep learning in parallel can improve performance. Prior research [21], [22], and [29] using the ensemble learning method further supports these results.

V. CONCLUSION

It can be concluded that the ensemble learning with a parallel layer configuration can increase accuracy, proven by a series of statistical tests of the accuracy of two groups of accuracy data from the training iteration process using the t-test method. For word embedding, Keras produced the highest accuracy among other word embedding methods used. The lowest accuracy was obtained by the Word2Vec self-train corpus dataset embedding. This suggests that the larger the word representation in a word embedding, the better the classification accuracy, as it has better ability to understand the semantic meaning of a sentence or word. The results of the three-parallel- layer configuration of GRU-BiLSTM-RNN with Keras embedding produced the best accuracy of 88% and outperformed the results of previous studies.

In the research process of the influence of neural network components using the ablation study method, it was found that the parallel network configuration increased classification accuracy. Reducing the number of parallel network configurations from 3 to 2 did not result in significant effect. Likewise, the implementation of SMOTE did not produce a significant effect because the training process had implemented k-fold cross validation, which overcame the weakness of dataset imbalance.

The experimental results of the learning rate value showed that the smaller the learning rate value, the better the accuracy, with a smaller variance value. The validation process of the training results with the classification report, confusion matrix, and ROC curve validated the accuracy of the training results, and the model had good performance.

Due to time constraints, this research has several shortcomings, including the combination of parallel networks, which can be tested with other algorithms. The comparison of SMOTE and non-SMOTE can be performed with or without k-fold cross-validation. The dataset used can also be used in future research using a larger dataset.

CONFLICTS OF INTEREST

The author hereby declares with full awareness and without coercion from any party that there is no conflict of interest in the research and preparation of this paper.

AUTHORS' CONTRIBUTIONS

Conceptualization, Agung Nugroho; data collection, Agung Nugroho; methodology, Agung Nugroho and Arief Setyanto; data extraction, Agung Nugroho; visualization, Agung Nugroho; software, Agung Nugroho and Arief Setyanto; analysis and interpretation of results, Agung Nugroho and Arief Setyanto; funding acquisition, Agung Nugroho; drafting of the paper, Agung Nugroho and Arief Setyanto.

ACKNOWLEDGMENT

Thank you to Amikom University Yogyakarta for supporting this research.

REFERENCES

[1] Y.C. Lee *et al.*, "An empirical research on customer satisfaction study: A consideration of different levels of performance," *SpringerPlus*, vol. 5, Sep. 2016, doi: 10.1186/s40064-016-3208-z.

[2] X. Zheng, M. Lee, and C.M.K. Cheung, "Examining e-loyalty towards online shopping platforms: The role of coupon proneness and value consciousness," *Internet Research*, vol. 27, no. 3, pp. 709–726, Jun. 2017, doi: 10.1108/IntR-01-2016-0002.

[3] J.Ł. Wilk-Jakubowski, "Analysis of broadband informatics services provided via the Internet, and the number of internet users on a global scale," unpublished.

[4] S. Zhang, Z. Wei, Y. Wang, and T. Liao, "Sentiment analysis of Chinese micro-blog text based on extended sentiment dictionary," *Future Gener. Comput. Syst.*, vol. 81, pp. 395–403, Apr. 2018, doi: 10.1016/j.future.2017.09.048.

[5] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[6] J. Pennington, R. Socher, and C.D. Manning, "GloVe: Global vectors for word representation," in *Proc. 2014 Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, 2014, pp. 1532–1543, doi: 10.3115/v1/d14-1162.

[7] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," 2016, *arXiv:1607.04606*.

[8] N. Ketkar, "Introduction to Keras," in *Deep Learning with Python*, Berkeley, CA, USA: Apress, 2017, pp. 97–111.

[9] J. Devlin *et al.*, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." [Online]. Available: https://github.com/tensorflow/tensor2tensor

[10] Y. Liu *et al*., "RoBERTa: A Robustly optimized BERT pretraining approach," 2019, arXiv:1907.11692.

[11] S. Haque, Z. Eberhart, A. Bansal, and C. McMillan, "Semantic similarity metrics for evaluating source code summarization," *in ICPC '22, Proc. 30th IEEE/ACM Int. Conf. Program Compr.*, 2022, pp. 36–47, doi: 10.1145/3524610.352790.

[12] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Phys. D, Nonlinear Phenom.*, vol. 404, pp. 1–28, Mar. 2020, doi: 10.1016/j.physd.2019.132306.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[14] S. Ghosh, A. Ekbal, and P. Bhattacharyya, "Chapter 2 - Natural language processing and sentiment analysis: perspectives from computational intelligence," in *Computational Intelligence Applications for Text and Sentiment Data Analysis*, D. Das, A. K. Kolya, A. Basu, and S. Sarkar, Eds., Cambridge, CA, USA: Academic Press, 2023, pp. 17–47.

[15] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[16] Y.A. Singgalen, "Sentiment analysis and trend mapping of hotel reviews using LSTM and GRU," *J. Inf. Syst. Inform.*, vol. 6, no. 4, pp. 2814–2836, Dec. 2024, doi: 10.51519/journalisi.v6i4.926.

[17] U.B. Mahadevaswamy and P. Swathi, "Sentiment analysis using bidirectional LSTM network," in *Procedia Comput. Sci.*, 2022, pp. 45–56, doi: 10.1016/j.procs.2022.12.400.

[18] Aakash, S. Gupta, and A. Noliya, "URL-based sentiment analysis of product reviews using LSTM and GRU," in *Procedia Comput. Sci.*, 2024, pp. 1814–1823, doi: 10.1016/j.procs.2024.04.172.

[19] S. Kumawat, I. Yadav, N. Pahal, and D. Goel, "Sentiment analysis using language models: A study," in *2021 11th Int. Conf. Cloud Comput. Data Sci. Eng. (Conflu.)*, 2021, pp. 984–988, doi: 10.1109/Confluence51648.2021.9377043.

[20] R. Wadawadagi and V. Pagi, "Sentiment analysis with deep neural networks: Comparative study and performance assessment*," Artif. Intell. Rev.*, vol. 53, pp. 6155–6195, Dec. 2020, doi: 10.1007/s10462-020-09845-2.

[21] J. Kazmaier and J.H. van Vuuren, "The power of ensemble learning in sentiment analysis," *Expert Syst. Appl.*, vol. 187, pp. 1–16, Jan, 2022, doi: 10.1016/j.eswa.2021.115819.

[22] A.F. Jadama, B. Jobarteh, M.M. Islam, and M.K. Toray, "Ensemble learning: Methods, techniques, application," unpublished.

[23] M. Vadivukarassi, N. Puviarasan, and P. Aruna, "An exploration of airline sentimental tweets with different classification model*," Int. J. Res. Eng. Appl. Manag. (IJREAM)*, vol. 4, no. 2, pp. 72–77, May 2018, doi: 10.18231/2454-9150.2018.0124.

[24] A.I. Saad, "Opinion mining on US airline Twitter data using machine learning techniques," in *2020 16th Int. Comput. Eng. Conf. (ICENCO), 2020*, pp. 59–63, doi: 10.1109/ICENCO49778.2020.9357390.

[25] F. Rustam *et al*., "Tweets classification on the base of sentiments for US airline companies," *Entropy*, vol. 21, no. 11, pp. 1–22, Nov. 2019, doi: 10.3390/e21111078.

[26] A. Rane and A. Kumar, "Sentiment classification system of Twitter data for US airline service analysis," in *2018 IEEE 42nd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, 2018, pp. 769–773, doi: 10.1109/COMPSAC.2018.00114.

[27] A. Alshamsi, R. Bayari, and S. Salloum, "Sentiment analysis in English texts," *Adv. Sci. Technol. Eng. Syst. J.*, vol. 5, no. 6, pp. 1683–1689, Dec. 2020, doi: 10.25046/AJ0506200.

[28] A. Patel, P. Oza, and S. Agrawal, "Sentiment analysis of customer feedback and reviews for airline services using language representation model," *Procedia Comput. Sci.*, vol. 218, pp. 2459–2467, doi: 10.1016/j.procs.2023.01.221.

[29] W. Guoyin and S. Hongbao, "Parallel neural network architectures," in *Proc. ICNN'95 - Int. Conf. Neural Netw.*, 1995, pp. 1234–1239, doi: 10.1109/ICNN.1995.487331.