Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 23, No. 1, November 2023, pp. 53∼66
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 200/M/KPT/2020
DOI: 10.30812/matrik.v23i1.3084

Hyperparamaters Fine Tuning for Bidirectional Long Short Term Memory on Food Delivery

Rahman , Teguh Iman Hermanto , Meriska Defriani
Sekolah Tinggi Teknologi Wastukancana, Purwakarta, Indonesia

Article Info

Article history:

Received June 19, 2023
Revised October 17, 2023
Accepted November 08, 2023

Keywords:

Long Short-Term Memory
Fine-tuning
Food Delivery
Hyperparameters
Sentiment Analysis

ABSTRACT

Every food delivery order carries out major promotions to attract users' attention. Users are only asked to rate drivers and restaurants on the platform, not services, which means the company doesn't get feedback. Many users discuss this service on Twitter social media, because these reviews are not on the platform, this data must be reprocessed so that it can be used as input. This research aims to understand user sentiment, and maximize model accuracy. Sentiment analysis can be used to determine user sentiment based on reviews, and the results of this analysis can provide suggestions for companies. This research method uses a two-way short-term memory model, because there will be many ambiguous sentences and different from other models, this model reads sentences in two directions. Gofood and Shopeefood research results have an accuracy of 98.1%, and Grabfood's is 97.4%. Gofood is indeed the most popular compared to other food delivery services. Bidirectional Short Term Memory Model, Word2Vec feature extraction, balanced dataset, hyperparameters and fine tuning improve accuracy well.

Copyright ©2022 The Authors.
This is an open access article under the CC BY-SA license.

Corresponding Author:

Rahman, +6285864358364,
Faculty of Engineering and Informatic Engineering,
Sekolah Tinggi Teknologi Waskutakancana Purwakarta, Indonesia,
Email: rahmanrahman89@wastukancana.ac.id

How to Cite:
R. Rahman, T. Hermanto, and M. Defriani, "Hyperparamaters Fine Tuning for Bidirectional Long Short Term Memory on Food Delivery", *MATRIK: Jurnal Managemen,Teknik Informatika, dan Rekayasa Komputer*, Vol. 23, No. 1, pp. 53-66, Nov, 2023.
This is an open access article under the CC BY-SA license (https://creativecommons.org/licenses/by-sa/4.0/)

1. INTRODUCTION

As technology develops, the community's needs will be more and more diverse. Users will give feedback to startups to fulfill their needs and will always evolve. Some startups start from shuttle services such as Gojek and Grab to have many features, including the food delivery provided by Gojek, Gofood, and Grab, Grabfood. Even E-commerce like Shopee has a Shopeefood feature, which differs greatly from the two applications. One of the digital marketing strategies used by caterers is food delivery. Food delivery connects consumers with caterers online. The users of this service are also predominantly young [1]. Sentiment analysis is processing text data categorized into positive and negative sentiments [2]. Bidirectional Long Short Term Memory (BiLSTM) can capture information without ignoring the sentence's meaning because BiLSTM is a bidirectional LSTM [3]. If transfer learning uses knowledge in solving problems, fine-tuning optimizes knowledge by changing parameters to get better results [4]. Every user completes an order, and the user will rate it. Still, the user only rates the driver and the restaurant, even though every food delivery provides many promotions and events. Although food delivery providers always have a lot of promotions, users still pay attention to several things, such as shipping rates, admin rates, and services, because the platform does not ask users to review its services; not a few users discuss this service review on social media and Twitter. Since this review data is not on the platform, the company cannot directly manage it. Sentiment analysis can help the sentiment given by users about this service; each review can be an input for the company so that this feature continues to grow and improve. Research also discusses this service, but on the shuttle service, which we also use as follows: users of the Grab shuttle service have varying levels of satisfaction with the service provided. There are still complaints and suggestions from users [5]. Based on this research, it was found that the same problem is that users often give either positive or negative reviews, so sentiment analysis is needed to find out the sentiment given by users.

The accuracy of the comment-based sentiment analysis method increases through research comparisons with other traditional sentiment analysis methods. However, the comment-based sentiment analysis method on BiLSTM takes much time. Some experiments include epochs, learning rate, max length, text length, NodeNum to be used as hyperparameters, and comparison with other word vectors. The results of the proposed method are compared with several other models. The accuracy obtained is 91.54% [6]. The model BiLSTM algorithm has an accuracy above 90% and can better classify and analyze the emotional tendencies of sports event users. The results show that the improved model can perform better in analyzing complex emotional characteristics of irregular text and the application of Danmaku. The proposed model is compared with other models regarding recall, accuracy, and F1-score, with all datasets applied under the same conditions [7]. The results show the effectiveness of BiLSTM for the performance of sequential model data. Compared to other basic models, this shows that in almost all cases, our deep learning model is more effective and efficient regarding classification quality. The proposed model is measured against other models, and results are compared using a stemmer and not a stemmer [8]. A two-stage prediction optimization approach on the Arabic Cyberbullying Corpus (ArcCybC) was obtained from Twitter. The first stage uses a pre-training word embedding model for fine-tuning. The second stage uses eXtreme Gradient Boosting (XGBoost) and Support Vector Machine (SVM). This research uses three-word embeddings: AraVec Unigram, AraVec N-gram, and GloVe. The model training process with fine-tuning includes many epochs of ArcCyBc training to increase the number of vocabularies. GA helps to reduce the time and cost of testing, which involves manual trial and error. Making the optimal hyperparameters achieved with maximum prediction capability despite the small dataset indicates the system's effectiveness. [9]. The research results show that the proposed model can improve classification performance and increase the effectiveness of the ability to learn and understand the semantics of the model. For datasets from hotel reviews, it gets an accuracy of 85.66%, and for emotional analysis, it is 76.78% compared to LSTM. This research has a hyperparameter of AB-LaBSE [10]. The experimental results show that the information-based topics proposed by BiLSTM are effective compared to other traditional models for sentiment classification with an accuracy of 85.02% [11].

From some of the related work above, there are many ways to get maximum model performance. Starting with how to process the dataset, such as stemmer, different types of word embedding, hyperparameters in the model, a little modification of the model, and fine-tuning for the model. Then, with the many ways that have been mentioned, we will test it for word embedding selection, dataset conditions when balanced, model hyperparameters, and fine-tuning hyperparameters. This trial will measure success by model accuracy, validation, and comparison with other model performances. Although several trials show good results, there are still some drawbacks, such as computational inefficiency, and each trial must be done manually one by one. Here, we add tests to compare balanced and imbalanced data, perform hyperparameter optimization, and compare the optimal results on other models with the same optimal parameter conditions. There is also research that resamples so that the dataset becomes balanced by using resamples on ADASYN, SMOTE, and SMOTE-ENN. It has been found that SMOTE has the most balanced data [12], whereas here, the researcher is undersampling to trim the most data to the least.

This experiment is conducted to obtain the most optimal accuracy by evaluating so that the model obtained is not close to overfitting to achieve good model performance. Measuring the success of some word embedding in providing judgment on each word, and measuring the success of the model in understanding the context of user sentiment. As well as measuring the success of

optimization and hyperparameter testing from multiple tests. There are still many ways that can be used for hyperparameters, such as in model structures or even on the structure of the word embedding used. This research is also aimed at users so they can choose this service more, and companies so that reviews are more focused. This article is structured with several sections. the next step from this stage is section 2 to carry out the stages in this article. the test results from the stages of section 2 are analyzed in sections 3 results and analysis. the results that have been analyzed in section 3 are made conclusions and accommodated in section 4 conclusions

2. RESEARCH METHOD

This type of research is quantitative because the data used is text that will be presented into vector values in the extraction features later, and this data can also be analyzed. Several stages in the methodology include data acquisition, including scrapping Twitter data, tweet data resulting from scrapping, cleaning data, and labeling. Next, there is text preprocessing, transformation, and filtering. Then, there is feature extraction, tokenization, and Word2Vec weighting. After that, classification using bidirectional is used as Figure 1.

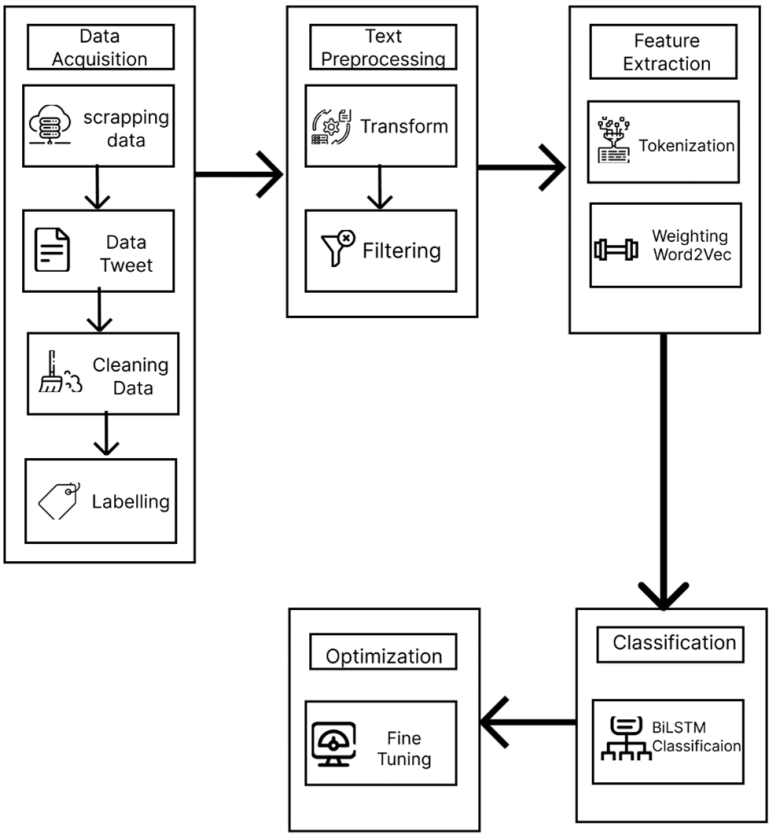

Figure 1. Research method

2.1. Data acquisition

Some platforms do not provide APIs for crawling, and users are not asked to rate the service. However, many users discuss this service on Twitter. We use scrapping to retrieve review data widely discussed by users on the Twitter social network. The library used is Snscrape from Python. The search queries used are ”Gofood,” ”Grabfood,” and ”Shopeefood”. Each search gets 3000 tweets, with a period of February 22, 2023, to 3 months back, and with comment filters and tweets containing links, so Gofood gets 3000 Tweets until February 3, 2023, Grabfood as many 2800 Tweets until February 1, 2023, and shopeefood 3000 Tweets until December 1, 2022. The dataset is saved in CSV format and semicolon separator because if using a comma separator, the Tweet content also

contains a comma symbol. Labeling is done manually because many sentences are ambiguous and off-topic. Sentiment labels are divided into three parts: negative, neutral, and positive. The results of data acquisition can be seen in Table 1.

Table 1. Scrapping

dataset	Tweet
Gofood	udah makan tapi masi laper, saatnya apa? saatnya gofood mcd
Grabfood	SEBLAK GUE KAPAN DATENG WOY GUE PENASARAN SOALNYA GUE DULY NEMU GRABFOOD SEBLAK ENAK TP GUE LUPA NAMANYA APAAN
Shopeefood	NGESELIN BGT PAGI PAGI ORDER SHOPEEFOOD GABISA CANCEL DRIVERNYA GABISA DIHUBUNGIN SEJAM

2.2. Text Preprocessing

Text preprocessing is the processing of text for further weighting in feature extraction. The text is transformed to make the text more uniform, as weighting involves calculating the frequency of occurrence of words, which affects the weight value of the word. This stage produces clean words and reduces words that are out of context. There are several stages of transformation, including case folding, changing the form of sentences, doing a lowercase on sentences so that all sentences become the same small, removing emoticons, removing symbols and punctuation marks, removing URLs, removing numbers, removing hashtags, and whitespace. Normalization and stemming transform abbreviated sentences into normal words, while stemming transforms sentences into basic words. All transformation stages are carried out so that the sentences become uniform and the weighting becomes the same because many abbreviations and terms must be standardized to have the same value. Filtering removes unnecessary words such as conjunctions and other out-of-context words unrelated to the topic. The filter here is focused on removing harsh words because both positive and negative sentiments often contain these words.

2.3. Feature Extraction

Data that has been processed will be given weight in $word2vec$, but it must be tokenized first by breaking the sentence into words. $Word2vec$ is a word embedding unsupervised learning neural network training in the form of a weight matrix. Semantic learning is influenced by surrounding words by relying on local information from the language. $Word2vec$ has two algorithms: a continuous bag of words (CBOW) and Skipgram. CBOW uses context in predicting the target word, while Skipgram predicts the target context using a word [13]. The preprocessed words are weighted or extracted into vector values using $Word2Vec$. Using the Gensim library, $Word2Vec$ can search for vector values of words and find the most similar or similar words. The algorithm used in this research is skip-gram because it is more suitable for small data. The structure of the skip-gram can be seen in the following figure 2.

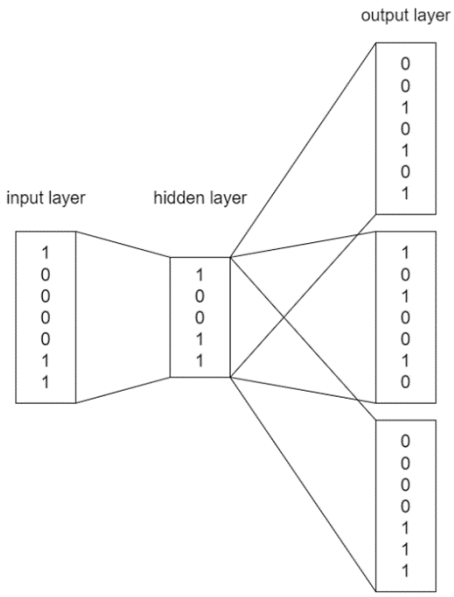

Figure 2. Skipgram

Several word embeddings will be tested in this research. Word2vec and FastText still have the same structure because they are just different vendors. Then, there is $doc2vec$, which does not have a skipgram algorithm. Finally, there is tf-idf, which is very different from word embedding.

2.4. Classification

Recurrent Neural Network (RNN) is an artificial neural network whose process is called repeatedly with sequential data. RNN is deep learning because it has many layers [14].LSTM complements the shortcomings of RNN in predicting past words over a long time period. LSTM has the selective ability to recall or delete information [15]. LSTM has a large memory and is suitable for data sequences [16]. LSTM has several gates, including a forget gate to sort out the information needed as in Equation (1), input gate to combine previous output and new input with candidate cell state in Equations (2) and (3), and cell state to place old data input in update gate with Equation (4), and output gate with hidden state in Equations (5) and (6) [17].

$$f_t = \sigma(W_f.[h_{t-1}, x_t]b_f) \tag{1}$$

Where :
ft : Forget Gate
σ : Activation Function Sigmoid
W_f : input gate Weight t
h_{t-1} : Hidden state on timestep before
x_t : Timestep values in t
b_t : Forget gate Bias

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{2}$$

Where :
i_t : Input Gate
σ : Activation Function Sigmoid
W_t : input gate Weight t
h_{t-1} : Hidden state on timestep before
x_t : Timestep values in t
b_i : input gate Bias

$$\tilde{c}_t = tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{3}$$

Where :
$\tilde{c}_t$: Candidate cell state
$tanh$: Activation Function hyperbolic tangent
W_c : candidate cell state Weight
h_{t-1} : Hidden state on timestep before
x_t : Timestep values in t
b_c : Bias Candidate cell state gate

$$C_t = f_t * C_{t-1} + i_t * \tilde{c}_t \tag{4}$$

Where :
C_t = Cell state
f_t = Forget gate
C_{t-1} = Cell state on timestep before
i_t = Input gate
$\tilde{c}_t$ = Candidate cell state

$$O_t = \sigma(W_O.[h_{t-1}, x_t] + b_O) \tag{5}$$

Where:
O_t = Output Gate
= Activation Function sigmoid
W_O = output gate Weight
$h_(t-1)$ = Hidden state on timestep before
x_t = timestamp values in t
b_O = Output gate Bias

$$h_t = O_t * tanh(C_t) \tag{6}$$

Where :
h_t = Hidden state on timestep t
O_t = Output gate
tanh = Activation Fuhyperbolic tangent
C_t = Cell state

Bidirectional Long Short-Term Memory is a development of the LSTM model where two layers process in opposite directions. This model is very good for recognizing sentence patterns because each word in the document is processed sequentially, and reviews can be understood if each word is learned sequentially. The lower layer moves forward, understanding and processing from the first word to the last, while the upper layer moves backward, understanding and processing from the last word to the first. With these two opposing layers, the model can understand and take perspective from the previous word and the leading word, deepening the learning process and allowing the model to better understand the context of the review [18].

The data set is divided into training data, test data, and validation data, which are used to measure the quality of the model. Training, test, and val data each have their x and y; especially for trainY, testY, and valY, a hot encoding has to be done first. The split data is subjected to a token embedding process to convert text into vectors to be used as tokens in the dictionary. The token to the embedding process is used as input to the embedding initializer in the input layer. The embedding in the initializer contains the results of the word embedding, namely Word2Vec. After that, text-to-sequence converts text tokens that refer to the dictionary. Padding converts the sequence into a 2D Numpy array. The padded sequence is used as a train in the model. The processed data will be classified using Bidirectional Long Short-Term Memory (BiLSTM). The structure of Bidirectional Long short-term memory (BiLSTM) can be seen in Figure 3, and the Equation for the output of Bidirectional Long short-term memory (BiLSTM) can be denoted as in Equation (7) where h1 as LSTM Forward, and hr as LSTM Reverse.

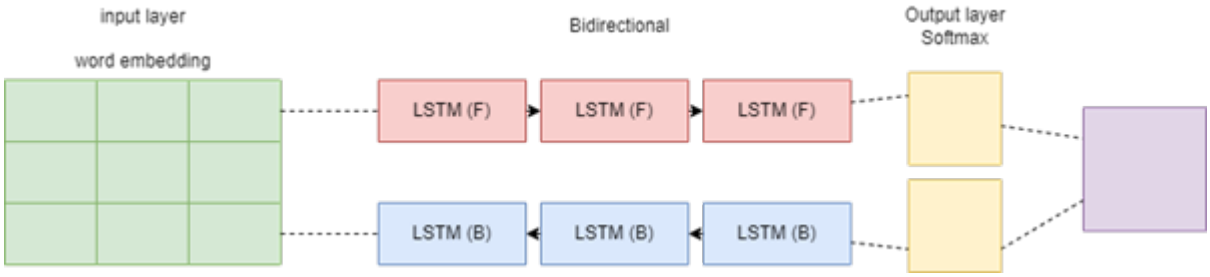

Figure 3. BiLSTM architecture[3]

$$H_t = h_1 \oplus h_r \tag{7}$$

At the classification stage, we perform hyperparameter tests on epoch and batch size to find the most suitable settings for the model and the pattern obtained from the results. Not only is accuracy a measure of model success, but we also pay attention to the validation obtained so that the model does not approach overfitting.

2.5. Optimization

A model trained to perform a task is then tuned, and a modified model is taken by fine-tuning to perform a similar task [19]. It is not always a trained model; fine-tuning can also train a pre-trained model. Transfer learning has two stages, namely pre-training and fine-tuning. Pre-training uses the base model and does not perform optimization to speed up the training process; however, fine-tuning performs training on the model produced by pre-training [20]. Parameters are input variables to the model, while hyperparameters are variables that can affect the model's output. Hyperparameters are not changed during optimization, which means they are independent of the data [21]. One of the optimization techniques is hyperparameter tuning to get the expected output results, one of which is accuracy [22]. The optimization carried out is hyperparameters for batch size and epoch in fine-tuning and proves the best model.

3. RESULTS AND ANALYSIS

3.1. Data Acquisition

Labeling is done manually because many sentences are ambiguous and off-topic. Sentiment labels are divided into three parts: negative, neutral, and positive. Neutral data is not used because neutral sentences already have positive and negative meanings. Next, the data is cleaned to remove duplicate data and empty data. After cleaning and labeling each dataset from the initial 3000 tweets, Gofood became 1276 tweets, Grabfood became 1189 tweets, and Shopeefood became 1003 tweets. Gofood has a positive sentiment of 710 and a negative sentiment of 556. Grabfood has a sentiment of 624 positive and 565 negative. Shopeefood has a positive sentiment of 334 and a negative of 669, so the data set used is shown in Figure 4, and the result of labeling is shown in Table 2.

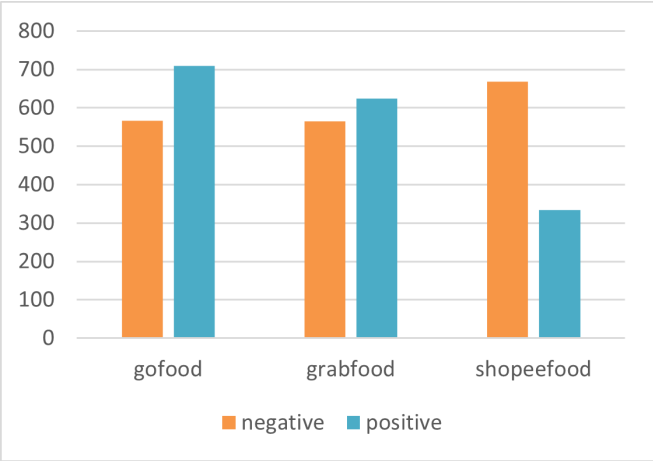

Figure 4. Distribution of dataset

Table 2. Labelling Result

Dataset	Tweet	Label
Gofood	udah makan tapi masi laper, saatnya apa? saatnya gofood mcd	Positive
Grabfood	SEBLAK GUE KAPAN DATENG WOY GUE PENASARAN SOALNYA GUE DULY NEMU GRABFOOD SEBLAK ENAK TP GUE LUPA NAMANYA APAAN	Negative
Shopeefood	NGESELIN BGT PAGI PAGI ORDER SHOPEEFOOD GABISA CANCEL DRIVERNYA GABISA DIHUBUNGIN SEJAM	Negative

3.2. Text Preprocessing

there are several stages in text preprocessing, including transformation, filtering, and labelling. in the transformation itself there is case folding to change the form of sentences, perform lower case on sentences so that all sentences become the same small, remove emoticons, remove symbols and punctuation, remove URLs, remove numbers, remove hashtags, and whitespace. still in the transformation there is word normalisation to convert abbreviated words into normal words. the Table 3 are the results of case folding and normalisation.

Table 3. Case Folding and Normalization

Before	Case Folding	Normalization
Ada voc voucher gofood 70% ga ya? Yg minbel 10/15k tpi Laper bgt hiks # zonauang	ada voc voucher gofood ga ya yg minbel k tpi laper bgt hiks	ada voucher voucher gofood tidak ya yang minimal beli ribu tapi lapar banget hiks

The next stage is still in transformation, there is stemming to change words into basic words. The transformation is complete then there is filtering to remove words in the text that we don't need. Following are the results of stemming and filtering in table 4.

Table 4. Stemming and Filtering

before	Stemming	Filtering
hidup ini bukan berlomba melawan teman melainkan berlomba melawan hujan biar sempet pesen gofood dulu	hidup ini bukan lomba lawan teman lain lomba lawan hujan biar sempet sen gofood dulu	hidup berlomba melawan teman berlomba melawan hujan biar sempet pesan gofood

3.3. Feature Extraction

Before weighting is done on Word2Vec, the text that has been preprocessed must be made in the form of tokens, or each sentence in the dataset must be broken down in the form of words. In the regex library, the split technique and the results of tokenization can be seen in Table 5.

Table 5. Tokenization

Before	Tokenization
hidup ini bukan lomba lawan teman lain lomba lawan hujan biar sempet sen gofood dulu	[hidup, berlomba, melawan, teman, berlomba, melawan, hujan, biar, sempet, pesan, gofood]

The parameters used on Word2Vec include min count to ignore words with low frequency, windows size, which is the distance between the current word and prediction, epoch number of repetitions, using skip-gram algorithm, vector dimension, and negative sampling to pay attention to target value 1, and value 0 to ignore the target. The parameters used can be seen in the Table 6 below.

Table 6. Word2vec Parameters

Parameter	amount
Min_count	1
Window	5
Epochs	100
SG	1
Vector_size	50
Negative	1

We have used a few experiments on the epochs and vector size parameters. For vector size and small epochs, it is suitable for the Long Short Term Memory (LSTM) model, while epochs and vector size, as in the table above, are suitable for Bidirectional LSTM (BiLSTM). The various embeddings we have tested include Word2Vec, FastText, Doc2Vec, and Tf Idf. All word embeddings have the same parameters, except that Doc2Vec does not use skip-gram parameters, and Tf Idf does not have the same parameters as other embeddings.

Initial test conditions were performed on an unbalanced Gofood dataset, 64 LSTM layers, 0.2 drop out, and Adam optimizer without using fine-tuning. Word embedding test results are in Table 7.

Table 7. Word Embedding

Embedding	accuracy	validation
Word2Vec	**94.6%**	**75%**
FastText	92.9%	71.4%
Doc2Vec	93.2%	72.7%
Tf Idf	71.2%	66.8%

Word2Vec gives the highest accuracy and score, so this model uses Word2Vec embedding. Having obtained the most optimal word embedding, since the above study uses unbalanced data, we conducted another study comparing accuracy and score on balanced and unbalanced data. This trial was carried out because the data with the most labels would make the accuracy results skewed greater where the data with the most labels had more words to train. so here we try to compare balanced and imbalanced data. To get balanced data, the researchers cut the dataset at the highest label, because if we increase the dataset with the lowest label, we have to do data acquisition again. So we used undersampling to reduce the data to the lowest number of labels. Here we use the sklearn resampling library. This method does not cut the data in order but in a random way. so the results obtained are as in the table 8.

Table 8. Balanced dataset

dataset	accuracy	validation
balanced	**96.4%**	**79.8%**
Imbalance	94.6%	75%

The balanced dataset showed better performance on the model than the initial condition of the imbalanced dataset both in accuracy and evaluation, which means that balanced data has better training and testing data for training the model. So, we use the balanced dataset to continue in other trials.

3.4. Classification

The dataset is divided into training, testing, and validation. The training is used for training data, while testing and validation measure the model's success. The division is 20% test data, 25% validation data, and the rest of the training data. The model still uses the initial parameters with Wod2Vec embedding and balanced data. Before training data for the model, there are two parameters that we made hyperparameters, namely epoch and batch size. Epoch is the repetition of model training, while the batch size is the number of data samples in an iteration. This experiment was carried out to obtain the model's most optimal parameters and observe the results obtained. Some tests that researchers do are in Table 9 Not only paying attention to the results of the train model, the evaluation results must also not be further from the results of the train model so that the model obtained does not become overfitting and the results are as in table 10.

Table 9. Train Model

Batch size epoch	32	64	128
25	91%	91%	87.2%
50	96.4%	94.1%	93.6%
75	97.4%	96.9%	95.6%

Table 10. Validation model

Batch size epoch	32	64	128
25	80.8%	81.8%	78.2%
50	78.2%	79.8%	77.7%
75	81.8%	79.8%	77.2%

Based on the results of the accuracy of the train, it can be seen that the more epochs against the batch size, the more accuracy will increase. We use batch size 32 and epoch 75 because it has the highest evaluation value. This shows that the hyperparameter successfully improves the model's performance and gets significant results. Although the results obtained were significant, at epoch 75, with an increased batch size, the accuracy actually decreased, as shown in figure 5.

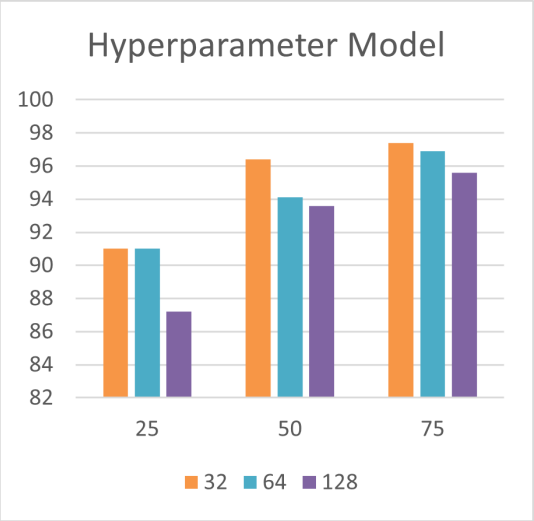

Figure 5. Hyperparameter Model

3.5. Optimization

Models that have been trained will be retrained by fine-tuning. Fine-tuning optimization performs the same training for models with the same parameters and additional learning rate parameters. The parameters made in this test are epoch and batch size with a

learning rate of 5e-5. The test results are presented in the fine-tuning train in Table 11, Such as when hyperparameterizing a model. We also carried out trials on validation for fine tuning optimization and the results of validation can be seen in table 12.

Table 11. Train Fine Tuning

Batch size epoch	32	64	128
10	97.8%	97.9%	97.8%
25	97.8%	96.8%	98.1%
50	97.3%	**98.4%**	98.4%

Table 12. Validation Fine Tuning

Batch size epoch	32	64	128
10	81.3%	82.8%	79.2%
25	81.3%	80.3%	79.8%
50	82.3%	**82.3%**	79.2%

In contrast to the model, testing accuracy and evaluation fine-tuning do not show significant results, so epoch 50 and batch size 64 have good accuracy values and the highest evaluation. Previously, the results of the hyperparameter model have shown high accuracy and evaluation values; with fine tuning and hyperparameters, these results can still be improved in accuracy and evaluation.

By getting the most optimal parameters from some of the tests above, we conducted a comparison with other types of Recurrent Neural Networks with Bidirectional Long Short-Term Memory (BiLSTM) performance, namely Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU), but before fine-tuning. The following are the test results of several models in Table 13.

Table 13. Accuracy Model

	Accuracy	F1 Score	Recall
BiLSTM	**97.4%**	**78%**	**77%**
LSTM	96.9%	78%	78.5%
GRU	94.7%	76%	76.5%

BiLSTM has the best accuracy of LSTM and GRU, indicating BiLSTM is better able to understand the context in this case study. GRU is a development of RNN. The gate owned by GRU is also not like LSTM. LSTM reads a context from one direction only. BiLSTM, the development of LSTM, makes it capable of reading two-way contexts simultaneously. These results show that BiLSTM is better than some of these models. The parameters used were applied to other datasets, and the accuracy of Gofood and Shopeefood was 98.1%, and Grabfood was 97.4%, as shown in Figure 6.

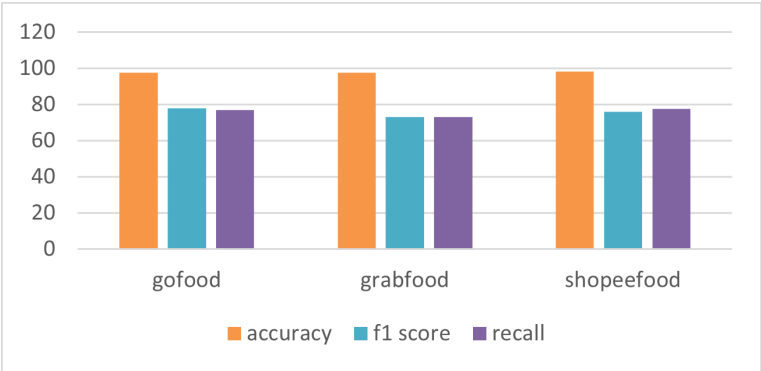

Figure 6. Other dataset results

based on several tests that have been carried out, Word2Vec obtained an accuracy of 94.6%, balanced data of 96.4%, hyperparameter model of 97.4%, and hyperparameter fine tuning of 98.1%. the accuracy value obtained is very significant as shown in the figure 7.

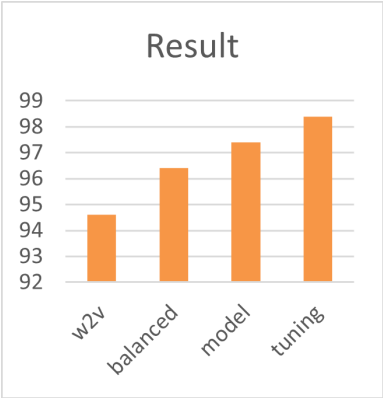

Figure 7. Accuracy Result

some researchers use Bidirectional Long Short Term Memory (BiLSTM) to test on several datasets, some also carry out optimization and hyperparameters at several stages. The following is the result of the author's work with previous research based on the highest accuracy value as an shown in table 14.

Table 14. Results and Previous Research

	Model	Optimization
Our Work	**97.4%**	**98.1%**
Guixian Xu[6]	91.5%	-
Shaokang wang[7]	90.2%	-
Elfaik[8]	92.6%	-
Fatima shannaq[9]	88.1%	88.2%
Yijie Pei[10]	-	85.6%
Yanming Huang[11]	95%	-

4. CONCLUSION

Every trial we conducted had good results, Word2Vec has better results than some other embeddings. Data balancing shows that the performance of the model can be improved. Hyperparameters run on the model, the more epochs against the batch size will give significant results. Hyperparameters in the fine-tuning also make the results optimal, although not significant. Bidirectional Long Short Term Memory (BiLSTM). in this case study understands the context of sentiment analysis better than Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU). In this study, more data is needed to avoid overfitting and the comparisons in the tests have more differences. This research is also still inefficient in the computational process, and each result in the trial is carried out one by one manually Based on user reviews, Gofood and Grabfood have more positive sentiments, in contrast to Shopeefood. So that Gofood and Shopeefood have an accuracy of 98.1%, and Grabfood of 97,4%. Based on the amount of data, Gofood is more popular than the others and tends to provide more positive sentiment. Suggestions for future research are to make more datasets so that the expected results are not too slightly different when compared with other methods or when applied to other datasets. there are still many other tests that can be done such as on the model structure on the number of LSTM layers, dropouts to avoid overfitting the model, changing the input size, and many others

REFERENCES

[1] P. Rahima and R. Rismayati, "Pengaplikasian Platform Food Delivery Service Shopee Food dalam Memasarkan Produk Minuman Kamsia Boba Mataram," *Bakti Sekawan : Jurnal Pengabdian Masyarakat*, vol. 2, no. 1, pp. 42–47, 2022.

[2] A. Rahman, I. #1, H. Sulistiani, B. Miftaq, H. #3, A. Nurkholis, and S. #5, "Analisis Perbandingan Algoritma LSTM dan Naive Bayes untuk Analisis Sentimen," *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. 8, no. 2, pp. 299–303, 2022.

[3] K. S. Nugroho, I. Akbar, A. N. Suksmawati, and I. Istiadi, "Deteksi Depresi dan Kecemasan Pengguna Twitter," *The 4th Conference on Innovation and Application of Science and Technology (CIASTECH 2021)*, no. Ciastech, pp. 287–296, 2021.

[4] M. Afif, A. Fawwaz, K. N. Ramadhani, and F. Sthevanie, "Klasifikasi Ras pada Kucing menggunakan Algoritma Convolutional Neural Network(CNN)," *Jurnal Tugas Akhir Fakultas Informatika*, vol. 8, no. 1, pp. 715–730, 2020.

[5] D. R. Alghifari, M. Edi, and L. Firmansyah, "Implementasi Bidirectional LSTM untuk Analisis Sentimen Terhadap Layanan Grab Indonesia," *Jurnal Manajemen Informatika (JAMIKA)*, vol. 12, no. 2, pp. 89–99, 2022.

[6] G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, "Sentiment analysis of comment texts based on BiLSTM," *IEEE Access*, vol. 7, pp. 51 522–51 532, 2019.

[7] S. Wang, Y. Chen, H. Ming, H. Huang, L. Mi, and Z. Shi, "Improved Danmaku Emotion Analysis and Its Application Based on Bi-LSTM Model," *IEEE Access*, vol. 8, pp. 114 123–114 134, 2020.

[8] H. Elfaik and E. H. Nfaoui, "Deep Bidirectional LSTM Network Learning-Based Sentiment Analysis for Arabic Text," *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 395–412, 2021.

[9] F. Shannaq, B. Hammo, H. Faris, and P. A. Castillo-Valdivieso, "Offensive Language Detection in Arabic Social Networks Using Evolutionary-Based Classifiers Learned From Fine-Tuned Embeddings," *IEEE Access*, vol. 10, no. June, pp. 75 018–75 039, 2022.

[10] Y. Pei, S. Chen, Z. Ke, W. Silamu, and Q. Guo, "AB-LaBSE: Uyghur Sentiment Analysis via the Pre-Training Model with BiLSTM," *Applied Sciences (Switzerland)*, vol. 12, no. 3, 2022.

[11] Y. Huang, Y. Jiang, T. Hasan, Q. Jiang, and C. Li, "Topic BiLSTM model for sentiment classification," *ACM International Conference Proceeding Series*, vol. Part F1376, pp. 143–147, 2018.

[12] C. Kaope and Y. Pristyanto, "The Effect of Class Imbalance Handling on Datasets Toward Classification Algorithm Performance," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 22, no. 2, pp. 227–238, 2023.

[13] A. Nurdin, B. Anggo Seno Aji, A. Bustamin, and Z. Abidin, "Perbandingan Kinerja Word Embedding Word2Vec, Glove, Dan Fasttext Pada Klasifikasi Teks," *Jurnal Tekno Kompak*, vol. 14, no. 2, p. 74, 2020.

[14] Y. Karyadi, "Prediksi Kualitas Udara Dengan Metoda LSTM, Bidirectional LSTM, dan GRU," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 9, no. 1, pp. 671–684, 2022.

[15] F. Hidayat, "Implementasi Klasifikasi Gambar Untuk Industri Pakaian Menggunakan Image Search Engine Berbasis Website," vol. 10, no. 1, pp. 356–362, 2023.

[16] A. S. Talita and A. Wiguna, "Implementasi Algoritma Long Short-Term Memory (LSTM) Untuk Mendeteksi Ujaran Kebencian (Hate Speech) Pada Kasus Pilpres 2019," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 19, no. 1, pp. 37–44, 2019.

[17] D. Aisyah, T. W. Purboyo, and M. Kallista, "Prediksi Penderita Tuberkulosis Dengan Algoritma Long Short-Term Memory Prediction of Tuberculosis Using Long Short-Term Memory (Lstm) Algorithm," vol. 10, no. 1, pp. 742–749, 2023.

[18] D. Junggu and M. Pasaribu, "Peningkatan Akurasi Klasifikasi Sentimen Ulasan Makanan Amazon dengan Bidirectional LSTM dan Bert Embedding," pp. 9–20.

[19] F. Hafifah, S. Rahman, and S. Asih, "Klasifikasi Jenis Kendaraan Pada Jalan Raya Menggunakan Metode Convolutional Neural Networks (CNN)," *TIN: Terapan Informatika Nusantara*, vol. 2, no. 5, pp. 292–301, 2021.

[20] E. I. Haksoro and A. Setiawan, "Pengenalan Jamur Yang Dapat Dikonsumsi Menggunakan Metode Transfer Learning Pada Convolutional Neural Network," *Jurnal ELTIKOM*, vol. 5, no. 2, pp. 81–91, 2021.

[21] M. I. Gunawan, D. Sugiarto, and I. Mardianto, "Peningkatan Kinerja Akurasi Prediksi Penyakit Diabetes Mellitus Menggunakan Metode Grid Seacrh pada Algoritma Logistic Regression," *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, vol. 6, no. 3, p. 280, 2020.

[22] I. G. T. Isa and B. Junedi, "Hyperparameter Tuning Epoch dalam Meningkatkan Akurasi Data Latih dan Data Validasi pada Citra Pengendara," *Prosiding Sains Nasional dan Teknologi*, vol. 12, no. 1, p. 231, 2022.

[This page intentionally left blank.]