ISSN 2355-0082

Beyond Traditional Methods: The Power of Bi-LSTM in Transforming Customer Review Sentiment Analysis

Casey Tjiptadjaja[1] and Moeljono Widjaja[2]
[1,2]Department of Informatics, Universitas Multimedia Nusantara, Tangerang 15810, Indonesia
[1]casey.tjiptadjaja@student.umn.ac.id, [2]moeljono.widjaja@umn.ac.id

Accepted 11 September 2024
Approved 14 March 2025

***Abstract*— In the current generation, many large and small companies compete fiercely to create things better than those on the market, such as smartphones, TVs, and many other things. One way they can do this is by guaranteeing the quality of services or goods that are better than others. The provider must investigate the feedback of their users or customers to improve the quality of service of the goods or services offered. Most medium and small companies, such as Micro, Small, and Medium enterprises (MSMEs), online stores, and so on, conduct research on customer feedback manually by looking at one-by-one feedback from customers, which is very ineffective and inefficient if a lot of customer feedback is obtained. Therefore, this research is conducted with the intention and purpose of helping medium and small companies analyze their customer sentiment, as well as trends over a certain period. This research will apply the Bidirectional Long Short-Term Memory (Bi-LSTM) algorithm to perform sentiment analysis on customer feedback. This research also compares other deep learning methods with the proposed method, namely the Uni-LSTM, GRU, CNN, and Simple-RNN algorithms. After testing, the accuracy results of the Uni-LSTM, Bi-LSTM, GRU, CNN, and Simple-RNN algorithms are 52.2%, 92.4%, 52.2%, 90.9%, and 49.7%, respectively. Then it was found that the Bi-LSTM algorithm with 64 units, five (5) training epochs, and a training and testing ratio of 80:20, as well as a model that implements 'relu' activation (rectified linear unit), obtained the maximum results, that is, got 92. 4% accuracy, which is the most superior accuracy compared to accuracy using other models and algorithms.**

***Index Terms*— Bi-LSTM; Customer Review; Deep Learning; Sentiment Analysis**

I. INTRODUCTION

In the 21st century, the urgency of community service is increasing. Society has progressed in various aspects of life, especially with the rapid development of technology and information. This development has opened access to information at all levels of society (information society), which makes it possible for people to compare different services between countries, companies, and individuals. Therefore, the demand for perfect service quality is inevitable. This must be balanced with better customer service because the company may integrate with the global society. In this context, it must be understood that society demands are growing, especially the demands for the quality of services provided by individuals or companies.

To improve the quality of service of the goods or services offered, the provider must research feedback from their users or customers. Most large companies already have a division that will research this customer feedback. For medium and small companies, such as Micro, Small, and Medium Enterprises (MSMEs), online stores (Instagram, Youtube, Tiktok, websites), most of them do this research manually by looking at customer feedback one by one. There are several reasons why this manual method is still used, such as the small number of customers and the cost factor. The manual method is still possible if they only have a small number of customers. However, if their customers are already in large numbers, such as hundreds and thousands, this method is no longer effective and efficient. Therefore, this research is done with the intention and purpose of helping medium and small companies analyze their customer sentiment and trends in a certain period, which in this research is in the context of months.

In the current era of globalization, transactions in the internet world are defined as e-Commerce [1]. In addition to promoting products that each seller owns, the e-Commerce department also handles reviews from E-Commerce users [2]. Over time, e-commerce users continue to grow and develop throughout the year, so the number of customer reviews is also increasing [3]. The development of information and communication technology makes it easier for humans to connect and share information through social networks [4]. Customers can write and post reviews about products and services on social media. Some of the most widely used social networks are Instagram, Twitter, TikTok, and YouTube.

Sentiment analysis, also known as opinion mining, is a technique used in natural language processing (NLP) to analyze and extract subjective information

from text, such as positive and negative emotions [5], [6]. This technique has gained widespread attention in research fields such as NLP and text analysis [7]. Not only among researchers, this technique is also widely used in businesses, governments, and organizations [8], especially to understand the public sentiment expressed in online comments and reviews. Sentiment analysis brings together various research fields, such as Natural Language Processing (NLP), data mining, and text mining, and is quickly becoming of great importance to organizations as they seek to integrate computational intelligence methods into their existing operations and seek to explain and improve their products and services [9]. There are different ways to perform sentiment analysis, ranging from rule-based methods that use a list of positive and negative terms as labeled data to train machine learning algorithms to build a classification [10].

Natural Language Processing (NLP) is a subset of artificial intelligence and linguistics that is devoted to helping computers understand statements or words written in human language. NLP was created to ease the user's work and satisfy the desire to communicate with computers in natural language [11]. In other words, Natural Language Processing aims to accommodate one or more specialization algorithms or systems. NLP assessment metrics in algorithmic systems enable the integration of language understanding and generation. Many things can be applied with NLP, such as detecting spam, becoming virtual agents or chatbots, and sentiment analysis [11].

Deep learning is a branch of machine learning [12]. The algorithm attempts to use high-level data abstraction using multiple processing layers consisting of complex structures or multiple non-linear transformations. Deep learning differs slightly from shallow learning in areas such as support vector machine and logistic regression. These shallow learning models have only one layer or no hidden layer nodes. Deep learning is based on multiple hidden layer nodes [13]. The core of deep learning is a multilayered network of nodes. Deep learning uses the input of the previous layer as the output of the next layer to learn highly abstract data features. So, machine learning algorithms (shallow learning) [14] are faster, but their performance or accuracy could be better. Meanwhile, deep learning methods perform better than the machine learning method for textual sentiment classification [15], [16].

Long-Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) that is robust and suitable for handling sequential data with long-term dependencies. LSTM is a type of RNN designed to overcome the missing gradient problem, which is a common problem (limitation) in RNNs [17]–[19]. LSTM has a unique architecture called memory cells that allows it to learn long-term dependencies in data sequences [20]. LSTM algorithm is one of the deep learning methods that can be applied in Natural Language Processing (NLP), including speech recognition, text translation, text summarization, and sentiment analysis. There are two types of LSTM, namely UniLSTM (unidirectional LSTM) and Bi-LSTM (bidirectional LSTM) [21].

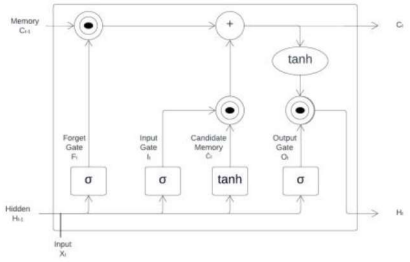

Fig. 1. Unidirectional LSTM Architecture
Source: Geeks For Geeks

Figure 1 shows the architecture of LSTM which consists of input gate, forget gate and output gate. Forget gate is where the information selection process occurs in the cell state using equation 1. Information will be discarded from the cell state if forget gate is 0, otherwise information will be stored in the cell state if forget gate is 1. Then at the input gate, information will pass through 2 layers, namely sigmoid and tanh. The cell state is generated from the output values of the two layers that have been combined. Then the process of updating the cell state value is carried out. Next, it will pass through the output gate, in which the output value of the cell state will be determined. The sigmoid layer is used to select the output based on the existing cell state, and will then be forwarded to the tanh layer.

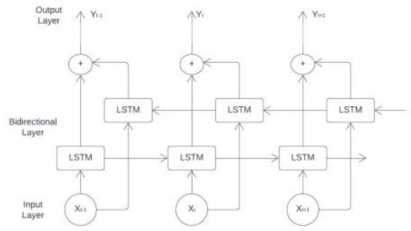

Fig. 2. Bidirectional LSTM Architecture
Source: Geeks For Geeks

The main difference between these two types of LSTM is that Uni-LSTM has the disadvantage [22] of only being able to process the input sequence in one direction (from beginning to end). In contrast, Bi-LSTM processes the input sequence in two directions (from beginning to end and from end to beginning) like shown in Figure 2 [23]. Uni-LSTM can only utilize information from past states. In contrast, Bi-LSTM can use information from past and future states to capture longterm dependencies and context in input sequences, making it suitable for tasks involving Natural Language Processing and data analysis [24]. The LSTM method is used in the research in [25]–[27] to

perform sentiment analysis and is one of the best algorithms to perform sentiment analysis compared to conventional methods (Support Vector, Na¨ıve Bayes, etc.).

Research in [25] conducted sentiment analysis comparing the Long Short-Term Memory (LSTM) and Na¨ıve Bayes (NB) algorithms, which found that LSTM has higher accuracy when compared to the NB algorithm; the LSTM algorithm has an accuracy rate of 72.85%, while the NB algorithm has an accuracy rate of 67.88%. In the study [26], a comparison of several algorithms was carried out to perform sentiment analysis, namely Simple Neural Network, Long Short-Term Memory, and Convolutional Neural Network, where the results show that the LSTM algorithm has an accuracy rate of 87%, which is the highest accuracy compared to the Simple Neural Network algorithm and CNN which has an algorithm of 81% and 82%. Gondhi [27] did a similar thing by comparing the LSTM, Na¨ıve Bayes, and Support Classification Algorithm algorithms to perform sentiment analysis, which the results show that the LSTM algorithm leads in terms of accuracy compared to the NB and SCA algorithms; the accuracy rate is 89%, 57.9%, and 67.7%, respectively.

Based on the problems described and some of the research presented, this research will use the Bi-LSTM method that uses Sigmoid activation, Relu hidden activation, Adam optimizer, and Binary cross-entropy loss function to perform sentiment analysis on customer reviews (feedback) using data on Tokopedia application review products from Mendeley Data [28]. This research also compares other deep learning methods with the proposed method, namely the Uni-LSTM, GRU, CNN, and Simple RNN algorithms. In conclusion, the contribution of this work presents a new algorithm (method) that significantly improves the accuracy of analyzing customer's sentiment and provides a comprehensive comparison of this method with other algorithm (method), highlighting its superior predictive capabilities. These advancements offer a more efficient approach for analyzing customer's sentiment, which could have wide applications in both MSMEs and companies.

II. METHODS

In this research, several stages are carried out to implement the Bi-LSTM algorithm for sentiment analysis of customer reviews (feedback). Figure 3 shows some of the steps taken in this research.

A. Data Collecting

The data used in this study are data obtained from the Mendeley Data website. The dataset was selected because it has balanced label statistics. Ideally, the data set should be balanced because a highly unbalanced dataset will make modeling difficult and lead to less precise accuracy. In the data used in this study, there are 42.75% of data with positive labels and 52.24% of data with negative labels. The total data used are 5400 data.

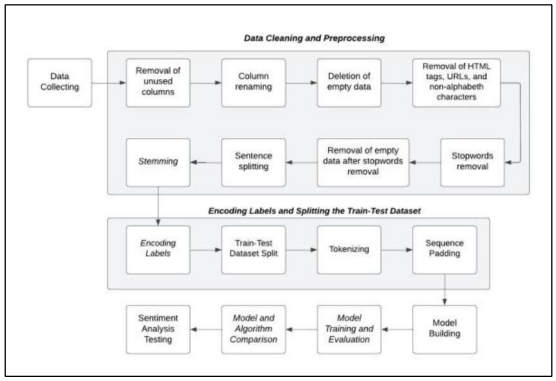

Fig. 3. The stages carried out in the research

B. Data Cleaning and Preprocessing

Data preprocessing is carried out at this stage, such as deleting some elements of the data set that have no value or will interfere with the analysis process. The following will explain in more detail the data cleaning and pre-processing process in this research, starting from the process at that stage until the results.

1) Deleting some unused columns using the drop function.
2) Rename the column using the rename function. The column previously "Customer Review" becomes "review," and the column "Sentiment" becomes "sentiment."
3) Deleting empty data using the dropna function. There is no empty data in the data used.
   Removal of HTML tags, URLs, and non-alphabetic characters. This is done with the help of functions from the Regex library. Table I shows the example of using the function that removes HTML tags, URLs, and nonalphabetic characters. In addition, all words are converted to lowercase using the 'lower' function. Table II shows an example of using the 'lower' function that converts all words to lowercase.
4) Remove stopwords from the corpus, such as 'which,' 'in,' and so on. This is done to not give special meaning to a sentence. This stage uses the library of an NLP, NLTK. Table III shows the process of removing stopwords.
5) After stopwords, there is a possibility that there are empty review data. Therefore, empty data are deleted after removing stopwords using the drop function.
6) Split the sentence into words using a function from the NLTK library, WhitespaceTokenizer(). This is done to facilitate the next stage of the stemming process. This is also used to see the ranking of words often mentioned. Table IV shows an example of breaking sentences into words using WhitespaceTokenizer().

TABLE I
THE RESULT OF REMOVING HTML TAGS, URLS, AND NON-ALPHABETIC CHARACTERS

Deletion Type	Before	After
HTML Tags	Penjual Ramah Melayani Pembeli Dengan Sabar dan Memberikan Berbagai Saran Yang Pembeli Tidak Tahu Mantap Lah	Penjual Ramah Melayani Pembeli Dengan Sabar dan Memberikan Berbagai Saran Yang Pembeli Tidak Tahu Mantap Lah
URL	mantap kipasnya kenceng https://www.tokopedia.com/, barangnya berkualitas sesuai sama harga	mantap kipasnya kenceng, barangnya berkualitas sesuai sama harga
Karakter yang bukan alfabet	Kualitas barang bagus, harga relatif murah, kiriman super cepat, response penjual baik, recommended seller, bintang 5, terima kasih yah, semoga SUKSES usaha-nya, aamiin ??????	Kualitas barang bagus harga relatif murah kiriman super cepat response penjual baik recommended seller bintang terima kasih yah semoga SUKSES usahanya aamiin

TABLE II
THE RESULT OF CONVERTING ALL WORDS INTO LOWERCASE LETTERS

Before	After
Penjual Ramah Melayani Pembeli Dengan Sabar dan Memberikan Berbagai Saran Yang Pembeli Tidak Tahu Mantap Lah	penjual ramah melayani pembeli dengan sabar dan memberikan berbagai saran yang pembeli tidak tahu mantap lah

TABLE III
RESULTS OF STOPWORDS REMOVAL

Before	After
barangnya sangat bagus	barangnya bagus
saya sangat kecewa dengan barang ini	kecewa barang

1) The stemming process is a technique that serves to retrieve the root of a word, commonly referred to as a lexicon. This is done to help reduce unnecessary computation in deciphering the whole word, where separate lemmas express most words' meaning well. This stage uses the library from Sastrawi.Stemmer.StemmerFactory. Table V shows an example of the stemming process.

TABLE IV
RESULT OF SENTENCE BREAKDOWN INTO WORDS

Before	After
barangnya sangat bagus	{barangnya}, {sangat}, {bagus}
saya sangat kecewa dengan barang ini	{saya}, {sangat}, {kecewa}, {dengan}, {barang}, {ini}

TABLE V
THE RESULT OF STEMMING PROCESS

Before	After
barang bagus berfungsi seler ramah pengiriman cepat	barang bagus fungsi seler ramah kirim cepat
barang mengecewakan	barang kecewa

C. *Encoding Labels and Splitting the Train-Test Dataset*

After data cleaning and preprocessing, the next stage is the encoding of labels and the division of the train-test dataset. Then, it will continue with the tokenizing process and sequence padding. Encoding is a method of converting one value into another, mostly from strings to numbers. This opportunity stage is done by converting the labels 'positive' and 'negative' into the numbers '1' and '0'. This is done with the LabelEncoder() function from the sklearn.preprocessing library.

After that, the data set is divided into train and test parts of 80% and 20% [29]. This is done using the train_test_split function from the sklearn.model_selection library. After dividing the train and the test, tokenizing and sequence padding is performed.

There are several parameters used in this research, namely vocab_size, oov_tok, embedding_dim, max_length, padding_type, and trunc_type. The parameter vocab_size is a parameter that specifies the number of dictionaries to be created. It is based on the total unique words when tokenized. The parameter oov_tok, or what stands for Out-of-Vocabulary, represents words that are not found in the vocabulary and are replaced during tokenization. The embedding_dim parameter represents the dimension of the embedding space of the word. It determines the size of the vector representation for words. The parameter max_length specifies the maximum length of a sentence. The padding_type parameter is a parameter that determines the padding in a sentence; in this study, the 'post' padding type is used, where padding is added after the end of the sentence. However, the trunc_type parameter reduces the number of words in a sentence if the sentence exceeds the number of max_length parameters. In this research, the *trunc* type 'post' is

used, where the word to be truncated is at the end of the sentence.

After these parameters are determined, the word is tokenized and converted into a sequence of integers, which uses the parameters vocab_size and oov_tok. This tokenizer builds a vocabulary based on the frequency of words in the training data ('training_sentences'). After that, the text_to_sequences function is used, which converts the sentences in the training and testing data into a sequence of integers using a tokenizer. Then, the pad_sequences function is also used, adding padding to adjust each sentence's length in the training and testing data.

1) Tokenizing: Tokenizing is a method of dividing a sentence into words and creating a dictionary of all the unique words found, and all the words are also assigned unique integers. Each sentence is converted into an array of integers representing all the words in it. In this research, the tokenizer API from the Keras library is used. Table VI shows an example of the tokenizing process.

TABLE VI
TOKENIZING PROCESS

Review	Tokenizing Result
barangnya bagus	kamus[0]: {barangnya}, kamus[1]: {bagus}
kecewa barang	kamus[2]: {kecewa}, kamus[3]: {barang}

2) Sequence Padding: At this stage, each array representing each sentence in the data set is filled with zeros from the end of the sentence to make the array size 200 and make all sentences the same length. Table VII shows an example of the sequence padding process with a maximum of 200 arrays and the 'post' method.

TABLE VII
SEQUENCE PADDING PROCESS

Review	Sequence Padding Result
barangnya bagus	[{barangnya}, {bagus}, {0}, {0}, ..., {0}]
kecewa barang	[{kecewa}, {barang}, {0}, {0}, ..., {0}]

D. Model Building

At this stage, like shown in Figure 4, the model is created with the Keras library and uses the Bi-LSTM method, which will output the probability of a positive sentence if the label is '1'. This model will be compiled using binary crossentropy loss and Adam's optimizer because the data contain binary classification. Adam's optimizer uses stochastic gradient descent to train the deep learning model and compare the probabilities of each predicted label class (0 or 1). Accuracy is used as the primary metric. Figure 5 shows the code snippet of the model implemented in this study. The created model uses the algorithm *Bidirectional* LSTM algorithm with 64 units.

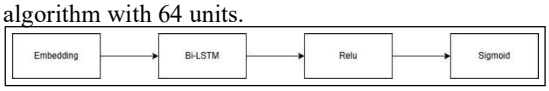

Fig. 4. Model Building

```
#INISIALISASI MODEL
model = keras.Sequential([
    keras.layers.Embedding(vocab_size, embedding_dim, input_length
    =max_length),
    keras.layers.Bidirectional(keras.layers.LSTM(64)),
    keras.layers.Dense(24, activation='relu'),
    keras.layers.Dense(1, activation='sigmoid')
])
#KOMPILASI MODEL
model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
#BENTUKAN MODEL
model.summary()
```

Fig. 5. Model Building Code

E. Model Training and Evaluation

The model in this study was trained using five (5) epochs. After that, the model is evaluated by calculating its accuracy. The accuracy is calculated by dividing the number of correct predictions by the total number of predictions.

F. Model and Algorithm Comparison

In this study, several models are tried, both models that use training and testing data with a division of 70 and 30, models that use epochs of six (6), ten (10), and fifteen (15), models that use LSTM units of 32 and 128, and models that do not use relu activation (Rectified Linear Unit). This study also compares the Bi-LSTM algorithm with the Uni-LSTM, GRU, CNN, and Simple-RNN algorithms, most of which are algorithms used for comparison in previous studies, namely research in [25]–[27].

G. Sentiment analysis Testing

This test uses the same model as the one previously created with Bi-LSTM. Here, users can input strings in the form of sentences and can input files as well. The output for users who input sentences is the sentiment prediction result ('Positive' or 'Negative'), while for users who input files, the output will be in the form of a diagram that illustrates the percentage of total sentiment predictions that are 'Positive' and 'Negative,' as well as sentiment analysis trends in a certain period, namely per month. Users who input files will receive the prediction result file and the date. The test data used for string input is manual input, while the test data used for file input is restaurant review data from Kaggle which is manually dated.

III. RESULTS AND DISCUSSION

After several data cleaning and preprocessing stages, the results will be used and entered into the model made in this study.

A. Data Results Before and After Data Cleaning and Preprocessing

Data must be cleaned and preprocessed before being entered into the model that has been created to eliminate irrelevant data that can interfere with the training and testing process.

B. Model Implementation

The model implemented in this study uses the Bidirectional LSTM algorithm with 64 units. After various experiments, both with other algorithms such as Unidirectional LSTM, GRU, and CNN and with the number of units of 32 and 128, it was found that even the Bidirectional LSTM algorithm with 64 units produced the best accuracy.

Table VIII shows the accuracy results of the Uni-LSTM, BiLSTM, GRU, CNN, and Simple-RNN algorithms are 52.2%, 92.4%, 52.2%, 90.9%, and 49.7% respectively. Table IX shows the accuracy results of using Bi-LSTM with LSTM units of 32 units, 64 units, and 128 units are 90.7%, 92.4%, and 91.1%, respectively.

It can be seen that the model used starts from the embedding layer first, then the Bi-LSTM layer, and then the dense layer.

TABLE VIII
COMPARISON RESULTS OF ALGORITHMS

Algorithms	Accuracy Percentage
Uni-LSTM	52.2%
Bi-LSTM	92.4%
GRU	52.2%
CNN	90.9%
Simple-RNN	49.7%

TABLE IX
LSTM UNIT COMPARISON RESULT

Total LSTM Units	Accuracy Percentage
32 unit	90.7%
64 unit	92.4%
128 unit	91.1%

The embedding layer is used to convert integer indices into fixed-size dense vectors. Table X shows the process performed in the embedding layer. After that, the input goes into the BiLSTM layer, which processes data from both directions.

TABLE X
100-DIMENSIONAL LAYER EMBEDDING PROCESS

Review	Embedding result
barangnya bagus	{barangnya}: {01001.....1010}, {bagus}: {10101.....1001}
kecewa barang	{kecewa}: {11010.....0101}, {barang}: {00101.....0110}

Then, two dense layers are used in this research; the first is 'relu' (Rectified Linear Unit), and the second is 'sigmoid.' 'Relu' introduces non-linearity to the model, thus allowing the model to learn complex mappings between input and output features. This minimizes loss and helps to effectively propagate the gradient through the network, allowing for efficient training. The 'sigmoid' serves to output 0 or 1 (binary class). In this research, 'relu' activation is used because a comparison of models that use 'relu' and not is made. Table XI shows the accuracy comparison results of models that use 'relu' activation and not. The model that uses 'relu' has an accuracy rate of 92.4%, while the model that does not use 'relu' has an accuracy rate of 90.4%.

TABLE XI
COMPARISON RESULTS FOR THE USE OF 'RELU' ACTIVATION

'relu' Activation	Accuracy Percentage
Using 'relu' activation	92.4%
Not using 'relu' activation	90.4%

The loss function used in this study is 'binary_crossentropy,' a general loss function used for binary classification problems. Then, the 'Adam' optimizer updates the neural network weights during training. Adam is an adaptive optimization algorithm that performs well in various deep-learning tasks. Then, the evaluation metric used in this research is 'accuracy,' which is used to monitor the model's performance during training. Finally, the model architecture is notified. This is done to make it easier to check the model structure and help debug and optimize the model architecture.

C. Training and Testing Process

As explained earlier, in this study, the training and testing data division was carried out with a ratio of 80:20. This is because after experimenting with the 70:30 ratio, the accuracy results still need to be improved compared to the 80:20 ratio. Table XII shows the accuracy results for models with a training and testing ratio of 70:30 and 80:20. The accuracy of the model using a training and testing ratio of 80:20 is 92.4%, while the accuracy of the model using a training and testing ratio of 70:30 is 89.9%.

TABLE XII
COMPARISON RESULTS OF THE USE OF TRAINING AND TESTING RATIOS

Training dan Testing Ratio	Accuracy Percentage
Training 70% dan testing 30%	89.9%
Training 80% dan testing 20%	92.4%

D. Training Process

After the data is implemented into the model, the data will be trained first. In this research, training is done with an epoch of five (5). This was determined after experimenting with epochs six (6), ten (10), and

fifteen (15), the results of which are shown in Table XIII. The accuracy results obtained with epochs five (5), six (6), ten (10), and fifteen (15) are 92.4%, 89.9%, 90.1%, and 89.9%, respectively. Figure 6 and Figure 7 show the result plots of loss and accuracy values from training and validation. The orange line shows the result of data accuracy from the 'Validation' variable, which is the data used for validation. In contrast, the blue line shows the result of data accuracy from the 'Training' variable, which is the data used for training.

TABLE XIII
DIFFERENT EPOCH COMPARISON RESULTS

Total epoch	Accuracy Percentage
5 *epoch*	92.4%
6 *epoch*	89.9%
10 *epoch*	90.1%
15 *epoch*	89.9%

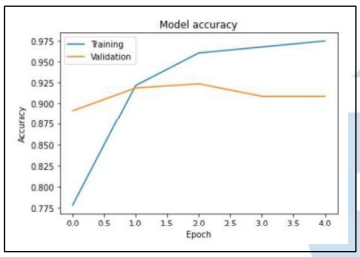

Fig. 6. Plot of training and validation accuracy results

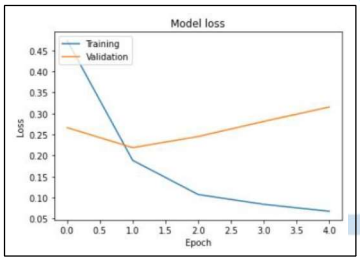

Fig. 7. Plot of training and validation loss values

E. Testing Process

After completing the training data, testing data will be carried out using the previously trained model. This is to determine the actual accuracy results, which with a training and testing ratio of 80:20, LSTM units of 64 units, and epochs of 5 produce very optimal accuracy results, which are 92.4%, this can be seen in Figure 8. Figure 8 also shows that the results of sentiment predictions more than 0.5 will go to the 'Positive' label, while prediction results less than 0.5 will go to the 'Negative' label.

```
43/43 [==============================] - 5s 82ms/step
Accuracy of prediction on test set :  0.9244444444444444
```

Fig. 8. Testing data accuracy results

In Figure 9, the results of the classification of confusion metrics in this study can also be seen. In Figure 9, we can see the results of the 'precision,' 'recall,' 'f1-score', and 'support' values. These values are the results obtained from the model that has been trained and *tested*. Figure 10 shows the results of the confusion metrics obtained in this study. The following section will explain the calculation formula for the data. Please note that TP (True Positive) is data that is in the lower right confusion matrix (positive, positive), FP (False Positive) is data that is in the upper right (positive, negative), FN (False Negative) is data that is in the lower left (negative, positive), and TN (True Negative) is data that is in the upper left (negative, negative). From the result shown, notice that the numbers of FN and FP are still quite large, which is a loss 10% after classification. This could be because many factors, such as insufficient training data, there is some of the data that uses another language, or some data have some typos that make the data invalid.

```
from sklearn import metrics
print(metrics.classification_report(test_labels,pred_labels))
```

	precision	recall	f1-score	support
0	0.95	0.91	0.93	705
1	0.90	0.94	0.92	645
accuracy			0.92	1350
macro avg	0.92	0.93	0.92	1350
weighted avg	0.93	0.92	0.92	1350

Fig. 9. Results from confusion metrics classification

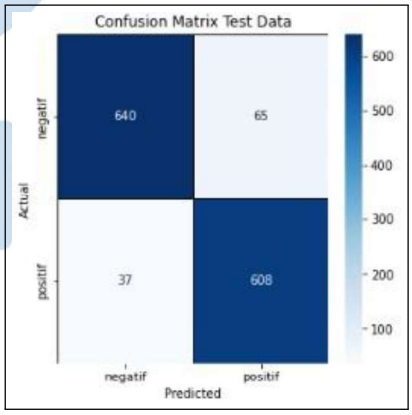

Fig. 10. Confusion metrics result

F. Usage

In this study, there are several types of use for the models that have been previously trained. There are sentiment analysis with manual sentence input, sentiment analysis with files without dates, and sentiment analysis and monthly sentiment trends with files that have dates. To get review files from customers, users can visit pages or download software

on the internet that provides data scrapping services, such as https://id.exportcomments.com, https://scrapy.org, and https: //webscrapper.io. After that, users can upload the data to platforms such as Jupyter Notebook, Google Colab, Kaggle, and others.

1) Manual sentence input: Here, the user can do sentiment analysis by inputting as many sentences as possible into the 'sentence' array to find the sentiment of these sentences. Figure 11 shows the test results performed by manually entering sentences. It can be seen that the results obtained are very accurate with the customer reviews, where positive reviews are predicted to have positive labels, and negative reviews are predicted to have negative labels.

```
1/1 [==============================] - 0s 161ms/step
Alhamdulillah barang sudah sampai, bisa dipakai semoga berfungsi dengan baik dan awet, kurirnya baik
smoga rezekinya lancar terus
Predicted sentiment :  Positive
Sangat kecewa. Baru 4 bulan scroll sudah rusak.
Predicted sentiment :  Negative
seller luar biasa.. packing baik, pengiriman cepat, penting nya barang sesuai deskripsi dan berfungs
i. terima kasih..
Predicted sentiment :  Positive
Penjualnya Ramah Melayani Pembeli Dengan Sabar dan Memberikan Berbagai Saran Yang Pembeli Tidak Tahu
<br /><br /> ... Mantap Lah ??????<br /><br />
Predicted sentiment :  Positive
Barangnya sangat bagus
Predicted sentiment :  Positive
Saya sangat kecewa dengan barang ini
Predicted sentiment :  Negative
```

Fig. 11. Results from testing by manually inputting sentences

2) Files without date: Here are the steps that users must take to do sentiment analysis using files without a date:

1) Perform data extraction using the previously mentioned pages or software regarding customer reviews on platforms such as Instagram, TikTok, etc.
2) After the data is obtained, the user can upload it to the platform used.
3) After a successful upload, the user can change the file name in the code according to the file name that was previously uploaded.
4) Then, if desired, the user can change the column index you want to remove or is unnecessary in this sentiment analysis. This can be done with the 'drop' function. In this research and testing, the columns used contain customer reviews and columns with sentiment labels only.
5) To make it easier for the user, the user can change the column's name containing the review data to 'review.' This can be done with the 'rename' function. In this test, we will change the name of the data column to be used, from the previous 'Customer Review' to 'review' and from the previous 'Sentiment' to 'sentiment'.
6) After that, the user only needs to run the entire customized code. The code has been modified to predict sentiment from customer reviews based on files uploaded by users before. The steps taken are the same as those taken in this research previously, mentioned in Chapter 3, from pre-processing data to tokenizing and padding data.
7) The user can see the sentiment results in a file called 'output.csv.' On the result, the pre-processed customer review data is entered into the column named 'Lemmatized Sentence' and the results of sentiment analysis (prediction) into the column called 'Predicted Sentiment.' The data is entered into a file with the format 'csv'.
8) Users can also see the positive and negative data ratio from the sentiment results obtained. Figure 12 shows the ratio of data before being processed into the model (the user cannot see this since the data uploaded by the user does not have sentiment labels). In contrast, Figure 13 shows the data ratio after input into the model. It can be seen in both figures that the difference in the label ratio is very little, which is about 1%.

```
Average length of each review :  16.095
Percentage of reviews with positive sentiment is 47.75925925925926%
Percentage of reviews with negative sentiment is 52.24074074074074%
```

Fig. 12. Before being entered into the model

```
Average length of each review :  11.273703703703704
Percentage of reviews with positive sentiment is 48.962962962962962%
Percentage of reviews with negative sentiment is 51.03703703703704%
```

Fig. 13. After being entered into the model

9) Users can also see the frequently mentioned words in each positive and negative label. Figure 14 shows 10 (ten) frequently mentioned words in the 'Positive' label and the total number of times they were mentioned. In contrast, Figure 15 shows 10 (ten) frequently mentioned words in the 'Negative' label and the total number of times they were mentioned.

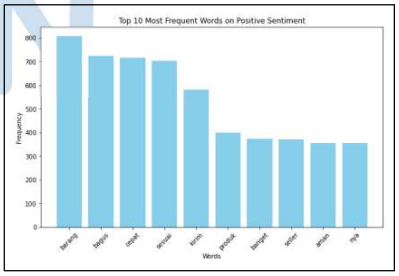

Fig. 14. Top 10 words on 'Positive' label

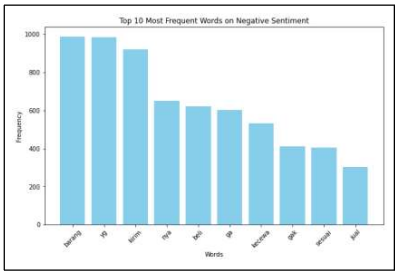

Fig. 15. Top 10 words on 'Negative' label

3) Files with date: Here are the steps that users must do to do sentiment analysis using files with dates:

1) Follow steps 1 - 6 from the usage steps of the file without a date.
2) The user can see the results of sentiment in the file 'output_date.csv' as shown in Figure 16. In the code in Figure 16, the pre-processed customer review data is entered into the column named 'Lemmatized_Sentence' and the results of sentiment analysis (prediction) into the column called 'Predicted Sentiment.' The data is entered into a file with the format 'csv'.

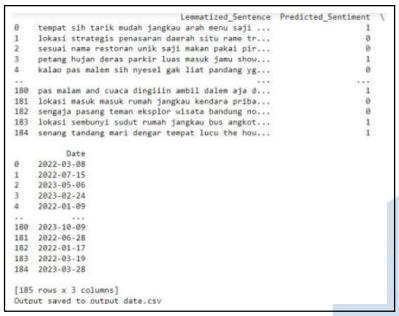

Fig. 16. The result of the processed data

3) Users can see the monthly trend results from the sentiment results obtained previously. Figure 17 shows the results of the trends obtained by the user. The orange plot illustrates the number of customer reviews with the label 'Negative,' while the blue plot illustrates the number of customer reviews with the label 'Positive.'

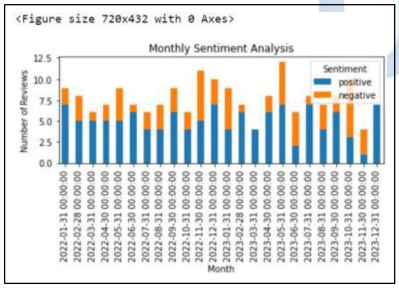

Fig. 17. Results of monthly sentiment trends

4) Users can also see the positive and negative data ratio from the sentiment results obtained. Figure 18 shows the data ratio before being processed into the model (the user cannot see this since the data uploaded by the user does not have sentiment labels). In contrast, Figure 19 shows the data ratio after inputting into the model. It can be seen in both figures that the difference in the label ratio is very little, which is about 1%.

```
Average length of each review :  38.78378378378378
Percentage of reviews with positive sentiment is 64.86486486486487%
Percentage of reviews with negative sentiment is 35.13513513513514%
```

Fig. 18. Before being entered into the model

```
Average length of each review :  22.335135135135136
Percentage of reviews with positive sentiment is 63.78378378378379%
Percentage of reviews with negative sentiment is 36.21621621621622%
```

Fig. 19. After being entered into the model

5) Users can also see the frequently mentioned words in each positive and negative label. Figure 20 shows 10 (ten) frequently mentioned words in the 'Positive' label and the total number of times they were mentioned. In contrast, Figure 21 shows 10 (ten) frequently mentioned words in the 'Negative' label and the total number of times they were mentioned.

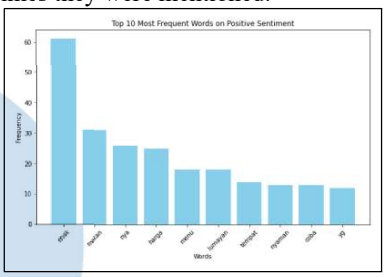

Fig. 20. Top 10 words on the 'Positive' label

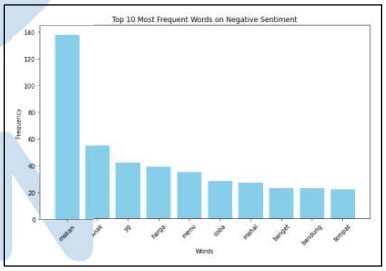

Fig. 21. Top 10 words on the 'Negative' label

IV. CONCLUSIONS

This research used 5400 customer review data samples to test sentiment analysis. This research implemented several deep learning algorithms, such as Bi-LSTM, Uni-LSTM, GRU, CNN, and Simple RNN. It was found that the BiLSTM algorithm with 64 units, five (5) training epochs, and a training and testing ratio of 80:20, as well as a model that implements 'relu' activation (Rectified Linear Unit), got the most maximum results, namely getting an accuracy of 92.4%, which is the most superior accuracy when compared to accuracy using other deep learning algorithms.

To obtain a better model performance, the user can add a stemming dictionary for unstandardized words and informal abbreviations, use a bilingual language

(Indonesian and English), use other algorithms than the algorithm that has been used in this research, because there are still many algorithms that are likely to get better model performance than the BiLSTM algorithm, and last but not least design (integrating) the website on the code or model that has been created so that testing for users can be more accessible, more effective, and efficient.

ACKNOWLEDGEMENTS

This research is fully funded by Universitas Multimedia Nusantara as part of the undergraduate thesis work.

REFERENCES

[1] K. Kasmi and A. N. Candra, "Penerapan e-commerce berbasis business to consumers untuk meningkatan penjualan produk makanan ringan khas pringsewu," *Jurnal AKTUAL*, vol. 15, no. 2, p. 109–116, Dec. 2017. [Online]. Available: http://dx.doi.org/10.47232/aktual.v15i2.27

[2] R. Nainggolan, F. Tobing, and E. Harianja, "Sentiment; clustering; k-means analysis sentiment in bukalapak comments with k-means clustering method," *IJNMT (International Journal of New Media Technology)*, vol. 9, no. 2, pp. 87–92, Jan. 2023. [Online]. Available: https://ejournals.umn.ac.id/index.php/IJNMT/article/view/2914

[3] P. H. Christian and R. I. Desanti, "The comparison of sentiment analysis algorithm for fake review detection of the leading online stores in indonesia," in *2022 Seventh International Conference on Informatics and Computing (ICIC)*, 2022, pp. 01–04.

[4] D. A. Kristiyanti, R. Aulianita, D. A. Putri, L. A. Utami, F. Agustini, and Z. I. Alfianti, "Sentiment classification twitter of lrt, mrt, and transjakarta transportation using support vector machine," in *2022 International Conference of Science and Information Technology in Smart Administration (ICSINTESA)*. IEEE, Nov. 2022. [Online]. Available: http://dx.doi.org/10.1109/ICSINTESA56431.2022.10041651

[5] B. Liu, *Sentiment Analysis and Opinion Mining*. Springer International Publishing, 2012. [Online]. Available: http://dx.doi. org/10.1007/978-3-031-02145-9

[6] S. Saad and B. Saberi, "Sentiment analysis or opinion mining: A review," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 7, no. 5, p. 1660, Oct. 2017. [Online]. Available: http://dx.doi.org/10.18517/ijaseit.7.4.2137

[7] M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artificial Intelligence Review*, vol. 55, no. 7, p. 5731–5780, Feb. 2022. [Online]. Available: http://dx.doi.org/10.1007/s10462-022-10144-1

[8] J. F. Sanchez-Rada and C. A. Iglesias, "Social context in sentiment´ analysis: Formal definition, overview of current trends and framework for comparison," *Information Fusion*, vol. 52, p. 344–356, Dec. 2019. [Online]. Available: http://dx.doi.org/10.1016/j.inffus.2019.05.003

[9] M. Farhadloo and E. Rolland, *Fundamentals of Sentiment Analysis and Its Applications*. Springer International Publishing, 2016, p. 1–24. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-30319-2 1

[10] F. Aftab, S. U. Bazai, S. Marjan, L. Baloch, S. Aslam, A. Amphawan, and T. K. Neo, "A comprehensive survey on sentiment analysis techniques," *International Journal of Technology*, vol. 14, no. 6, p. 1288, Oct. 2023. [Online]. Available: http://dx.doi.org/10.14716/ijtech. v14i6.6632

[11] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: state of the art, current trends and challenges," *Multimedia Tools and Applications*, vol. 82, no. 3, p. 3713–3744, Jul. 2022. [Online]. Available: http://dx.doi.org/10.1007/s11042-022-13428-4

[12] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, p. 27–48, Apr. 2016. [Online]. Available: http://dx.doi.org/10. 1016/j.neucom.2015.09.116

[13] Z. Hao, "Deep learning review and discussion of its future development," *MATEC Web of Conferences*, vol. 277, p. 02035, 2019. [Online]. Available: http://dx.doi.org/10.1051/matecconf/201927702035

[14] K. Gulati, S. Saravana Kumar, R. Sarath Kumar Boddu, K. Sarvakar, D. Kumar Sharma, and M. Nomani, "Comparative analysis of machine learning-based classification models using sentiment classification of tweets related to covid-19 pandemic," *Materials Today: Proceedings*, vol. 51, p. 38–41, 2022. [Online]. Available: http://dx.doi.org/10.1016/j.matpr.2021.04.364

[15] S. K. Bharti, R. K. Gupta, P. K. Shukla, W. A. Hatamleh, H. Tarazi, and S. J. Nuagah, "Multimodal sarcasm detection: A deep learning approach," *Wireless Communications and Mobile Computing*, vol. 2022, p. 1–10, May 2022. [Online]. Available: http://dx.doi.org/10.1155/2022/1653696

[16] P. D. Mahendhiran and S. Kannimuthu, "Deep learning techniques for polarity classification in multimodal sentiment analysis," *International Journal of Information Technology & Decision Making*, vol. 17, no. 03, p. 883–910, May 2018. [Online]. Available: http://dx.doi.org/10.1142/S0219622018500128

[17] B. Ghojogh and A. Ghodsi, "Recurrent neural networks and long shortterm memory networks: Tutorial and survey," 2023. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2023arXiv230411461G/abstract

[18] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ser. ICML'13. JMLR.org, 2013, p. III–1310–III–1318. [Online]. Available: https://dl.acm.org/doi/10.5555/3042817.3043083

[19] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 06, no. 02, p. 107–116, Apr. 1998. [Online]. Available: http://dx.doi.org/10.1142/S0218488598000094

[20] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, p. 2222–2232, Oct. 2017. [Online]. Available: http://dx.doi.org/10.1109/TNNLS.2016. 2582924

[21] G. for Geeks, "Deep learning — introduction to long short term memory," 12 2023. [Online]. Available: https://www.geeksforgeeks.org/ deep-learning-introduction-to-long-short-term-memory/

[22] 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, p. 395–412, Dec. 2020. [Online]. Available: http://dx.doi.org/10.1515/jisys-2020-0021

[23] G. for Geeks, "Bidirectional lstm in nlp," 12 2023. [Online]. Available: https://www.geeksforgeeks.org/bidirectional-lstm-in-nlp/

[24] A. Agarwal, "Sentiment analysis using bi-directional lstm," 5 2020. [Online]. Available: https://www.linkedin.com/pulse/sentiment-analysis-using-bi-directional-lstm-ankit-agarwal

[25] M. A. Nurrohmat and A. SN, "Sentiment analysis of novel review using long short-term memory method," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*,

vol. 13, no. 3, p. 209, Jul. 2019. [Online]. Available: http://dx.doi.org/10.22146/ijccs.41236

[26] A. C. M. V. Srinivas, C. Satyanarayana, C. Divakar, and K. P. Sirisha, "Sentiment analysis using neural network and lstm," *IOP Conference Series: Materials Science and Engineering*, vol. 1074, no. 1, p. 012007, Feb. 2021. [Online]. Available: http://dx.doi.org/10.1088/1757-899X/ 1074/1/012007

[27] N. K. Gondhi, Chaahat, E. Sharma, A. H. Alharbi, R. Verma, and M. A. Shah, "Efficient long short-term memory-based sentiment analysis of e-commerce reviews," *Computational Intelligence and Neuroscience*, vol. 2022, p. 1–9, Jun. 2022. [Online]. Available: http://dx.doi.org/10.1155/2022/3464524

[28] said achmad, "Product reviews dataset for emotions classification tasks - indonesian (prdect-id) dataset," 2022. [Online]. Available: https://data.mendeley.com/datasets/574v66hf2v/1

[29] A. Gholamy, V. Kreinovich, and O. Kosheleva, "Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation," 2 2018. [Online]. Available: https://scholarworks.utep.edu/cs techrep/ 1209/