Emotional Text Classification Using TF-IDF (Term Frequency-Inverse Document Frequency) And LSTM (Long Short-Term Memory)

Muhammad Ibnu Alfarizi[1], Lailis Syafaah[2], Merinda Lestandy[3*]

[1,2,3]*Department of Electrical Engineering, Faculty of Engineering, Universitas Muhammadiyah Malang*

[1]alfarizi.student@webmail.umm.ac.id, [2]lailis@umm.ac.id,
[3*]corr_author: merindalestandy@umm.ac.id

Abstract - Humans in carrying out communication activities can express their feelings either verbally or non-verbally. Verbal communication can be in the form of oral or written communication. A person's feelings or emotions can usually be seen by their behavior, tone of voice, and expression. Not everyone can see emotion only through writing, whether in the form of words, sentences, or paragraphs. Therefore, a classification system is needed to help someone determine the emotions contained in a piece of writing. The novelty of this study is a development of previous research using a similar method, namely LSTM but improved on the word weighting process using the TF-IDF method as a further process of LSTM classification. The method proposed in this research is called Natural Language Processing (NLP). The purpose of this study was to compare the classification method with the LSTM (Long Short-Term Memory) model by adding the word weighting TF-IDF (Term Frequency–Inverse Document Frequency) and the LinearSVC model, as well to increase accuracy in determining an emotion (sadness, anger, fear, love, joy, and surprise) contained in the text. The dataset used is 18000, which is divided into 16000 training data and 2000 test data with 6 classifications of emotion classes, namely sadness, anger, fear, love, joy, and surprise. The results of the classification accuracy of emotions using the LSTM method yielded a 97.50% accuracy while using the LinearSVC method resulted in an accuracy value of 89%.

Keywords: Emotional text classification, TF-IDF, LSTM, LinearSVC

I. INTRODUCTION

Text is one of the media used to effectively communicate and transmit information or messages in the form of text that contains emotion. Voice, facial expressions, gestures, and text sentences are all common ways for people to express their emotions [1]. Emotions are classified into numerous categories, including happy, sad, furious, fearful, and so on. Classification is the process of grouping items with a predefined label or class, whereas sentence classification is the process of classifying a phrase's major theme into numerous groups based on predetermined classes [2]. The classification of emotional texts is one type of sentiment analysis that is used to classify the emotions contained in the text. Text classification can be accomplished in a variety of ways, one of which is through the use of Natural Language Processing (NLP). NLP is a tool that aids a machine's (computer's) understanding of human language. Several prior studies on NLP have yielded positive results and responses in its application. There have been several earlier NLP studies focusing on sentiment analysis utilizing Twitter data [3], scientific sentence classification[2], [4-7], and NLP for opinion mining [8].

The F1 value was 79% in a study [9] that used two separate datasets (subjective and objective) and emotional classes (sadness, joy, fear, wrath, surprise, trust, disgust, and anticipation). Furthermore, the Naïve Bayes approach is used in the study [10] to classify emotions in Indonesian texts, with a ratio of train data to test data of 0.8, resulting in an F-measure value of 62.15%. According to [10] 80% of the training data had a classification accuracy of 94.32% when utilizing the Vector Space Model approach with five emotion classes (fear, anger, disgust, sadness, and joy) [11].

The Recurrent Neural Network (RNN) is one of the most popular architectures used in NLP because its recurrent structure is suitable for text processing [5]. One of the deep learning methods proposed in this research is RNN, with the implementation of the Long Short-Term Memory (LSTM) architecture. RNN can use a distributed word representation by first converting the token consisting of each text into a vector that forms a matrix. In addition, in supervised learning settings, the training data is equipped with a certain label. However, vectorizer standards such as Count vectorizer do not take into account label information when creating text vectors. The TF-IDF (Term Frequency-Inverse Document Frequency) technique aims to overcome this limitation by giving weight to the relationship of a word (term) in a document[7], [12].

Sentiment analysis research using machine learning has been widely carried out in the last few years[13], [14]. Using 10000 positive sentiment data yielded a sentiment accuracy of 93.96%, whereas 8000 negative sentiment data had an accuracy of 88.28% research [15]. In addition, a study [16] utilizing an Indonesian dataset with two sentiment classes (positive and negative) yielded the greatest accuracy of 95%.

LSTM can use a distributed word representation by first converting the tokens consisting of each text into vectors that form a matrix. In addition, in a supervised learning setting, the training data is provided with a specific label. However, vectorizer standards such as the Count vectorizer do not take label information into account when vectorizing text. TF-IDF technique aims to overcome these limitations by giving weight to the relationship of a word (term) in a document [12].

In this study, the LSTM method was applied for classification cases, namely making an accurate prediction of a variable. LSTM is a method that adds a memory cell that can store information for a long time. TF-IDF is used as a word weighting method, TF-IDF not only includes the number of words appearing in a document but also sorts out important and unimportant words in the document, therefore this research uses the LSTM method and the TF-IDF word weighting. The purpose and contribution of this research are the application of the LSTM method to emotional text, the application of word weighting using TF-IDF on each word in the text, improving and analyzing the performance of the method used, and improving the performance of previous research related to classification.

II. METHOD

Obtaining a dataset of emotional data sets for NLP is the first step in the research phase. The data is presented as emotional text and is divided into two sections: 16000 training data was used to train the algorithm in finding the appropriate model and 2000 testing was used to test and determine the performance of the model. The information is divided into six categories or labels (sadness, anger, fear, love, joy, and surprise). Preprocessing will be used to process the emotional text data, which will include tokenizing which is used to break sentences into l/one word, stopword removal is used to sort out less important words or words that have nothing to do with the main word, and stemming is used to find basic words from previously processed words. The data will then be fed into the TF-IDF word weighting algorithm to obtain the vector value of each word. The LSTM approach will be used to classify the outcomes of preprocessing and word weighting. Fig. 1 shows the research methodology. After getting the classification results, the next step is to perform calculations, accuracy, precision, and recall (confusion matrix). A confusion matrix is used to measure performance which has several parameters.

A. Dataset

The dataset used in this study is a collection of documents and their emotions to assist in the case of NLP classification which is divided into two sections to prevent overfitting, namely training data and testing data, both of which are available at https://atapdata.ai/dataset/192/HIMPUNAN_DATA_EMOSI_UNTUK_NLP. based on research [9] taken using the Twitter API. A sentence with six emotional classes was used as the dataset (sadness, anger, fear, love, joy, and surprise) which can represent several other emotion classes such as sadness which represents depression and grief emotions, joy represents fun emotions, and fear represents worried anger represents mad emotions. pissed and others. Table I shows some of the sentiments expressed in emotional text.

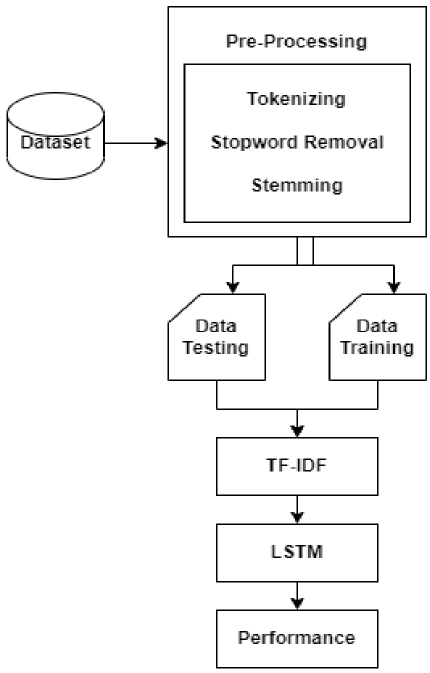

Fig. 1 Proposed method

TABLE I
SENTIMENT TEXT EMOTION

Text	Sentiment
i can go from feeling so hopeless to damned hopeful just from being around someone who cares and is awake	Sadness
im grabbing a minute to post i feel greedy wrong	Anger
i am ever feeling nostalgic about the fireplace i will know that it is still on the property	Love
i feel as confused about life as a teenager or as jaded as a year old man	Fear
i do not feel reassured anxiety is on each side	Joy
i am now nearly finished the week detox and i feel amazing	Surprise

B. Data Preprocessing

Preprocessing of data is performed at this step to speed up data processing. Tokenizing, stopword removal and stemming are among the four phases of preprocessing used in this study.

Tokenization is the initial step. Tokenizing is the initial step in the preprocessing process, and it is used to break down a text's letters or sentences into chunks of words. Whitespace (enter, spaces, and tabulations) can be used to separate characters [17]. Fig. 2 illustrates the tokenization process in emotional text.

The second stage is stopword removal. As shown in Fig. 3, stopwords are used to remove less important words that are not related to the main word, for example, about, such, each, etc. [18]. In this stage, 'i' and 'can' are omitted because after the tokenizing process, 'i' will be considered as a letter, not a word, while can will be considered a conjunction because in English the prefix 'can' is considered as an opening for an interrogative sentence.

The last stage is stemming. Stemming is used to find or change the form of words into basic forms that have the same meaning [19]. The results of the stemming process can be seen in Fig. 4.

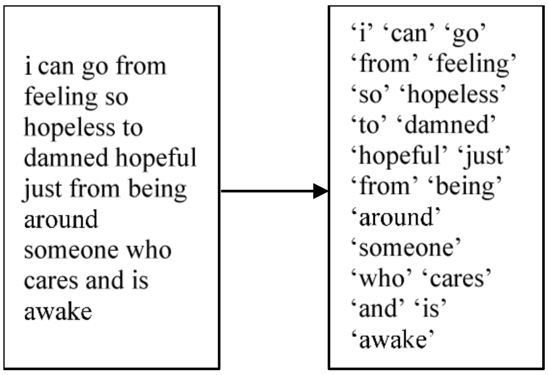

Fig. 2 Tokenizing

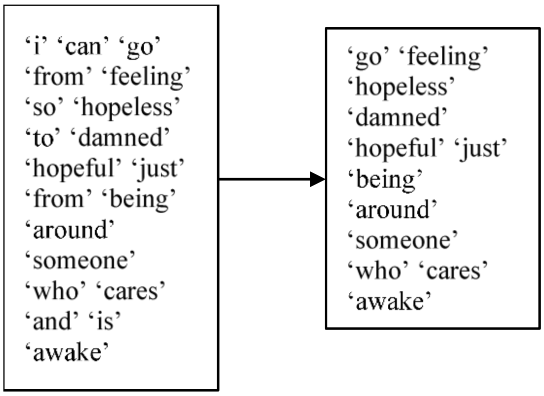

Fig. 3 Stopword

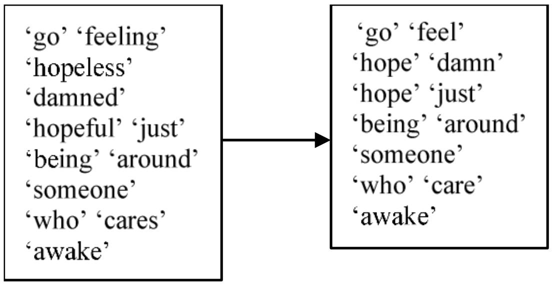

Fig. 4 Stemming

C. TF-IDF

TF-IDF (Terms Frequency-Inverse Document Frequency) aims to give weight to words (terms) in documents that have been preprocessed previously [17]. In this method, two concepts are combined to calculate word weights, namely the frequency of occurrence of a word in a particular document and the inverse frequency of documents containing words that appear in the document. TF-IDF will calculate the value of the Term Frequency (TF) and Inverse Document Frequency (IDF) for each document. To calculate the weight of each term $\boldsymbol{t}$ in the document $\boldsymbol{d}$, eq. (1) [20] is used.

$$\boldsymbol{W_{dt} = TF_{dt} \times IDF_t} \quad (1)$$

where W is the weight of the document d against the word t, and tf is the number of words searched for in a document. In word weighting, the first calculation process is to find the value of TF, which is to calculate the value of the number or frequency of occurrence of a word in the document. Then the second is to calculate the IDF, which is the value of the total number of words in the document, and the last is the calculation of the $TF \times IDF$ value, where the values of TF and IDF will be multiplied. The weighted values will then be converted into an array to be included in the LSTM classification method. There are 3 TF-IDF weighting processes, namely TF calculation to count the number of occurrences of words in the document, IDF and $TF \times IDF$. Table II shows the results of the weighting of emotional texts, The dataset used in this word weighting is the data train.

From the calculation results in Table II, it can be explained that six documents are used in the calculation, namely, D1 is the first document, D2 is the second document, D3 is the third document, and so on as examples of documents used with different emotional classes. The results obtained from each word in each document such as the word 'go' which only exists in the D1 document, the result is 0.778 while the word 'feel' is worth 1 because the word is in each document used. Then the results displayed are by the calculation of the TF-IDF method.

The final vector results are presented in Table II. Each sample consists of the feature set D = {D1, D2, ..., Dn}. The method used in this study is the LSTM method. The second vectorization method used in this study is the Keras tokenizer. To build the tokenizer, we use the following pre-processing process: a) converting all words to lowercase, b) removing less important words that are not related to the main word and c) searching or converting word forms into meaningful basic forms. the same one. The same word index dictionary is used for the test sample to convert the text into sequences to be fed to the LSTM. The final vector for this case has the same structure as that used in the TF-IDF case (presented in Table II).

TABLE II
TF*IDF CALCULATION RESULTS

	TF*IDF					
Term	**D1**	**D2**	**D3**	**D7**	**D15**	**D56**
go	0,778	0	0	0	0	0
feel	1	1	1	1	1	1
hopeless	0,778	0	0	0	0	0
damn	0,778	0	0	0	0	0
hope	0,778	0	0	0	0	0
around	0,778	0	0	0	0	0
someone	0,778	0	0	0	0	0
care	0,778	0	0	0	0	0
awake	0,778	0	0	0	0	0
grab	0	0,778	0	0	0	0
minute	0	0,778	0	0	0	0
greed	0	0,778	0	0	0	0
post	0	0,778	0	0	0	0
wrong	0	0,778	0	0	0	0
ever	0	0	0,778	0	0	0
…	…	…	…	…	…	…
amaze	0	0	0	0	0	0,778

D. LSTM

Long Short-Term Memory is referred to as a neural network with an adaptable architecture, its shape can be adjusted depending on the application. LSTM is a modified type of RNN model by adding a memory cell that can store information for a long period, by eliminating the gradient problem in the RNN that causes errors in capturing long-term dependencies. As a result, the RNN's accuracy on the predicted value is reduced. RNN is an iterative neural network that is used to handle sequential data. However, RNN has vanishing and exploding gradient problems, which is when there is a change in the range of values from one layer to the next in an architecture. LSTM has a function to solve/calculate the problem of gradient disappearing from RNN when dealing with vanishing and exploding gradients. The LSTM consists of an input layer, an output layer, and a hidden layer which is presented in Fig. 5.

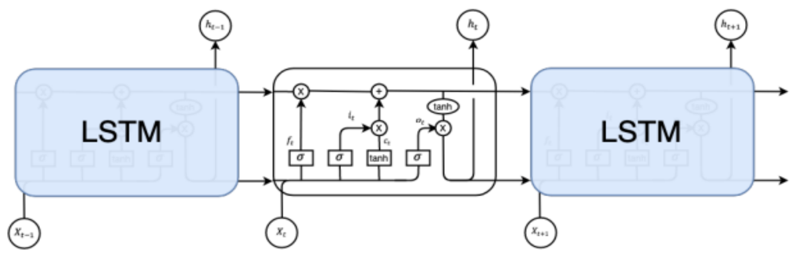

Fig. 5 LSTM layers

The hidden layer consists of memory cells, one memory cell has three gates, namely the input gate, forget gate, and output gate. The input gate controls how much information should be stored in the cell state. This prevents the cell from storing unnecessary data. Forget gate functions to control the extent to which the value remains in the memory cell. Output Gate serves to decide how much content or value in a memory cell, is used to calculate the output. The LSTM model can process information more accurately and has the advantage of having a reminder architecture and forgetting the output, which will then be reprocessed into input [22]. Then the LSTM can maintain errors that occur when backpropagation is done so that the error rate is low.

From Figure 6, it is explained that the first thing to do is sentences that have been processed at the preprocessing stage will be entered into word weighting using the TF-IDF method to convert words into vectors and then words into TF-IDF arrays. The results of the TF-IDF weighting then go to the input layer in the LSTM method. After that, the word will be entered into the LSTM layer. The words will then be reconnected into sentences in a fully connected layer, and the last step is the classification of emotion classes.

E. Performance

Performance measurement in the classification algorithm of this study uses a confusion matrix. The model will be tested using data testing to determine the value of the classification performance using LSTM. The performance results are in the form of accuracy, precision, and recall values. Accuracy is the measure of the degree of closeness between the predicted value and the actual value. Accuracy can be calculated using (2).

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN=FP=FN}} \quad (2)$$

Precision is the value of the level of accuracy of the data from accuracy and prediction. Precision can be calculated using (3).

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \quad (3)$$

While the recall is the success rate of a model in recognizing a class. Recall can be calculated using (4).

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \quad (4)$$

From the above equation, it can be explained that TP (True Positive) is positive data that is predicted correctly, TN (True Negative) is negative data that is predicted to be correct, FP (False Positive) is negative data that is predicted as positive data, and FN (False Negative) is positive data that is predicted as negative data.

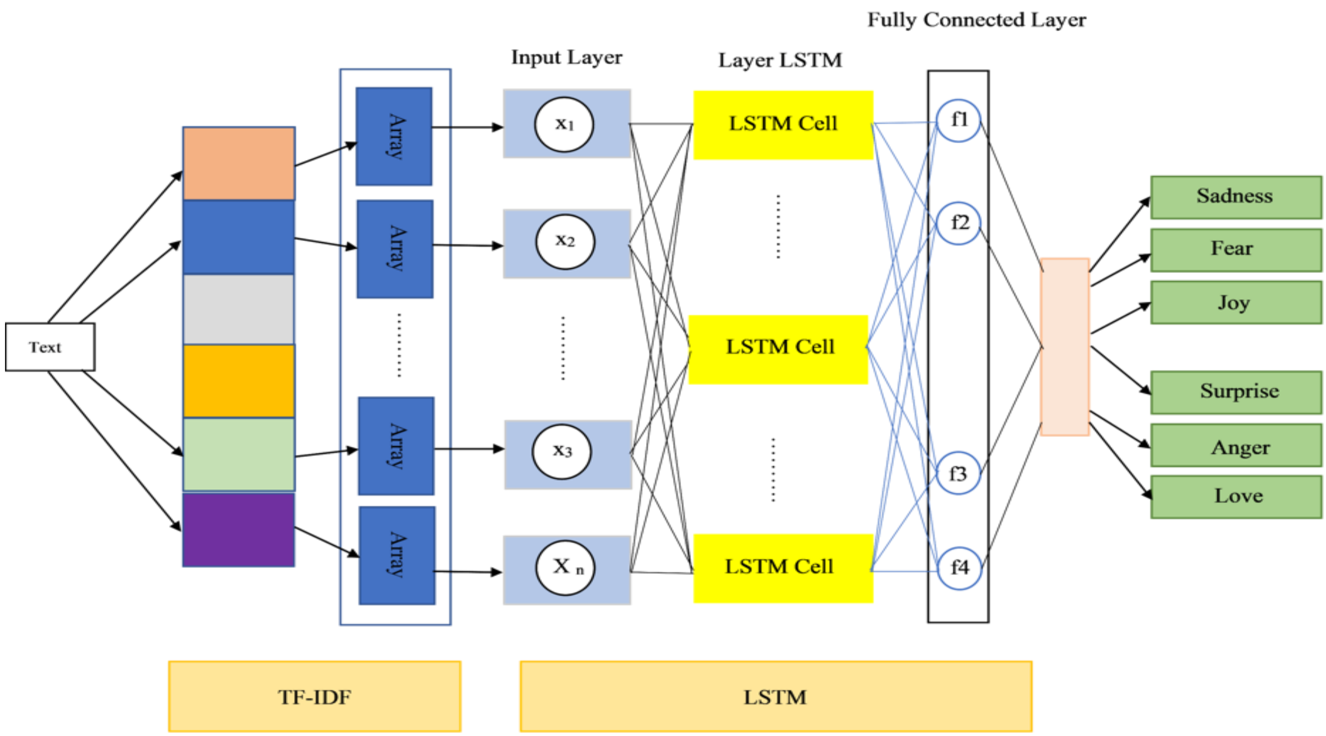

Fig. 6 LSTM architecture

III. RESULTS AND DISCUSSION

The classification algorithms used in this study are LSTM and LinearSVC, which are applied using the TF-IDF weighting technique using the NLP emotional text dataset. The test is carried out by dividing the composition of the training data by 80% and the test data by 20% of the dataset. The results of the classification of the two methods are appropriate for the classification of NLP emotional texts.

Fig. 7 shows that the hidden layer used is 80 with an embedding size of 64. The layer used is 80 and the bidirectional layer is 160 with a dense 6. In this study, several tests were carried out by changing the value of the parameter to find the best value. The parameters to be changed included the number of neurons, activation function, and epoch value.

In Table III, there is a test of the number of neurons in the layer. The number of neurons will be changed four times with the numbers 100, 200, 300, and 400. This test aims to determine the optimal number of neurons in the classification model by showing the highest accuracy test. The best results were obtained on the number of neurons, 200, with a test accuracy value of 93.25%. Next is a test by changing the activation function. This test is done by changing the softmax and sigmoid activation functions in Table IV. The highest accuracy is obtained at 96.89% using the sigmoid activation function.

The last test is to change the epoch value. The results can be seen in Table V. With 256 neurons, a sigmoid activation function, and an epoch value of 30, the results are 93.50%. In the LSTM model, there is a dropout parameter to make the network not experience overfitting, overfitting is a problem that arises when data is trained, the loss is reduced and accuracy is increased. But the accuracy obtained during testing does not increase or continue to decrease. Next is testing the value of cross validation on the LinearSVC method, and the best results are obtained with a cross validation of 20, as shown in Table VI.

The comparison of the confusion matrix between the LSTM and LinearSVC methods using TF-IDF can be seen in Fig. 8 evaluated based on the results of accuracy, precision, and recall. LSTM can classify with an accuracy value of 0.9350, a precision value of 0.93, and a recall value of 0.92. The classification of the LinearSVC method produces an accuracy value of 0.89, a precision value of 0.89, and a recall of 0.84. It can be seen from the results that LSTM accuracy is higher than LinearSVC because the LSTM method has parameters such as embedding, dropout, and dense. While the LinearSVC method only uses cross validation parameters on the model.

TABLE III
TEST THE NUMBER OF NEURONS

Neuron	Test Accuracy	Test Loss
100	92.75%	0.19
200	93.25%	0.16
256	93.30%	0.17
300	92.54%	0.18

TABLE IV
ACTIVATION FUNCTION TEST

Neuron	Activation Function	Accuracy	Loss
256	Softmax	96.79%	0.0592
256	Sigmoid	97.78%	0.0593

```
Model: "sequential_18"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_17 (Embedding)    (None, 80, 64)            973568

 dropout_16 (Dropout)        (None, 80, 64)            0

 bidirectional_35 (Bidirecti  (None, 80, 160)          92800
 onal)

 bidirectional_36 (Bidirecti  (None, 512)              854016
 onal)

 dense_17 (Dense)            (None, 6)                 3078

=================================================================
Total params: 1,923,462
Trainable params: 1,923,462
Non-trainable params: 0
_________________________________________________________________
None
```

Fig. 7. LSTM Parameters

TABLE V
EPOCH TESTING

Epoch	Accuracy	Loss	Validation Accuracy	Validation Loss	Test Accuracy
10	0.9693	0.0839	0.9295	0.1910	0.9244
20	0.9927	0.0955	0.9335	0.2548	0.9319
30	0.9935	0.0935	0.9355	0.1632	0.9350
40	0.9928	0.0924	0.9385	0.3692	0.9235
50	0.9935	0.0919	0.9376	0.3744	0.9204

TABLE VI
CROSS VALIDATION TEST

Cross validation	Mean F1 Macro	Standard Deviation
10	0.856706	0.008168
20	0.859044	0.015267
30	0.857872	0.01594

The comparison of the accuracy values between the classification methods is shown in table 7. The Vector Space Model method in [11] produces an accuracy value of 94.32%. Another study using the Naive Bayes method obtained an accuracy of 61.57% [10]. The results of LSTM accuracy are [15-16], [23]of 93.96%, 95%, and 79.46%, respectively the LSTM (TF-IDF) method has a high accuracy value of 97.50%. A comparison of accuracy between methods is shown in Table VII.

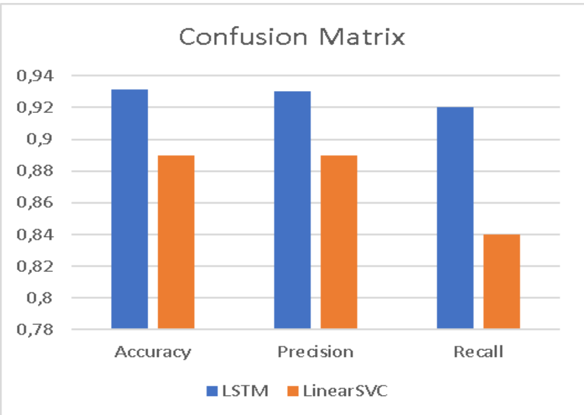

Fig. 8 Confusion matrix

TABLE VII
METHOD COMPARISON

Methods	Accuracy
LSTM (TF-IDF)	97,50%
LinearSVC (TF-IDF)	89%
Multinomial Naïve Bayes [10]	62,15%
Vector Space Model [11]	94,32%
LSTM [24]	93,96%
LSTM [16]	95%

IV. CONCLUSION

This research was successfully carried out by comparing the performance of the RNN and LinearSVC methods using TF-IDF weighting. The dataset used is 18000, which is divided into 16000 training data and 2000 test data with 6 classifications of emotion classes, namely sadness, anger, fear, love, joy, and surprise. The LSTM method (TF-IDF) shows the best accuracy result, which is 97.50% compared to LinearSVC (TF-IDF), with an accuracy value of 89%. This paper aims to increase the accuracy in determining an emotion contained in the text using the LSTM method and the LinearSVC method as a comparison.

REFERENCES

[1] J. Abdillah, “Klasifikasi Emosi pada Lirik Lagu menggunakan Metode Bidirectional LSTM dengan Pembobotan GloVe Word Representation Program Studi Sarjana Informatika Fakultas Informatika Universitas Telkom Bandung,” 2020, doi: 10.29207/resti.v4i4.2156.

[2] M. R. Firmansyah, R. Ilyas, and F. Kasyidi, “Klasifikasi Kalimat Ilmiah Menggunakan Recurrent Neural Network,” *Pros. 11th Ind. Res. Work. Natl. Semin.*, vol. 11, no. 1, pp. 488–495, 2020, doi: 10.35313/irwns.v11i1.2055.

[3] M. R. Hasan, M. Maliha, and M. Arifuzzaman, “Sentiment Analysis with NLP on Twitter Data,” *5th Int. Conf. Comput. Commun. Chem. Mater. Electron. Eng. IC4ME2 2019*, pp. 1–4, 2019, doi: 10.1109/IC4ME247184.2019.9036670.

[4] M. A. Fauzi, “Random forest approach fo sentiment analysis in Indonesian language,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 12, no. 1, pp. 46–50, 2018, doi: 10.11591/ijeecs.v12.i1.pp46-50.

[5] W. K. Sari, D. P. Rini, and R. F. Malik, “Text Classification Using Long Short-Term Memory With GloVe Features,” *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 5, no. 2, p. 85, 2020, doi: 10.26555/jiteki.v5i2.15021.

[6] W. K. Sari, D. P. Rini, R. F. Malik, and I. S. B. Azhar, “Klasifikasi Teks Multilabel pada Artikel Berita Menggunakan Long Short- Term Memory dengan Word2Vec,” *Resti*, vol. 1, no. 10, pp. 276–285, 2017, doi: 10.29207/resti.v4i2.1655.

[7] M. Lestandy, A. Abdurrahim, and L. Syafa’ah, “Analisis Sentimen Tweet Vaksin COVID-19 Menggunakan Recurrent Neural Network dan Naive Bayes,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 2, pp. 802–808, 2021, doi: 10.29207/resti.v5i4.3308.

[8] Y. A. Solangi, Z. A. Solangi, S. Aarain, A. Abro, G. A. Mallah, and A. Shah, “Review on Natural Language Processing (NLP) and Its Toolkits for Opinion Mining and Sentiment Analysis,” *2018 IEEE 5th Int. Conf. Eng.*

Technol. Appl. Sci. ICETAS 2018, pp. 1–4, 2019, doi: 10.1109/ICETAS.2018.8629198.

[9] E. Saravia, H. C. Toby Liu, Y. H. Huang, J. Wu, and Y. S. Chen, "Carer: Contextualized affect representations for emotion recognition," *Proc. 2018 Conf. Empir. Methods Nat. Lang. Process. EMNLP 2018*, no. January, pp. 3687–3697, 2020, doi: 10.18653/v1/d18-1404.

[10] S. Sumpeno, "KLASIFIKASI EMOSI UNTUK TEKS BAHASA INDONESIA - Sumpeno - 2009.pdf," no. c, 2009.

[11] G. Sidorov, "Vector space model," *SpringerBriefs Comput. Sci.*, pp. 5–10, 2019, doi: 10.1007/978-3-030-14771-6_2.

[12] Imamah and F. H. Rachman, "Twitter sentiment analysis of Covid-19 using term weighting TF-IDF and logistic regresion," *Proceeding - 6th Inf. Technol. Int. Semin. ITIS 2020*, pp. 238–242, 2020, doi: 10.1109/ITIS50118.2020.9320958.

[13] A. F. Anees, A. Shaikh, A. Shaikh, and S. Shaikh, "Survey Paper on Sentiment Analysis : Techniques and Challenges," *EasyChair*, pp. 2516–2314, 2020.

[14] N. Yadav, O. Kudale, A. Rao, S. Gupta, and A. Shitole, "Twitter Sentiment Analysis Using Supervised Machine Learning," *Lect. Notes Data Eng. Commun. Technol.*, vol. 57, no. April 2020, pp. 631–642, 2021, doi: 10.1007/978-981-15-9509-7_51.

[15] Q. J. Li. Dan, "Text Sentiment Analysis Based on Long Short-Term Memory.pdf," pp. 471–475, 2016, doi: 10.1109/CCI.2016.7778967.

[16] L. Kurniasari and A. Setyanto, "Sentiment analysis using recurrent neural network-lstm in bahasa Indonesia," *J. Eng. Sci. Technol.*, vol. 15, no. 5, pp. 3242–3256, 2020, doi: 10.1088/1742-6596/1471/1/012018.

[17] M. Nurjannah and I. Fitri Astuti, "PENERAPAN ALGORITMA TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF) UNTUK TEXT MINING Mahasiswa S1 Program Studi Ilmu Komputer FMIPA Universitas Mulawarman Dosen Program Studi Ilmu Komputer FMIPA Universitas Mulawarman," *J. Inform. Mulawarman*, vol. 8, no. 3, pp. 110–113, 2013, doi: 10.30872/jim.v8i3.113.

[18] S. Sarica and J. Luo, "Stopwords in technical language processing," *PLoS One*, vol. 16, no. 8 August, pp. 1–13, 2021, doi: 10.1371/journal.pone.0254937.

[19] D. N. Chandra, G. Indrawan, and I. N. Sukajaya, "Klasifikasi Berita Lokal Radar Malang Menggunakan Metode Naive Bayes dengan Fitur N-Gram," *J. Ilm. Teknol. dan Infomasia ASIA*, vol. 10, no. 2, pp. 11–19, 2019, doi: 10.23887/jik.v4i2.2772.

[20] A. A. Maarif, "Penerapan Algoritma TF-IDF untuk Pencarian Karya Ilmiah," *Dok. Karya Ilm. | Tugas Akhir | Progr. Stud. Tek. Inform. - S1 | Fak. Ilmu Komput. | Univ. Dian Nuswantoro Semarang*, no. 5, p. 4, 2015, [Online]. Available: mahasiswa.dinus.ac.id/docs/skripsi/jurnal/15309.pdf

[21] F. Masri, D. Saepudin, and D. Adytia, "Forecasting of Sea Level Time Series using Deep Learning RNN, LSTM, and BiLSTM, Case Study in Jakarta Bay, Indonesia," *e-Proceeding Eng.*, vol. 7, no. 2, pp. 8544–8551, 2020.

[22] K. Ivanedra and M. Mustikasari, "Implementasi Metode Reccurrent Neural Network pada Text Summarization dengan Teknik Abstraktif," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 4, p. 377, 2019, doi: 10.25126/jtiik.2019641067.

[23] R. Cahyadi, A. Damayanti, and D. Aryadani, "Recurrent Neural Network (RNN) dengan Long Short Term Memory (LSTM) untuk Analisis Sentimen data instagram," *J. Inform. dan Komput.*, vol. 5, no. 1, pp. 1–9, 2020, doi: 10.26798/jiko.v5i1.407.