KLIK: Kajian Ilmiah Informatika dan Komputer
ISSN 2723-3898 (Media Online)
Vol 3, No 6, Juni 2023, Hal 1005-1012
DOI 10.30865/klik.v3i6.866
https://djournals.com/klik

Sentiment Analysis of Forest Fires on Social Media Networks Twitter Using the Long Short Term Memory (LSTM) Method

Aziz Alfauzi, Warih Maharani*

Fakultas Informatika, Program Studi Informatika, Telkom University, Bandung, Indonesia
Email: [1]azizalfa@student.telkomuniversity.ac.id, [2,*]wmaharani@telkomuniversity.ac.id
Email Penulis Korespondensi: wmaharani@telkomuniversity.ac.id

Abstract−The topic of forest fires is of significant interest on social media platforms. In this case, Twitter has been used by 11.8 million users as a means to spread information about forest fires. Twitter, a microblogging service launched on July 13, 2006, allows users to share information for free to themselves and others. Public sentiment related to forest fires can be analyzed through opinions and discussions on Twitter social media. This research aims to analyze the Sentiment of Forest Fires on Twitter Social Networks using the Long Short Term Memory (LSTM) Method. The research data was obtained by crawling the Twitter API using the keyword "forest fire." After crawling, 7,000 tweet texts were collected and labeled as "Negative" and "Positive." Through the preprocessing stage, using a 7,000 dataset, the TF-IDF accuracy of the developed LSTM model reached 68.14%. In addition, the GloVe expansion feature was performed with the Tweet corpus, which resulted in an increase in accuracy of 11.77% to 80.13% in the LSTM model. Meanwhile, the FastText expansion feature with the Common Crawl corpus also increased the accuracy by 11.99% to 80.59% on the LSTM model.

Keywords: Forest Fire; Twitter; LSTM; FastText; GloVe

1. INTRODUCTION

Information and communication technology development has allowed users to connect and share information through social media platforms. One popular platform is Twitter, a social network often used for social media interactions. Founded on July 13, 2006 [1], Twitter is a social networking service that prioritizes short updates or microblogging. Twitter has a very large number of users, including in Indonesia, with 11.8 million Twitter social media users [2], based on data (We Are Social & Hootsuite, 2020).

Twitter allows users to tweet about certain topics and participate in discussions. In the context of information dissemination, Twitter can be an effective tool because shared tweets can be analyzed to find out sentiments related to the topic. One of the most frequently discussed topics on Twitter is forest fires. Forest fires have become a major environmental problem in Southeast Asia in recent decades, with significant impacts on the atmosphere, carbon cycle, and various ecosystem services [3]. Forest fires can broadly impact the population, economy, business, and health in the short and long term [3][4]. Indonesia has an amount of carbon stored in peatlands of 28.1 gigatonnes [5]. For example, forest fires in Indonesia in 2015 resulted in the release of 0.89 gigatonnes of carbon dioxide [6].

Community sentiment towards forest fires also varies. In the Indonesian context, disseminating electronic information that harms one's reputation can be a criminal act following the 2008 ITE Law Article 27 Paragraph (3) [7]. Therefore, analyzing sentiment related to forest fires in more detail and accuracy is important. Some previous studies have used deep learning methods, especially the LSTM model, to perform sentiment analysis with good results. For example, in Yuliana Romadhoni's research entitled "Sentiment Analysis of PERMENDIKBUD No.30 on Twitter Social Media Using Naive Bayes and LSTM Methods", the LSTM method achieved an accuracy of 77% [9]. Another study by Lilyani Asri Utami used PSO-based SVM with 87.11% accuracy in sentiment analysis related to forest fires on the news [7]. Usha Devi Gandhi, Priyan Malarvizhi Kumar, Gokulnath Chandra Babu, and Gayathri Karthick also conducted research using Convolutional Neural Networks (CNN) and LSTM with an accuracy of 87.74% in sentiment analysis on Twitter data [8]. Previous research of Winda Kurnia Sari [10] The multilabel text classification task involves categorizing text into categories. In this study, the Deep Learning method, specifically LSTM with GloVe and Word2Vec features, achieved good performance with the highest accuracy of 95.17% and 95.38%, respectively, as well as an average precision, recall, and F1-score of 95%. Previous research [11] showed that the proposed approach in Arabic sentiment analysis outperformed the basic and competitive methods with better results. Using bidirectional LSTM with feature embedding improves the extraction of aspectual opinion target expressions by 39% compared to the baseline method. In addition, using attention-based features by linking aspect embedding to corresponding word embedding in sentiment polarity classification (AB-LSTM PC) improves aspect sentiment polarity identification by 6%. Research [12] uses deep learning techniques with Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) models to identify sentiment patterns in tweet data. Features like word2vec, stopwords, and tweet words are used in model training and integration. The dataset is 50,000 movie reviews from IMDB, and large Twitter data is processed for sentiment classification. The proposed methodology distinguishes the samples from the real-time environment well and improves the system's effectiveness. The Deep Learning algorithms achieved a % testing accuracy of 87.74% for CNN and 88.02% for LSTM in scoring review tweets and identifying movie reviews. Research [13] Muh Amin Nurrohmat, Azhari SN Classify Indonesian novel reviews based on positive, neutral, and negative sentiments using the LSTM method. The test results show that the LSTM method has better accuracy, precision, recall, and f-measure than the Naïve Bayes method. These studies show significant developments in text mining and text processing. However, there are challenges in feature extraction on unstructured text.

The study conducted by Arliyanti Nurdin, Bernadus Anggo Seno Aji, Anugrayani Bustamin, and Zaenal Abidin aims to compare the performance of insertion of words such as Word2Vec, GloVe, and FastText with classification using the Convolutional Neural Network algorithm. The study results reached an F-Measure value of 0.979 for the 20 Newsgroup dataset and 0.715 for Reuters [14]. Therefore, in this study, we will use the LSTM method and the extended GloVe and FastText features to carry out a sentiment analysis of English tweets related to forest fires. The purpose of this study is to measure the performance of the LSTM model in classifying public sentiment regarding forest fires and to analyze the results of sentiment analysis built using this method. This research will also build two corpus, namely the tweet corpus for the GloVe extension feature and the common crawl corpus for the FastText extension feature, to increase the model's accuracy. Classification of reviews through this process will make it easier for users to categorize opinions into negative or positive more precisely. With a deep learning approach, especially the LSTM method, this research is expected to contribute to developing sentiment analysis in the context of forest fires. In this study, the authors will evaluate the performance of the LSTM model and the expansion features used. In addition, the sentiment analysis will provide a deeper understanding of public sentiment regarding forest fires.

2. RESEARCH METHODOLOGY

2.1 Research Stages

This research uses a crawled Twitter dataset and focuses on forest fires. The stages include data preprocessing, data division, and feature extraction using the TF-IDF method. Next, layer addition, feature expansion, and modeling process using Long Short Term Memory (LSTM) with several stages, such as LSTM + Data Ratio + TF-IDF. The results are then obtained by adding layer changes and features. In the final stage, a training and validation process is carried out where the machine learning model is trained using already categorized text data to learn the patterns and features associated with each category, then tested on text data that has never been seen before to measure the extent to which the model can classify unknown text accurately. Research stages is shown in Figure 1.

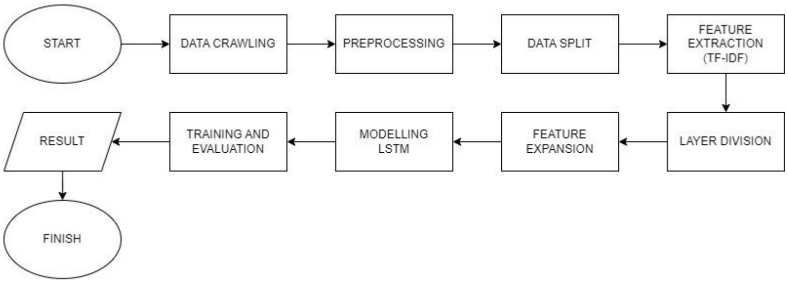

Figure 1. Research Stages

Referring to Figure 1, the research to be carried out has several stages including:

a. Data Crawling
The dataset was retrieved from Twitter using the Twitter API [15], focusing on English tweets related to the topic of truthful information; furthermore, the data was stored in a database for the data preprocessing stage, with a total of 7,000 data collected for sentiment analysis, then the data was saved in CSV format and manually labeled by the author with 0 indicating negative sentiment and 1 indicating positive sentiment based on an internet news search.

b. Preprocessing
The data collected through thorough data retrieval consists of unprocessed raw text, and the data to be used is English tweet data. Before performing sentiment analysis, several preprocessing steps need to be performed to transform unstructured data into more relevant structured data, known as preprocessing.

c. Data Split
At this stage, the 7,000 data is divided into split ratios of 90:10, 80:20, and 70:30, where the data is divided into training data and test data. In addition, an exploration was conducted to find the best ratio for the model-building process.

d. Feature Extraction (TF-IDF)
In this method, TF-IDF is used to obtain the separation of functions. This method measures each word individually, producing an adjusted weight for each word [16]. Equation (1) calculates the mass value, which will later be used in the sentiment analysis procedure. Using the TF-IDF, we can observe how different scales develop over time [20]. The TF value is multiplied by the IDF to produce the final TF-IDF value.

$$idfj = log\ log(D\ dfj) \quad (1)$$

Then do the TF-IDF calculations to get the results. Equation (2) is the TF-IDF calculation formula.

$$wij = tfij \times idfj \quad (2)$$

Information:

tfij = Number of occurrences of the term in the document
wij = The weight of the time in the document
D = Number of all documents
idfj = Distribution of words in the document
dfj = Number of documents containing the term

e. Layer Division
Adding layers to the LSTM can help improve the model's ability to understand and capture long-term patterns in sequential data, thus allowing the model to make more accurate and complex predictions.

f. Feature Expansion
Expanding features can help improve the model's ability to capture additional information or relevant context from the data, thereby enriching the feature representation. It can improve the model's performance in sentiment analysis tasks by providing a more holistic understanding of the observed text. Two expansion features will be used, namely FastText and Glove. The following explanation is related to this expansion method:

1. Glove
GloVe is an unsupervised learning algorithm to obtain word vector representations [17]. GloVe uses the global matrix factorization method, which is a matrix that represents the presence or absence of words in a document [17]. The following are examples of similar words in GloVe in Table 1:

Table 6. GloVe model architecture

Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
carefull	function	london	skiing	ev1ahoucza

In Table 1 above, there is an explanation of the architecture of the GloVe model, which is a method for generating rich vector representations of words based on the statistics of relationships between words in a text corpus. For example, the table shows that the vector representation of the word "careful" obtained from the Twitter tweet dataset in the GloVe model reflects its relationship with the words "function" and "London."

2. FastText
FastText learns word representations by considering subword information. Each word is represented as a set of n-gram characters. Compared with the traditional bag-of-word model, the input layer of the fastText model not only takes the word representation corresponding to each word in the sentence as input and uses the n-gram sentence features as additional features for information [18]. The FastText architectural model is shown in Figure 2 [18].

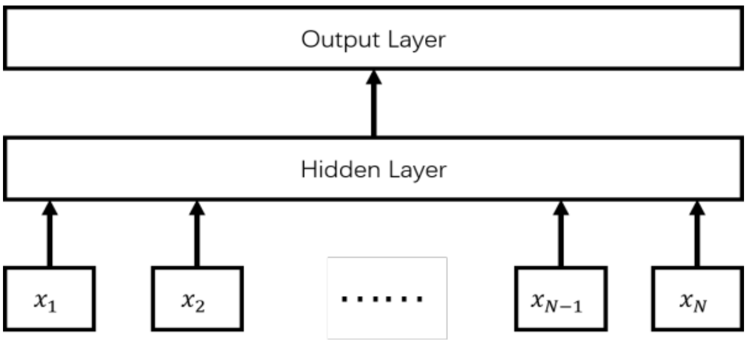

Figure 2. FastText model architecture [18]

Figure 2 above explains the architecture of the FastText model, which is a method for generating word representations by considering subword information in a text corpus. As an example in the table, it is shown that for the sentence "I like play basketball," the inputs of the traditional bag-of-words model are "I," "like," "play," and "basketball," and fastText is the basis of the bag-of-words model. An additional n-gram feature is added above. If n = 2, the additional feature is the average value of the words that correspond to "I like," "like to play," and "play basketball" [18].

g. Modelling LSTM
The Long Short Term Memory (LSTM) method is an approach in deep learning that is highly relevant for various Natural Language Processing (NLP) applications, such as text translation, speech recognition, and image or text classification. Many studies have been conducted using LSTM, showing that this method performs better than conventional methods. The LSTM method has also proven effective in applying sentiment analysis [19]. LSTM consists of four main gates: Input Gate, Forget Gate, Cell Gate, and Output Gate [20]. Figure 3 shows the structure of the LSTM network and its calculation formula.

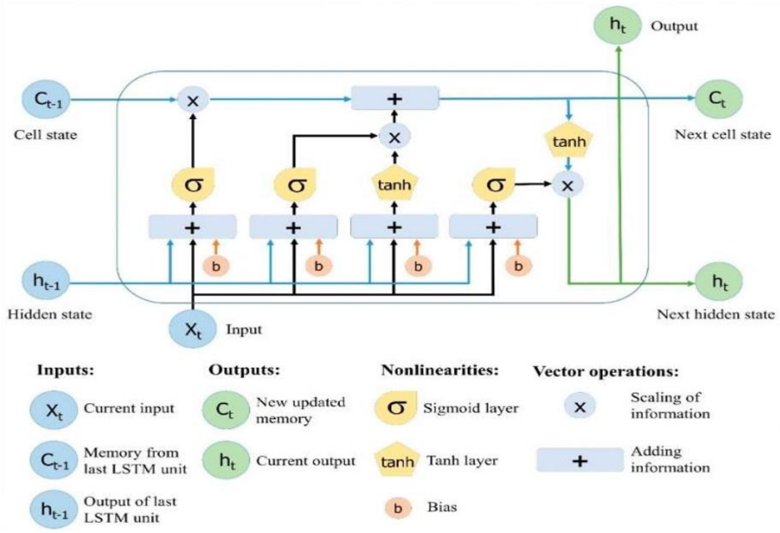

Figure 2. LSTM model architecture [20]

Forget Gate The information contained in the input data will be processed and sorted to determine which data is appropriate to be stored or not feasible to be held in memory cells. There is an equation for the forget gate function, namely [20] :

$$f_1 = \sigma(W_f x1 + U_f h_{t-1} + b_f) \quad (3)$$

Two gates in Input gates will be implemented; the first determines the updated value with the sigmoid activation function. And the tanh activation value will create the latest value vector, which will be stored in the memory cell. The equation is [20] :

$$i_1 = \sigma(W_i x1 + U_i h_{t-1} + b_1) \quad (4)$$

$$C'_1 = tanh(W_c x1 + U_i h_{t-1} + b_c) \quad (5)$$

Cell gates will replace old memory cells with new ones; this value comes from combining forget gates and input gates. The following is the equation of Cell gates [20] :

$$C_1 = (f_1 * C_{t-1} * C'_1) \quad (6)$$

The last step that will be carried out is the output gate, which will carry out two steps. The first step determines the value in the memory cell that will be removed using the sigmoid activation function, followed by placing the memory cell value using the tanh function. The final step of the two gates will be added to get the output value. With the following agreement [20] :

$$o_1 = \sigma(W_o x1 + U_o h_{t-1} + b_o) \quad (7)$$

$$h_1 = o_1 * \ tanh(C_1) \quad (8)$$

Information :
$f_t = Forget\ gate$
$i_t = Input\ gate$
$c_t = Cell\ gate$
$o_t = Ouput\ gate$
$h_t = Hidden\ state$
$c_t = Cell\ gate$
$b = Bias$
$W = Bobot$

h. Performance Measurement

The measurement and evaluation of the performance of the algorithms that have been made have an essential role in assessing and evaluating the performance of the algorithms. System performance uses binary classification metrics from various pre-existing binary classifications to determine accuracy scores [21]. Determining the optimal value for the existing model depends on the desired level of precision. For this evaluation, we used a four-category scale of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) [22]. The Confusion Matrix is used as a framework for further investigation and analysis.

$$Akurasi\,\% = \frac{TP+TN}{TP+FP+FP+FN} \times 100\% \quad (9)$$

$$Presisi\ \% = \frac{TP}{TP+FP} \times 100\% \quad (10)$$

$$Recall\ \% = \frac{TP}{TP+FN} \times 100\% \quad (11)$$

$$F1 - Score\ \% = \frac{2 \times Presisi + Recall}{Presisi + Recall} \times 100\% \quad (12)$$

3. RESULT AND DISCUSSION

3.1 Data Crawling

The dataset is obtained through the Twitter API, which is used to collect tweet data according to the desired topic to find the truth of the information, and the tweet data that this crawling process is carried out in English tweet data. Data is collected using the API (Application Program Interface) provided by Twitter developers as a link from Twitter so that data can be retrieved and processed [15].

After successfully collecting the dataset, the next step is to save the data into the database for reuse in the data preprocessing stage. Data is searched through tweets containing the word "Forest Fire". The total data generated was 7,000 data for the sentiment analysis process. After the data collection process, the data was saved in a CSV file and labeled, with 0 indicating negative sentiment and 1 indicating positive sentiment. Data labeling is done manually by the author by searching for news on the internet. Examples of data that have been labeled can be seen in Table 2.

Table 2. Data Labeling

No	Teks	Label
1	I had to grill for a school function. One of the grills we had going was pretty much either off or forest fire. No inbetween! Made it work	0
2	BE CAREFUL anyone who lives west of Beaverton. Forest Grove has a rapidly spreading fire heading east	1
3	I added a video to a @YouTube playlist http://t.co/bcjYleRRYX Ori and the Bind forest ep 6 'Fire and death'	1
4	This is the first year the Forest Service spent more than half its annual budget on fighting fires. #climatechange http://t.co/D62zfZy0Mi	1
5	Cartoon Bears. Without them we would qave no knowlddge of forest fires or toilet paper.	0

3.2 Preprocessing

a. Data Cleansing

This stage is designed to remove elements that will not be used, such as symbols, punctuation, numbers, emoji, link URLs, mentions in a text tweet, etc [9]. The goal is to clean data from outside elements so that the resulting text data is cleaner and more organized. The Data Cleansing process can be shown in Table 3.

Table 3. Cleansing data

Input	Output
This is the first year the Forest Service spent more than half its annual budget on fighting fires. #climatechange http://t.co/D62zfZy0Mi	This is the first year the Forest Service spent more than half its annual budget on fighting fires

b. Case Folding

The case folding process helps convert all uppercase letters in a text to lowercase. Do this for all words, including sentence beginnings, names of people, names of cities, etc [9]. The Case Folding process can be shown in Table 4.

Table 4. Case folding

Input	Output
This is the first year the Forest Service spent more than half its annual budget on fighting fires	this is the first year the forest service spent more than half its annual budget on fighting fires

c. Stopword

The stopword process is an important step in data preprocessing in text analysis which aims to eliminate words that are considered unimportant or meaningless, such as conjunctions (such as "and", "or", or "however"), prepositions (such as " in", "from", "to"), and other common words that often appear but are not significant in text analysis. Stopword process can be shown in table 5.

Table 5. Stopword

Input	Output
this is the first year the forest service spent more than half its annual budget on fighting fires	first year forest service spent half annual budget fighting fires

d. Tokenizing

The tokenizing process aims to prepare raw text data to be processed with machine learning models, especially in NLP. This process involves several steps, such as removing certain characters, splitting and combining words, and labeling each token. After tokenization, the text data can be converted into a numeric vector representation for further processing with machine learning models such as text classification models, sentiment analysis, or language translation. The tokenizing process can be shown in Table 6.

Table 6. Tokenizing

Input	Output
first year forest service spent half annual budget fighting fires	"first","year","forest","service","spent", "half","annual","budget","fighting","fires"

3.3 Test results

System testing and analysis of the test results are carried out at this stage. This test and research follow the objectives described in the introduction. System testing consists of three scenarios. The first scenario is split data 70:30, 80:20, and 90:10. The second scenario involves testing using layer changes, namely using two, three, and four layers of LSTM. In the last system, testing into the common crawl corpus from FastText, the results will be seen for the accuracy value and tested on the tweet corpus from GloVe, and the results will be seen for the accuracy value in the model built. Namely, Long Short Term Memory and the final process will be evaluated.

3.3.1 Scenario 1 (Ratio + TF-IDF)

In this scenario, model testing is carried out by dividing the data into the same ratio for each model, namely 80:20, 70:30, and 90:10, and adding feature extraction using the TF-IDF method. Table 7 shows that the data ratio of 80:20, after calculating using the confusion matrix, experienced a slight increase in accuracy and F1-score compared to the previous scenario but decreased in accuracy and F1-score from the following procedure. Precision and recall show an adequate balance, showing good model quality in classifying and recalling data. The results of this ratio will be used in the next stage, including accuracy, F1-score, precision, and recall, as a comparison. Model testing is carried out using the data available in the LSTM model. The following are the results of testing the model with variations in data ratios :

Table 7. Experimental results with Data Ratio + TF-IDF.

Model	Ratio	Accuracy	F1-Score	Precision	Recall
LSTM + TF-IDF	90:10	67,63%	68,45%	67,92%	67,80%
LSTM + TF-IDF	80:20	68,14%	67,99%	68,17%	68,15%
LSTM + TF-IDF	70:30	68,67%	66,07%	68,74%	68,49%

3.3.2 Scenario 2 (Rasio + TF-IDF + Layer)

In this scenario, model testing is carried out by adding several parameters, namely layer 2, layer 3, and layer 4, as well as additional feature extraction using TF-IDF. Table 8 shows that the ratio used results from previous experiments, namely 80:20, with the addition of the TF-IDF feature extraction. The test results show increased accuracy, recall, and F1-Score compared to the last 80:20 scenario, which only uses one layer and TF-IDF feature extraction. However, there is a slight drop in the F1-Score. Models with two layers on the LSTM can increase accuracy by 1.37%, recall by 1.44%, and precision by 1.19%. Even though there was a decrease in the F1-Score of 0.83%, the high F1-Score was maintained. This is possible because adding layers can strengthen the model's ability to classify samples to increase recall. However, adding layers can also make the model more complex and prone to overfitting, reducing the F1-Score. Following are the results of testing the model using several layer parameters:

Table 8. LSTM + Ratio + TF-IDF + Layer test results.

Model	Ratio	Accuracy	F1-Score	Precision	Recall
LSTM + 2 Layer	80:20	69,51%(+1,37%)	67,16%(-0,83 %)	69,61%(+1,19%)	69,37%(+1,44%)
LSTM + 3 Layer	80:20	68,14%(+0%)	64,51%(-3,48%)	68,46%(+0,29%)	67,90%(-0,25%)

3.2.3 Scenario 3 (Feature expansion GloVe and FastText)

In this scenario, testing is carried out by extending the features using the GloVe and FastText methods. Testing was carried out in scenario 3 using the previously built corpus. Table 9 shows that the ratio used results from previous experiments, namely 80:20, with the addition of TF-IDF feature extraction and two layers. The test results show increased accuracy, recall, and F1-Score compared to scenario 2. In the third scenario, there is a consistent increase in accuracy, precision, and recall. This shows that the expansion of features with GloVe and FastText has significantly contributed to increasing the model's ability to classify and remember data accurately. The LSTM model, with the addition of the FastText expansion feature, can improve accuracy by 11.99%, F1-score by 14.68%, recall by 19.42%, and precision by 16.85%. Meanwhile, the LSTM model, with the addition of the GloVe expansion feature, increased accuracy by 11.73%, F1-score by 14.68%, recall by 19.03%, and precision by 15.67%. The following are the results of model testing using feature extensions:

Table 9. Performance of GloVe tweet corpus and FastText common crawl on the LSTM model.

Model	Accuracy	F1-Score	Precision	Recall
LSTM + 2 Layer+ GloVe (tweet corpus)	80,13%(+11,77%)	78,61%(+14,68%)	80,37%(+15,67%)	80%(+19,03%)
LSTM + 2 Layer+ FastText (common crawl corpus)	80,59%(+11,99%)	78,61%(+14,68%)	81,13%(+16,58%)	80,39%(+19,42%)

4. CONCLUSION

This research has developed Sentiment Analysis of Forest Fires on Twitter Social Networks Using the Long Short Term Memory (LSTM) Method. This research uses two feature expansion methods, namely FastText and GloVe. The gloVe is an extension feature that reduces sentence vocabulary differences in a data set. FastText is an extension of the word representation feature that considers subword information. Feature expansion is done by building a GloVe corpus (Tweets), and Feature expansion is done by building a FastText corpus (Common Crawl). Researchers also tested the model by determining the same data ratio in each model, namely 80:20, 70:30, and 90:10. The division of this data ratio is done to find the data ratio that produces accuracy with the largest results in the 80:20 ratio with an accuracy value of 68.14% with the addition of TF-IDF extraction features, the purpose of dividing this ratio is to get a good model in classifying and recalling data. Furthermore, the data ratio is used to find the accuracy of the model to be built using the TF-IDF extraction feature, and adding layers obtained an accuracy of 69.51%, from the test results of adding this layer will be tested with the addition of expansion features it can be seen that the Long Short Term Memory (LSTM) model with the expansion of FastText (common crawl) features has higher accuracy than the LSTM model with the expansion of GloVe (tweet) features, which is 80.59% and 80.13% respectively. Thus, adding the expansion feature using FastText (common crawl) can increase the accuracy of the previous model by 11.99%, while adding the expansion feature using GloVe (tweet) can increase the accuracy by 11.77%. These results show that the use of expansion features with FastText (common crawl) provides better results in improving the accuracy of the LSTM model. For future research, it is recommended to explore other expansion techniques to improve accuracy further.

REFERENCES

[1] A. Fauzi, E. B. Setiawan, and Z. K. A. Baizal, "Hoax News Detection on Twitter using Term Frequency Inverse Document Frequency and Support Vector Machine Method," in Journal of Physics: Conference Series, May 2019, vol. 1192, no. 1. doi: 10.1088/1742-6596/1192/1/012025.

[2] Andi Dwi Riyanto, "Hootsuite (We are Social): Indonesian Digital Report 2020." [Online]. Available: https://andi.link/hootsuite-we-are-social-indonesian-digital-report-2020/.

[3] Jaafar, Z.; and Loh, T.-L. 2014. "Linking land, air and sea: potential impacts of biomass burning and the resultant haze on marine ecosystems of Southeast Asia".Global Change Biology 2701–2707.

[4] Doerr, S. H.; and Sant´ın, C. 2016. "Global trends in wildfire and its impacts:perceptions versus realities in a changing world". Philosophical Transactions of the Royal Society B: Biological Sciences 20150345.

[5] Warren, M.; Hergoualc'h, K.; Kauffman, J.; Murdiyarso, D.; and Kolka, R.2017. "An appraisal of Indonesia's immense peat carbon stock using national peatland maps: Uncertainties and potential losses from conversion". Carbon Balance and Management 12.

[6] Lohberger, S.; Stangel, M.; Atwood, E.; and Siegert, F. 2018. ¨ Spatial evaluation of Indonesia's 2015 fire affected area and estimated carbon emissions using Sentinel-1. Global Change Biology 644–654.

[7] Y Asri Utami , Lilyani, M.Kom.Maret 2017. "Analisis Sentimen Opini Publik Berita Kebakaran Hutan Melalui Komparasi Alforitma Support VectorMachine (SVM) dan KNearest Neighbor (KNN)".

[8] Usha Devi Gandhi, Priyan Malarvizhi Kumar, Gokulnath Chandra Babu, Gayathri Karthick," Sentiment Analysis on Twitter Data by Using Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM)". https://doi.org/10.1007/s11277-021-08580-3, 4 May 2021.

[9] Yuliana Romadhoni1, Khadijah Fahmi Hayati Holle" Analisis SentimenTerhadap PERMENDIKBUD No.30 pada Media Sosial Twitter Menggunakan Metode Naive Bayes dan LSTM". Jurnal Informatika: Jurnal pengembangan IT (JPIT), Vol.7, No.2, Mei 2022.

[10] Winda Kurnia Sari, "Klasifikasi Teks Menggunakan Long ShortTerm Memory Dengan Fitur Word Embedding Glove Dan Word2Vec" 2020.

[11] M. Al-Smadi, B. Talafha, M. Al-Ayyoub, and Y. Jararweh, "Using long short-term memory deep neural networks for aspect-based sentimen analysis of Arabic reviews," Int. J. Mach. Learn. Cybern., vol. 10, no. 8, pp. 2163–2175, 2019, doi: 10.1007/s13042-018-0799- 4.

[12] J. Shobana and M. Murali, "An efficient sentimen analysis methodology based on long short-term memory networks," Complex Intell. Syst., vol. 7, no. 5, pp. 2485–2501, 2021, doi: 10.1007/s40747- 021-00436-4.

[13] Muh Amin Nurrohmat, Azhari SN."Sentiment Analysis of Novel Review Using Long Short-Term Memory Method" IJCCS (Indonesian Journal of Computing and Cybernetics Systems) Vol.13, No.3, July 2019, pp. 209~218.

[14] Arliyanti Nurdin, Bernadus Anggo Seno Aji, Anugrayani Bustamin, and Zaenal Abidin, "PERBANDINGAN KINERJA WORD EMBEDDING WORD2VEC, GLOVE, DAN FASTTEXT PADA KLASIFIKASI TEKS" Jurnal TEKNOKOMPAK, Vol. 14, No. 2, 2020.

[15] J. Eka Sembodo, E. Budi Setiawan, and Z. Abdurahman Baizal, "Data Crawling Otomatis pada Twitter," Sep. 2016, pp. 11– 16. doi: 10.21108/indosc.2016.111.

[16] A. Aizawa, "An information-theoretic perspective of tf-idf measures q." [Online]. Available: www.elsevier.com/locate/infoproman.

[17] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation." [Online]. Available:http://nlp.

[18] Tengjun Yao, Zhengang Zhai, Bingtao Gao, "Text Classification Model Based on fastText,"2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS).

[19] M. A. Nurrohmat and A. SN, "Sentimen Analysis of Novel Review Using Long Short-Term Memory Method," IJCCS (Indonesian J. Comput. Cybern. Syst., vol. 13, no. 3, p. 209, 2019, doi: 10.22146/ijccs.41236.

[20] X. H. Le, H. V. Ho, G. Lee, and S. Jung, "Application of Long ShortTerm Memory (LSTM) neural network for flood forecasting," Water (Switzerland), vol. 11, no. 7, 2019, doi: 10.3390/w11071387.

[21] V. M. Patro and M. Ranjan Patra, "Augmenting Weighted Average with Confusion Matrix to Enhance Classification Accuracy," Transactions on Machine Learning and Artificial Intelligence, vol. 2, no. 4, Aug. 2014, doi: 10.

[22] Dion Pratama Putra, Erwin Budi Setiawan Hoax Detection Using Long Short-Term Memory (LSTM) and Gate Recurrent Unit (GRU) on Social Media Building of Informatics, Technology and Science (BITS) Volume 4, No 4, Maret 2023 Page: 1815−1820.