Digital Zone
Jurnal Teknologi Informasi dan Komunikasi
Vol: 13 No 02 2022 E-ISSN: 2477-3255
Diterima Redaksi: 27-07-2022 | Revisi: 25-08-2022 | Diterbitkan: 26-10-2022

Sentiment Analysis on the Construction of the Jakarta-Bandung High-Speed Train on Twitter Social Media Using Recurrent Neural Networks Method

Titan Kinan Salaatsa[1], Yuliant Sibaroni[2]
[1,2] Informatics Study Program, Telkom University
[1,2]Jl. Telekomunikasi No. 1, Dayeuh Kolot, Kab. Bandung, Telp (022) 7566456
e-mail: [1]titankinan@student.telkomuniversity.ac.id, [2]yuliant@telkomuniversity.ac.id

Abstract

During the construction of the Jakarta-Bandung high-speed train, many Indonesian people gave their responses to the public. The answers were also varied, with some giving positive and negative reactions. The purpose of this study is to analyze the sentiments of the responses given by the public to the construction of the Jakarta-Bandung high-speed train on Indonesian-language Twitter. To perform sentiment analysis, tweet data was collected utilizing data crawling based on keywords related to the construction of the Jakarta-Bandung high-speed train and given positive, negative, and neutral labels and then represented into numbers using the Keras tokenizer. The method used for sentiment classification of tweet data is the Recurrent Neural Networks method. The highest accuracy results were obtained using the GRU architecture with an accuracy of 69.62%.

Keywords: *high-speed train, sentiment analysis, recurrent neural networks, twitter*

1. Introduction

Transportation is one of the main elements that big cities must begin to accommodate [1]. The train is one of the most important forms of transportation for developing countries such as Indonesia because of its capability to carry loads with large capacities, such as transporting people and goods to solve traffic problems and increase economic development [2]. To facilitate mobility and connectivity on the island of Java, the Indonesian government has begun to build a high-speed train [3]. This high-speed train is proposed for the Jakarta-Surabaya route, with the construction of the Jakarta-Bandung route as the first phase. During the high-speed train construction, many Indonesian people gave their thoughts and opinions. The opinions were varied, some showed positive, and some gave negative opinions

Currently, most people give their thoughts and opinions through various media, including social media [4]. Twitter is one effective social media that reports experiences and shares ideas [5]. To out the tendency of public opinion on Twitter towards something or an event can be done with sentiment analysis [6]. Sentiment analysis provides an understanding of public views to automatically determine what public expressions look like.[7]

https://doi.org/10.31849/digitalzone.v13i1.10777

Several machine learning techniques, such as Naïve Bayes, Maximum Entropy, and Support Vector Machines (SVM), can perform sentiment analysis. However, a neural network model can achieve better accuracy and performance than traditional machine learning models [8].]. The neural network is used because it can provide better results than conventional machine learning methods when using a large amount of data. RNN is also used because the data that is processed is in the form of text, the text is sequential data that has a sequence, and RNN can process sequential data well.

Much research has been done for sentiment analysis using traditional machine learning methods or neural networks. However, sentiment analysis research on constructing the Jakarta-Bandung high-speed train is still difficult to find. Study [8] compared sentiment analysis using three deep learning methods in 8 different datasets, 2 of which are tweets data. The deep learning methods used are DNN, CNN, and RNN. The study also compared the performance of the three methods used when using word embedding and TF-IDF as feature extraction. For the opinion dataset of United States airlines, the best accuracy results using word embedding in the RNN method are 90.45%. The use of word embedding in the RNN method produces better accuracy than the use of TF-IDF in the RNN method.

A subsequent study [9] compared the performance of SVM, LR, and DCNN for sentiment analysis of Twitter data using five different datasets. The SVM and LR methods are implemented using Bag of Words and GloVe as expansion features, while the DCNN method only uses GloVe. The results of testing the five datasets prove that the neural network method used, namely DCNN, had a better performance than SVM and LR methods. The highest result was obtained in the dataset of The Stanford Twitter Sentiment Test (STSTd), with an accuracy of 87.62%. Research [10] comparing two models for sentiment analysis of COVID-19 vaccine tweets. The two models used for sentiment analysis are RNN and Naïve Bayes. The best results were obtained with the RNN model using the TF-IDF technique with an accuracy of 97.77%. The hybrid type gets the best results because it combines the architecture of the other two RNN types but also increases the complexity. There are other studies that analyze sentiment using the IMDB dataset and review polarity in research [11]. The study used the RNN, LSTM, GRU, and LSTM-GRU models. From the resulting accuracy, it proves that the LSTM-GRU model outperforms other models with an accuracy value of 87.10% on the IMDB dataset and 73.24% on the Polarity Review dataset.

Based on the research that has been discussed previously that the RNN method can obtain good accuracy and performance compared to several other methods, the author proposes to use the RNN method in this study. The RNN model was chosen because it has high reliability over the traditional machine learning method [8]. In this study, the author will use random sampling to balance the number of data labels. The author will also compare the RNN approach using the LSTM and GRU architectures using 10-fold cross validation to evaluate the performance of the model. Both LSTM and GRU architectures are used because LSTM and GRU can overcome the shortcomings of RNN, where RNN has short-term memory problems if the weight is too large [12]. This study aims to see how much accuracy is obtained from the method chosen by the author.

2. Research Method

The following is the design of the system built to analyze sentiment using RNN in this study as shown in Figure 1

2.1. Data Crawling

In this study, datasets were collected from social media Twitter using several keywords and hashtags related to the construction of the Jakarta-Bandung high-speed train, such as "kereta cepat Jakarta-Bandung" and "#pembangunankeretacepat" in the period of January 2016 until February 2022. The method used for Twitter data collection in this study uses the data crawling method using the Twitter API and using the Python Twint library. After the crawled data is successfully obtained, all the data is stored in a csv file. This data will be used as a training model for sentiment analysis.

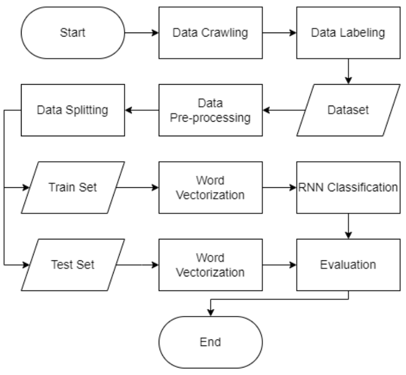

Figure 1. RNN Classification Model Process

2.2. Data Labeling

After getting Twitter data, the data is manually labeled into 3 different labels, namely positive labels, negative labels, and neutral labels. Data labeling was carried out in this study using the majority voting method with 3 annotators. Data that has a positive label is data that contains a positive sentiment towards the construction of the Jakarta-Bandung high-speed train, as well as data that has a negative label, while data that has a neutral label is data that is neither has positive nor negative sentiment or not related to the construction of the Jakarta-Bandung high-speed train. In the following table 1 is an example of labeling Twitter data.

Table 1. Data Labeling

Tweet	Label
Negara bangkrut sok sok an mau bangun ibu Kota baru di Kalimantan ditambah lagi kereta cepat Jakarta Bandung sudah membengkak, DPR cuma satu orang dua orang yg menyuarakan yg lain mana? #NegaraRampasJHTrakyat	Negative
@jokowi Keren pak mantap moga agenda pagi ini lancar terutama project sinkanzen bhasa kerennya kereta cepat terealisasi ga pake lama sebelum akhir 2022 uda diresmikan rute Jakarta - Bandung	Positive
@geloraco Katanya TNI disegani di dunia, ayolah jgn sampai ada berita lagi TNI meninggal di tembak KKB. Mana kepedulian pemerintah kenapa beda banget kalo orang Islam langsung gerak cepat secepat kereta cepat Jakarta Bandung, pasalnya dicari cari gak ada	Neutral

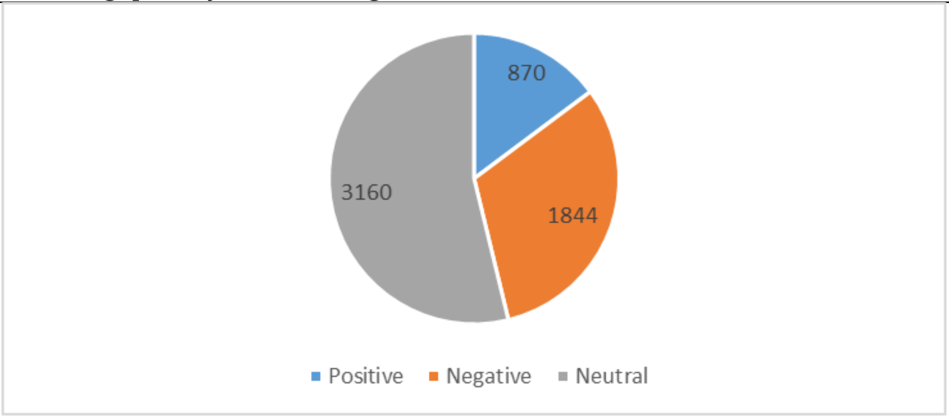

Figure 2. Labels Comparison

From the graph in Figure 2, it shows that the results of the data labeling process obtained 3160 data with a neutral label, 1844 data with a negative label, and 870 data with a positive label.

2.3. Data Preprocessing

Before the data is processed for sentiment analysis, the data needs to go through preprocessing. Data preprocessing is used to convert raw data into data that can be easier to get the information contained. There are several steps of data preprocessing carried out in this study including:

1. Case folding, converts all letters to lowercase.
2. Cleaning, remove noise in text such as usernames, hashtags, URL links, numbers, and punctuation.
3. Tokenization, break the text into tokens or per word contained in the text.
4. Normalization, change the text into standard language and use the correct spelling by using the corpus in the study [13] which has been modified and then matched with the dataset used.
5. Stemming, make it just a root word, removing the suffix and prefix using stemmer from the Python Sastrawi library.
6. Stopwords, delete words that are considered unimportant such as “yang”, “itu”, “dan” using a dictionary that has been customized to the dataset used.

Table 2 below is an example of the application of each preprocessing step in the table:

Table 2. Preprocessing Steps

Steps	Tweet
Before Preprocessing	@CNNIndonesia Sampai selesai nga? China bukan yg talangi dana pembangunan nya? Jangan2 nasibnya kaya kereta api cepat jakarta bandung wkwkkwkw
Case folding	@cnnindonesia sampai selesai nga? china bukan yg talangi dana pembangunan nya? jangan2 nasibnya kaya kereta api cepat jakarta bandung wkwkkwkw
Cleaning	sampai selesai nga china bukan yg talangi dana pembangunan nya jangan nasibnya kaya kereta api cepat jakarta bandung wkwkkwkw
Tokenization	[sampai, selesai, nga, china, bukan, yg, talangi, dana, pembangunan, nya, jangan, nasibnya, kaya, kereta, api, cepat, jakarta, bandung, wkwkkwkw]
Normalization	[sampai, selesai, tidak, cina, bukan, yang, talangi, dana, pembangunan, nya, jangan, nasibnya, seperti, kereta, api, cepat, jakarta, bandung, wkwk]
Stemming	[sampai, selesai, tidak, cina, bukan, yang, talang, dana, bangun, nya, jangan, nasib, seperti, kereta, api, cepat, jakarta, bandung, wkwk]
Stopwords	[selesai, cina, talang, dana, bangun, nya, nasib, kereta, api, cepat, jakarta, bandung, wkwk]

2.4. Word Vectorization

After passing the preprocessing steps, the data will be converted into vector form by converting every word contained in the dataset into integer numbers using the keras tokenizer library. Each integer used is an index from a dictionary that is created based on the number of words. After that, each data will be padded so that all data have the same length and can be used as input to the RNN model by determining the maximum length first. By using the hard library pad_sequences, all data that has a length exceeding the specified length will be discarded, and data shorter than the specified length will be filled with 0 [14].

2.5. Recurrent Neural Networks

Tweet data used in this study is text data. Text is one example of sequential, sequential data is data that has a sequence or is continuous such as video, audio, and time series [15]. Because RNN works repeatedly in processing sequential input data [16], RNN can properly process and recognize sequential data such as tweet data used in this study.

RNN has 3 main layers including input layer, hidden layer, and output layer. The input layer receives the data to be processed, the hidden layer consists of 1 or more layers to process the data given from the input layer, and the output layer stores the results of the previous process. The created model can minimize errors with a high level of complexity if given the appropriate number of hidden layers [17]. RNN is an architecture of an artificial neural network that has feedback as output from the hidden layer and enters the input layer [15]. In other words, the output of the network on the RNN can be reused as input. With this capability, the RNN can process the current input and remember what has been learned from the previous input [18]. The following is an architectural description of the RNN as shown in Figure 3:

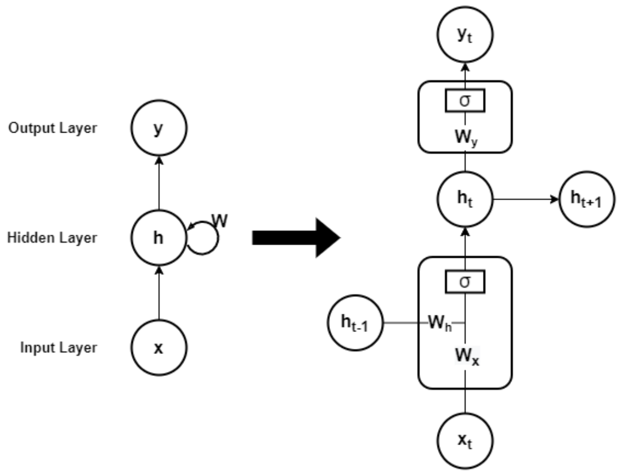

Figure 3. RNN Architecture

In the Tensorflow Keras library, the calculations for the RNN architecture are as follows:

$$h_t = \sigma(W_x x_t + W_h h_{t-1} + b_h) \tag{1}$$

$$y_t = \sigma\left(W_y h_t + b_y\right) \tag{2}$$

Where $W_x x_t$ is the weight of the input, $W_h h_{t-1}$ is the weight stored in the previous hidden layer, $W_y h_t$ is the weight of the current hidden layer, and b is the bias.

One of the limitations of RNN is short-term memory, for this reason, this study will use another architecture, namely LSTM and GRU. LSTM can remember inputs received for a long time, because LSTM has cell gates that can be opened and closed, namely input gate, forget gate, and output gate [18]. With this capability, the LSTM can choose to forget or keep the processed information based on the how importance of the information it obtains. While the GRU has an update gate and a reset gate, the update gate decides how much information from the previous memory is to be stored, and the reset gate determines how to combine the new input with the previous memory [19].

2.6. Evaluation

The results of the classification need to be evaluated to determine the performance of the model that has been made. There are several performance evaluations that can be calculated using the confusion matrix elements including accuracy, precision, recall, and f1-score [20]. The results of the performance evaluation can be obtained using the following formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{6}$$

Where True Positive (TP) is the proportion of positive labels that are predicted to be correct, False Negative (FN) is the proportion of positive labels that are predicted to be wrong, False Positive (FP) is the proportion of negative labels that are predicted to be wrong, and True Negative (TN) is the proportion of negative labels that are predicted to be correct. The four elements can be obtained from the following Confusion Matrix table:

Table 3. Confusion Matrix

		Actual Class	
		True	**False**
Predicted Class	**True**	TP	FP
	False	FN	TN

3. Result and Analysis

In this study, there are several steps carried out before reaching the evaluation step. The first step is data crawling and data labeling from the tweet data that has been obtained. Second, the previous data will be preprocessed and then the data will be splitted into training data and testing data. After that the data is converted into vector form. Then the data is classified using the RNN model. The following are some of the scenarios used in this study:

1. Scenario 1: Testing the effect of balancing data train on the RNN model
2. Scenario 2: Testing the effect of using different layers architecture and the number of neurons on the RNN model

3.1. Scenario 1 Testing the Balanced Training Data Labels

This scenario was conducted to determine the effect on the balanced training data on the RNN model. There are 3 training data used, namely oversampling training data, undersampling training data, and baseline training data. The test results from scenario 1 can be seen in table 3.

Table 4. Balanced Train Data Test Result

Method	**Accuracy**	**F1-score**
Baseline	**68.76%**	**68.75%**
Oversampling	68.51%	68.50%
Undersampling	60.58%	60.39%

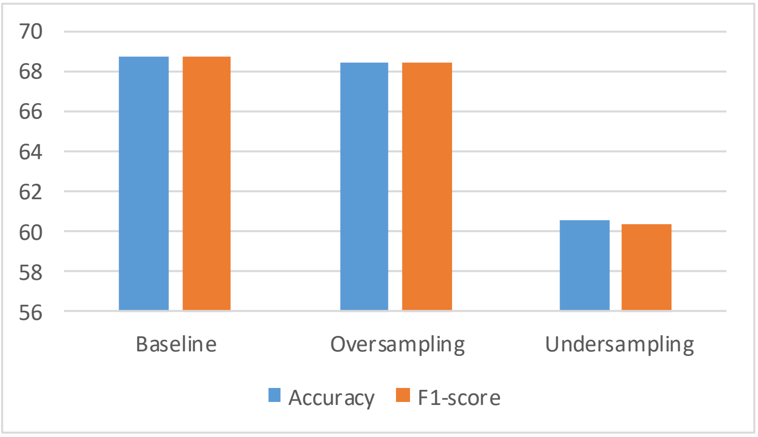

Figure 4. Balanced Training Data Labels Accuracy and F1-score Comparison

From the results obtained in this scenario, the RNN model trained using baseline training data has the highest accuracy compared to using training data that has been balanced by oversampling and undersampling. By using train baseline data, the highest accuracy is obtained, which is 68.76%.

3.2. Scenario 2 Testing the Use of Different Layers and the Number of Neurons

In scenario 2, the use of different layers and the number of neurons was tested to determine the effect on the RNN model created. The layers used are SimpleRNN, LSTM, and GRU, while the number of neurons used are 50, 100, 150, 200, and 250. Evaluation in this scenario uses 10-fold cross validation to estimate the performance of the model created. The results of the scenario 2 test can be seen in table 5.

Table 5. Layers and Neurons Test Result

Number of Neuron	SimpleRNN			LSTM			GRU		
	Akurasi	**F1-score**	**Waktu Training**	**Akurasi**	**F1-score**	**Waktu Training**	**Akurasi**	**F1-score**	**Waktu Training**
50	68.58%	68.57%	1631 second	69.38%	69.30%	3223 second	68.66%	68.57%	2347 second
100	**69.37%**	**69.32%**	1659 second	69.48%	69.52%	5702 second	69.37%	69.34%	3042 second
150	69.13%	69.06%	2588 second	69.23%	69.19%	7119 second	69.20%	69.19%	5096 second
200	68.92%	68.92%	3747 second	69.48%	69.46%	7552 second	**69.62%**	**69.65%**	6047 second
250	69.26%	69.28%	3797 second	**69.57%**	**69.62%**	10840 second	69.37%	69.36%	15809 second

From the results obtained in scenario 2, it can be seen that the higher number of neurons used in the layer does not mean it will increase the performance of the RNN model created, so the number of neurons must be in the right amount. Models made using LSTM and GRU tend to require a longer training time than using SimpleRNN. The highest accuracy obtained from this scenario is when using the GRU layer with 200 neurons, which is 69.62%, and highest accuracy for the LSTM layer is 69.57% using 250 neurons. While the best results that can be obtained using the SimpleRNN layer is 69.37% using 100 neurons. The following is a comparison of the accuracy with the best model training time for each architecture:

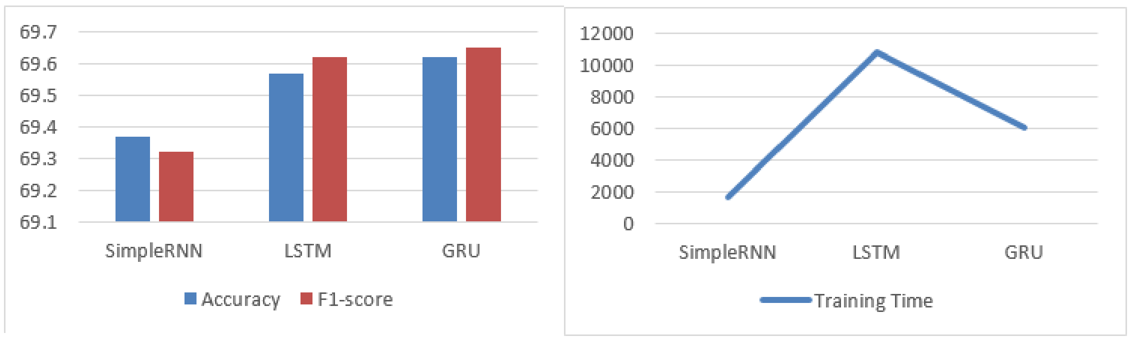

Figure 5. Best Model Accuracy and F1-score Compared with Training Time

3.3. Test Result Analysis

After doing the two test scenarios, it can be concluded that each test can affect the performance of the RNN model created. For the first scenario, the results of testing on random oversample training data, random undersample training data, and baseline training data, it was found that the baseline training data had higher accuracy than the balanced training data. The second scenario is testing the use of the SimpleRNN, LSTM, and GRU layers with different numbers of neurons, the highest accuracy results are obtained using the GRU layer with 200 neurons. The accuracy obtained from the model using the SimpleRNN layer is lower than other models using the LSTM or GRU layer, however, LSTM and GRU require a longer training time than using SimpleRNN. Using the GRU architecture which has 200 neurons can increase accuracy by 0.86% when compared to the best results from scenario 1.

Based on the two scenarios carried out, scenario 1, that is balancing labels on the dataset used, gets the highest accuracy of 68.76%, then scenario 2 is the effect of layers and the number of neurons in the RNN model, gets the highest accuracy of 69.62%. The use of balanced training data provides lower accuracy than baseline training data because random oversample training data will provide duplicate data and result in an overfitting of the model created, while random undersample training data will reduce training data with majority labels so that the number of labels is equal to training data with minority labels causing the model has less training data to train. Model evaluation using 10-fold cross validation can maximize performance because with k-fold cross validation, the model will be trained by taking different train and test data k times to get an estimate of optimal model performance.

When compared with research [11], the results of the greatest accuracy of 87.10% were obtained using the hybrid LSTM-GRU architecture method, while in this study, the authors did not use the hybrid method and only used one hidden layer. Research [11] also uses a larger amount of data, namely 25000 review data, while in this study, only 5874 tweets were used, so it is considerably low. The difference in the amount of data used is proven to have an effect on the performance of the model because the experimental results in research [11] using a polarity review dataset which only amounted to 2000 data, resulted in lower accuracy.

4. Conclusion

After conducting two test scenario, it can be concluded that balancing training data using random oversample and random undersample does not provide higher accuracy than the baseline training data. The LSTM and GRU layers can increase the accuracy of the model created but it takes a longer time than the SimpleRNN layer, in this study the best results were obtained using the GRU layer with 200 neurons with an accuracy of 69.62%. The number of neurons used can affect the accuracy of the model created, where the use of too many neurons can reduce the performance of the model. Model evaluation using 10-fold cross validation can optimize the model created by using the average of each fold result.

Suggestions for further research use more datasets where in this study, only 5874 data and labels are more balanced than the datasets used in this study and test the use of models with more complex hidden layers because the experiments carried out in this study only used one hidden layer. Testing the effect of other parameters such as batch size and number of epochs when training models using hyperparameter tuning can also be used for further research.

References

[1] K. Mouratidis, S. Peters, and B. van Wee, "Transportation technologies, sharing economy, and teleactivities: Implications for built environment and travel," *Transp. Res. Part D Transp. Environ.*, vol. 92, no. February, p. 102716, 2021, doi: 10.1016/j.trd.2021.102716.

[2] M. A. Hairi, "Governance and administrative process of the Light Rail Train project in Palembang, Indonesia," *Public Adm. Policy*, vol. 23, no. 3, pp. 299–313, 2020, doi: 10.1108/PAP-06-2020-0031.

[3] T. Tjahjono, A. Kusuma, N. Tinumbia, and A. Septiawan, "The Indonesia high-speed train traveler preference analysis (case study: Jakarta- Bandung)," *AIP Conf. Proc.*, vol. 2227, no. May, 2020, doi: 10.1063/5.0005009.

[4] E. Sutoyo and A. Almaarif, "Twitter sentiment analysis of the relocation of Indonesia's capital city," *Bull. Electr. Eng. Informatics*, vol. 9, no. 4, pp. 1620–1630, 2020, doi: 10.11591/eei.v9i4.2352.

[5] K. M. Carley, M. Malik, M. Kowalchuck, J. Pfeffer, and P. Landwehr, "Twitter Usage in Indonesia," *SSRN Electron. J.*, 2018, doi: 10.2139/ssrn.2720332.

[6] C. W. Park and D. R. Seo, "Sentiment analysis of Twitter corpus related to artificial intelligence assistants," *2018 5th Int. Conf. Ind. Eng. Appl. ICIEA 2018*, pp. 495–498, 2018, doi: 10.1109/IEA.2018.8387151.

[7] Q. Tul *et al.*, "Sentiment Analysis Using Deep Learning Techniques: A Review," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 6, 2017, doi: 10.14569/ijacsa.2017.080657.

[8] N. C. Dang, M. N. Moreno-García, and F. De la Prieta, "Sentiment analysis based on

deep learning: A comparative study," *Electron.*, vol. 9, no. 3, 2020, doi: 10.3390/electronics9030483.

[9] Z. Jianqiang, G. Xiaolin, and Z. Xuejun, "Deep Convolution Neural Networks for Twitter Sentiment Analysis," *IEEE Access*, vol. 6, pp. 23253–23260, 2018, doi: 10.1109/ACCESS.2017.2776930.

[10] Merinda Lestandy, Abdurrahim Abdurrahim, and Lailis Syafa'ah, "Analisis Sentimen Tweet Vaksin COVID-19 Menggunakan Recurrent Neural Network dan Naïve Bayes," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 4, pp. 802–808, 2021, doi: 10.29207/resti.v5i4.3308.

[11] R. Ni and H. Cao, "Sentiment Analysis based on GloVe and LSTM-GRU," *Chinese Control Conf. CCC*, vol. 2020-July, pp. 7492–7497, 2020, doi: 10.23919/CCC50068.2020.9188578.

[12] S. Yang, X. Yu, and Y. Zhou, "LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example," *Proc. - 2020 Int. Work. Electron. Commun. Artif. Intell. IWECAI 2020*, no. June, pp. 98–101, 2020, doi: 10.1109/IWECAI50956.2020.00027.

[13] M. S. Saputri, R. Mahendra, and M. Adriani, "Emotion Classification on Indonesian Twitter Dataset," *Proc. 2018 Int. Conf. Asian Lang. Process. IALP 2018*, no. January 2019, pp. 90–95, 2019, doi: 10.1109/IALP.2018.8629262.

[14] V. Vinayakumar, M. Alazab, A. Jolfaei, S. Kp, and P. Poornachandran, "Ransomware triage using deep learning: Twitter as a case study," *Proc. - 2019 Cybersecurity Cyberforensics Conf. CCC 2019*, no. Ccc, pp. 67–73, 2019, doi: 10.1109/CCC.2019.000-7.

[15] N. K. Manaswi, "Deep Learning with Applications Using Python," *Deep Learn. with Appl. Using Python*, vol. 1, pp. 115–126, 2018, doi: 10.1007/978-1-4842-3516-4.

[16] Silvin, "Analisis Sentimen Media Twitter Menggunakan Long Short-Term Memory Recurrent Neural Network," 2019.

[17] S. Shofura, S. Suryani M.Si, L. Salma, and S. Harini, "The Effect of Number of Factors and Data on Monthly Weather Classification Performance Using Artificial Neural Networks," *Int. J. Inf. Commun. Technol.*, vol. 7, no. 2, pp. 23–35, 2021, doi: 10.21108/ijoict.v7i2.602.

[18] K. Ramasubramanian and A. Singh, "Machine Learning Using R," *Mach. Learn. Using R*, pp. 667–688, 2019, doi: 10.1007/978-1-4842-4215-5.

[19] H. M. Lynn, S. B. Pan, and P. Kim, "A Deep Bidirectional GRU Network Model for Biometric Electrocardiogram Classification Based on Recurrent Neural Networks," *IEEE Access*, vol. 7, pp. 145395–145405, 2019, doi: 10.1109/ACCESS.2019.2939947.

[20] A. Luque, A. Carrasco, A. Martín, and A. de las Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix," *Pattern Recognit.*, vol. 91, pp. 216–231, 2019, doi: 10.1016/j.patcog.2019.02.023.