khazanah informatika
Jurnal Ilmu Komputer dan Informatika

Sentiment Analysis on Social Media Using Long Short-Term Memory and Word2Vec Feature Expansion Methods with Adam Optimization

Sanabila Khoirunnisa [1], Erwin Budi Setiawan [2*]
erwinbudisetiawan@telkomuniversity.ac.id
[1,2]School Of Computing, Telkom University
Bandung

Abstract Twitter is one of Indonesia's most popular social media, so it has many users. The intensity of Twitter use can be used to carry out sentiment analysis related to topics being widely discussed, especially regarding the 2024 Indonesian presidential election. To understand public views, public opinion is divided from text data into positive and negative polarities to measure public sentiment. The classification model uses Long Short-Term Memory (LSTM) for feature extraction, utilizing TF-IDF. In addition, this model also combines Word2Vec based on the Indonews corpus, which contains 142,545 articles for feature expansion. This model is further optimized using the Adam optimization technique to improve accuracy. By using a dataset of 37,391 data, the results of this research obtained an accuracy score of 83.04% and an f1 score of 82.62%. This is an increase in accuracy of 11.22%; for the f1 score, it was a 10.92% increase from the baseline. This indicates that the classification model using Long Short-Term Memory (LSTM) with the application of TF-IDF as feature extraction, Word2Vec as feature expansion, and Adam optimization successfully produced optimal sentiment predictions regarding the 2024 Indonesian Presidential Election.

Keywords: Sentiment Analysis, LSTM, TF-IDF, Word2Vec, Adam, Twiter

Article info: *submitted January 12, 2024, revised October 13, 2024, accepted April 30, 2025*

1. Introduction

In the current digital era, information technology is developing quite rapidly. With the development of information, society can communicate more quickly and efficiently by using social media. Social media is an online media that users can use easily to participate, share, and create content like social networks. Wrong One social networking site that is often used is Twitter. Based on statistical data from Statista [1], Indonesia is ranked fifth in the world, with the number of users reaching 24 million users. With so many Twitter users, this platform has become one of the fastest digital media disseminating information.

Twitter is a social media platform that people can use freely to express their opinions. People can use Twitter to send messages or what are usually called tweets. One of the topics that is being widely discussed is tweets related to the 2024 Indonesian presidential candidates. The existence of tweets causes people to have views on trending topics on Twitter. However, the more information a user receives, the more tweets there will be, which sometimes have ambiguous meanings, thus affecting understanding of the tweet. This has led to various proactive and counter-responses from Twitter users. Emotions in textual form are called sentiment analysis[2].

Sentiment analysis or opinion mining is a branch of research text mining. Sentiment analysis can be used to analyze opinions, emotions, and written assessments on specific topics using processing techniques in natural language[3]. Sentiment analysis is related to the polarity identification sentiment of a text, which states whether the text is positive, neutral, or negative. Identification of hidden sentiments refers to emotions in between anger, happiness, and annoyance [4]. There are several branches of sentiment analysis, one of which is aspect-based sentiment analysis (ABSA). This analysis makes it possible to obtain more detailed information based on each aspect[5]. There are many methods for conducting sentiment analysis for a topic.

One of the sentiment analysis processing methods that can be used is deep learning. Deep learning allows computers to learn complex concepts by breaking them down into more straightforward concepts to gain a broader understanding[6]. In recent years, there have been significant advances in deep learning technology in the field of natural language processing, and many online text sentiment analysis methods based on deep learning have been proposed[7].

One of the classification models of deep learning is Long Short-Term Memory (LSTM), which is the development of the Recurrent Neural Network (RNN) algorithm. LSTM Models complement the shortcomings of the RNN model, which cannot predict words stored for an extended period [8]. Research written by M.Z proves this. Rahman et al. [9] 2021 compared RNN and LSTM, showing that LSTM could provide better results than

RNN. Strengthened by the accuracy results on LSTM as much as 81%, while on RNN as much as 71%, which means the model produced by The LSTM method has accurate results 10% better than RNN method. To produce accurate predictions, optimization of the LSTM is needed.

Stochastic gradient-based optimization is of great importance in various fields of science and engineering. For example, the adaptive method of moment estimation (Adam), an efficient stochastic method in optimization, requires only first-order gradients with low memory requirements. This method calculates the adaptive learning rate individually for various estimated parameters in advance and also considers the gradient's second-order moments [10]. Combining two popular methods, namely Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp), makes Adam more predictive and more accurate for LSTM models than other optimizers[11]. This was proven in research conducted by Irfan D et al. [12] in 2022, which compared the optimization of Stochastic Gradient Descent (SGD), ADADELTA, and Adam. The research results show that Adam optimization produces the best score of 0.8350, ADADELTA optimization produces the best score of 0.8225, and SGD optimization produces the best score of 0.6375 so Adam optimization produces the highest accuracy compared to the SGD and ADADELTA optimization methods.

This research's main contribution is optimizing the Long Short-Term Memory method using Adam optimization and expanding the Word2Vec feature in sentiment analysis related to the 2024 Indonesian presidential election on Twitter. Based on the author's knowledge, no research has been conducted on this research, and this has the potential to increase the accuracy value in sentiment analysis. To produce a model with the best performance, researchers applied several methods such as LSTM, TF-IDF as a feature extraction method, Word2Vec as a feature expansion method, and Adam optimization as an optimization technique for the model. Various scenarios will be tried in this research, such as choosing the data division ratio, determining the best max feature in feature extraction, utilizing Word2Vec by selecting a corpus from various corpora, and optimizing the model with Adam's learning rate optimization.

The structure of this paper consists of the following sections. Section 2 explains the methodology applied in the experiments. Section 3 presents the results of the research conducted, section 4 contains a discussion of the research, and Section 5 provides the conclusions of this research.

Table 1. Keyword crawling dataset

Keyword	#Data
Anies Baswedan	10,434
Ganjar Pranowo	8,027
Capres	7,296
Calon Presiden	6,972
Prabowo Subianto	4,662
Total	37,391

2. Methods

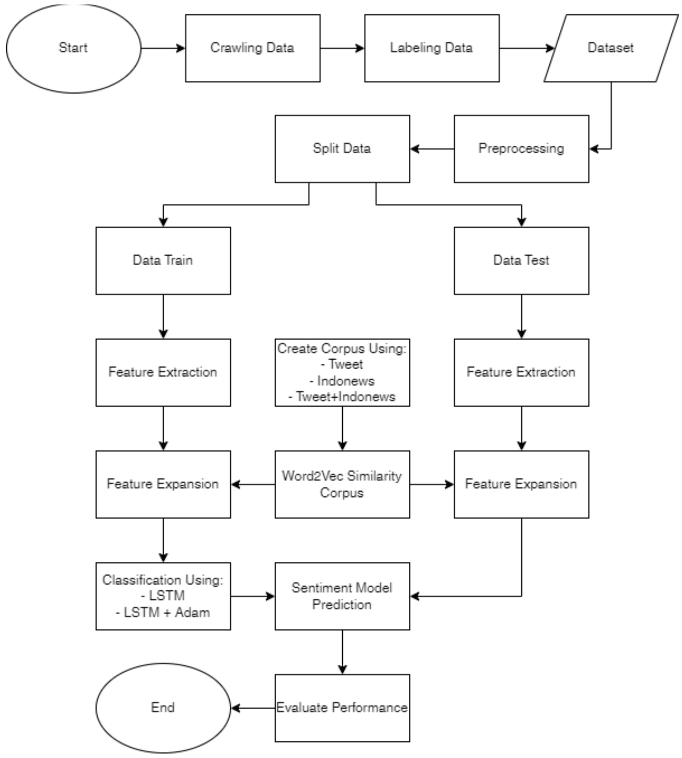

Figure 1 LSTM Design System for Analysis Sentiment

This research has several system design processes that are built, as shown in Figure 1. The system begins with a data crawling process, namely, the process of collecting data from Twitter. Then, the data for each text is manually labeled, either positive or negative. Then, proceed to the preprocessing stage so that the data is clean and ready for the next stage, namely, feature extraction using the Term Frequency - Inverse Adam (Adaptive Moment Estimation) optimizer, then performance evaluation.

1. Data Crawling

The dataset used in this research is data in Indonesian and taken from Twitter social media using several keywords such as "Anies Baswedan," "Calon Presiden," "Capres," "Prabowo Subianto," "Ganjar Pranowo," as shown in Table 1. Data collection was carried out using the crawling method and the Application Program Interface (API) provided by Twitter, assisted by the tweet-harvest library with the Python programming language. The results of the data crawling process obtained 37,391 data, and the dataset was saved in Comma Separated Value (CSV) format.

2. Labeling Data

Labeling of previously collected data is carried out, and data labeling is done manually by taking the most votes based on the opinions of 3 people. Text data will be labeled "1" for positive and "-1" for negative. Data that has been labeled based on sentiment on textual data can be seen in Table 2.

Table 2. Number of Sentiment Labels from Entire Dataset

Label	quantity
Negative	15,525
Positive	21,866

3. Preprocessing

The data that has been collected will then enter the next stage of initial processing. This stage is carried out to prepare clean and high-quality data so that the data used for the classification stage can be processed efficiently. In this research, the stages of preprocessing are divided into several stages, namely Data Cleaning, Case Folding, Tokenizing, Data Normalization, StopWord Removal, and Stemming.

The following are the stages in Preprocessing:

A) Data Cleaning

This stage is a process of cleaning elements that are not needed in the analysis, such as punctuation, emoji, site addresses, user names, links, and hashtags.

B) Case Folding

At this stage, the process of replacing existing words is carried out with capital letters or capital letters that become lowercase so that the data becomes consistent and makes the analysis process easier.

C) Tokenizing

This stage is the process of separating sentences into word-by-word called tokens.

D) Data Normalization

At this stage is the process of changing the writing of words into a standard format in accordance with the general guidelines for Indonesian spelling (PUEBI) to make it easier to understand so as to improve the quality of analysis.

E) StopWord Removal

At this stage, it is a process of deleting common words that often appear in language that does not provide much important information in the analysis, so it can improve efficiency and analyze accuracy by focusing on more informative words.

F) Stemming

At this stage, the process of removing the word affix is carried out at the suffix, prefix, or a combination of both so that it produces relatively the same basic words, which can reduce complexity in data processing.

4. Feature Extraction With TF-IDF

After the preprocessing stage, tweets will enter feature extraction, which is the process of taking characteristics from tweets and producing an overview of the characteristics of the tweet[13]. Using the Term Frequency-Inverse Document Frequency (TF-IDF) method can give weight to each word by assessing each word based on its relevance to the content of the entire document. Salton and Buckley were the originators of the TF-IDF algorithm in 1988 and used it for information retrieval purposes, which was then used as an algorithm with a feature weighting method in text mining[14]. The formula for TF-IDF can be seen from the following equation:

$$\mathbf{TF - IDF = TF \times IDF} \qquad (1)$$

This formula can be translated into the Frequency Term of a particular term (t), which is calculated from the number of occurrences of a term in a document with the total number of words in the document. IDF (Inverse Document Frequency) is useful for calculating the importance of a term, where N is the number of documents and DF is the number of documents containing the term t[15].

$$\boldsymbol{IDF(t) = \log\left(\frac{N}{DF}\right)} \qquad (2)$$

5. Feature Expansion With Word2Vec

Word2Vec, developed by Mikolov, excels in representing vectors that capture the syntax and semantic meaning of natural language. This algorithm groups similar words with identical vectors. Its architecture uses neural networks with text bodies as input and generates a vector space as output, creating low-dimensional word vectors that carry semantic meaning. There are two types of Word2Vec architectures, namely Skip-Gram and Continuous Bag of Words (CBOW) models[16].

CBOW works by predicting the probability of words given the context, and the context can be a close word or a group of adjacent words. Meanwhile, the skip-gram model works upside down by trying to predict the context for the words around it in a sentence[17]. Both models produce vector representations for each word that can be used in various natural language processing applications (NLP). The Word2VEC method encourages corpus as an input and produces an output in the form of vectors. Corpus was made a gensim library assistance in Python. Corpus used is tweet, IndoNews with a lot of data, 142,545 articles, which contain news in Indonesia such as Liputan 6 CNN Indonesia, Detik.com, Republika, Kompas, and Tempo[18], as well as using a combination of indonews corpus and tweets.

6. Long Short-Term Memory Classification (LSTM)

Long Short-Term Memory (LSTM) is a variant of Recurrent Neural Network (RNN). RNN has the problem of not being able to perform long-term information learning due to the exploding and vanishing gradient problem. LSTM can avoid this problem by replacing RNN nodes in hidden layers with LSTM cells designed to deviate from prior information[8]. In standard RNNs, these iterative modules usually have a simple structure. However, the loop module for LSTM is more complicated. Instead of having one neural network layer, four layers interact in special ways. Additionally, LSTM has two states: hidden state and cell state[19]. Four layers are interconnected in a special way in LSTM. They are input gates, which are responsible for how much new information is stored in the LSTM memory cell. Forget gates are useful for controlling how much information previously stored in the memory cell must be forgotten or deleted, output gates function to regulate how much information is taken from the memory cell LSTM to produce output at certain time steps, and memory cells are an important component in it LSTM is useful for storing information from previous input and retaining this information during the training process. The following is an overview of the LSTM network, which can be seen in Figure 2.

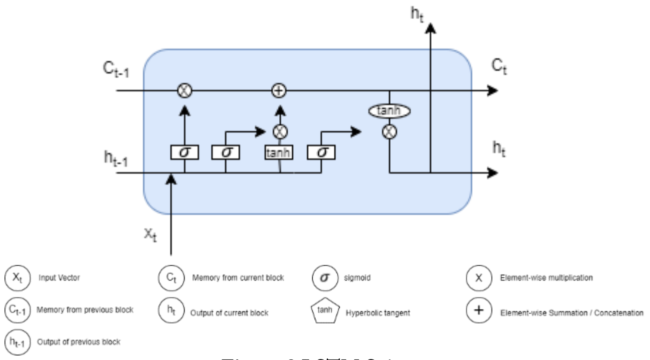

Figure 2 LSTM Scheme

Using LSTM, there are several steps, first deciding what information will be deleted or discarded from the state cell using a sigmoid function or layer called a "forget gate." This gate takes ht-1 and xt then returns a number between 0 and 1, where the value "0" means the data is deleted or discarded, and the value "1" means the data is saved. Following is the calculation of the forget gate value:

$$\boldsymbol{f_t = \sigma(W_f[h_{t-1,}x_t] + b_f)} \quad (3)$$

Then the LSTM decides which new information will be stored in the cell state, which has two steps: first, the sigmoid function or layer called "Input gate" decides which value the LSTM updates; second, the tanh function or layer will create a new candidate vector $\widetilde{C}_t$ the results will be added to the memory cell.

$$\boldsymbol{i_t = \sigma(W_i.[h_{t-1,}x_t] + b_i)} \quad (4)$$

$$\boldsymbol{\widetilde{C}_t = \tanh(W_c.[h_{t-1}, x_t] + b_c)}$$

Next, update the old cell state Ct-1 to the new cell state Ct by paying attention to the forget gate. You can delete or update the thing you want to throw away.

$$\boldsymbol{C_t = f_t \times C_{t-1} + i_t \times \widetilde{C}_t} \quad (5)$$

In the final process, LSTM runs a sigmoid layer, which decides which part of the cell state will be output or called the output gate. Next, the LSTM will output a separate part by entering the cell state through the tanh layer and then multiplying it by the gate output.

$$\boldsymbol{O_t = \sigma(W_o.[h_{t-1}, x_t] + b_o)} \quad (6)$$

$$\boldsymbol{h_t = O_t \times \tanh(C_t)}$$

7. Adam Optimizer

Diederik Kingma and Jimmy Ba first introduced this method in a paper entitled "Adam: A Method for Stochastic Optimization" in 2014. Adam (Adaptive Moment Estimation) optimizer is an optimization algorithm that is widely used in deep learning because it has an independent learning rate and a relatively fast convergence speed[20]. The Adam Method is designed to combine the advantages of two popular methods, namely AdaGrad and RMSProp. Some advantages of Adam are that the magnitude of parameter updates is not affected by gradient rescaling, its step size is approximately limited by the step size hyperparameter, it does not require a stable objective, it works with sparse gradients, and naturally performs step size reduction[10].

8. Performance Measurement

A major challenge in classifying imbalanced data is its potentially significant impact on the value and interpretation of performance measurements. To overcome this problem, it is important to identify appropriate performance measures[21].

Confusion matrix is a method for evaluating the performance of a classifier. The confusion matrix displays the classification results in a table with four possible results, namely True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN)[22].

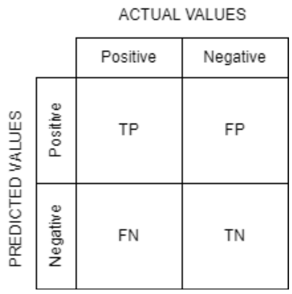

Figure 3 Confusion Matrix

From the confusion matrix, performance values can be calculated using the formulas for accuracy, precision, recall, and F1 score.

A) Accuracy

Accuracy is the ratio of correctly classified data to the total classification data carried out. The accuracy value calculation can be seen from the following equation:

$$\textbf{Accuracy} = \frac{\boldsymbol{TP + TN}}{\boldsymbol{TP + FP + TN + FN}} \quad (7)$$

B) Precision

Precision is a comparison between the amount of data that is correctly classified as a positive class and all data that is classified as a positive class. The calculation of the precision value can be seen from the following equation:

$$\textbf{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FP}} \quad (8)$$

C) Recall

Recall is the proportion of data that is accurately identified as a positive class compared to the total data. The calculation of the recall value can be seen from the following equation:

$$\textbf{recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FN}} \quad (9)$$

D) F1 Score

The F1 Score is the result of calculating the harmonic average between precision and recall, providing a balanced value between these two values. The calculation of the F1 score value can be seen from the following equation:

$$\boldsymbol{F1\ score} = \frac{\mathbf{2} \times (\boldsymbol{precision} \times \boldsymbol{recall})}{(\boldsymbol{precision} + \boldsymbol{recall})} \tag{10}$$

3. Result

In this research, four test scenarios were carried out to evaluate model performance. The first scenario performs a baseline evaluation by applying different ratios of sharing test data and training data. In the second scenario, TF-IDF feature extraction is combined with baseline, with the aim of evaluating its impact on classification performance and relevance of words. The third scenario combines Baseline, TF-IDF, with Word2Vec expansion, utilizing a corpus. The fourth scenario focuses on optimizing the method using Adam's optimization and optimal learning rate. Each scenario uses the Long Short-Term Memory (LSTM) classification model and runs each scenario five times. Test results from each scenario include average accuracy values and F1-Score.

A) First Scenario (Baseline)

This scenario is carried out to find a comparison of training data and test data to determine the optimal data division that has the highest accuracy. Different data split scenarios are applied with ratios of 90:10, 80:20, and 70:30. Table 3 shows the results obtained from scenario 1.

Table 3. Testing Results of Scenario 1

Test Size	Accuracy (%)	F1-Score (%)
90:10	**71.82**	**71.70**
80:20	71.15	69.56
70:30	71.58	71.07

The LSTM model obtained the best results with an accuracy value of 71.82% and an F1 score of 71.70% with data division 90:10. This ratio value is set to be the highest value in scenario stage one (baseline).

B) Second scenario (the best max feature)

In the second scenario, after achieving the highest accuracy performance in the first scenario (baseline), the next step involves using Term Frequency-Inverse Document Frequency (TF-IDF) as an extraction parameter. Comparisons were made to find optimal feature values, with the number of features as comparison, namely 500, 1000, 3000, 5000, and 10000, using the training data set size from scenario 1. The test results for the second scenario are documented in Table 4.

It was found that the best maximum features were 5000, resulting in an accuracy of 82.25% and an F1 score of 81.72%. This maximum number of features will be used in the next scenario.

Table 4. Testing Results of Scenario 2

Max Feature	Accuracy (%)	F1-Score (%)
500	78.46 (+6.64)	77.07 (+5.37)
1000	80.18 (+8.36)	79.69 (+7.99)
3000	81.82 (+10.00)	80.66 (+8.96)
5000	**82.25** **(+10.43)**	**81.72** **(+10.02)**
10000	82.03 (+10.21)	81.41 (+9.71)

C) Third Scenario (Feature Expansion)

In the third scenario stage, adding feature expansion using Word2Vec from scenario 2. The corpus used to implement the expansion of the Word2Vec feature has been previously trained using Indonews and Tweet+Indonews to look for similarities. Scenario 3 compares the similarity scores of words from the trained corpus: top 1, top 2, top 3, top 5, and top 10. The results of the third scenario can be seen in Table 5 and Table 6.

Table 5. Testing Accuracy Results of Scenario 3

	Accuracy (%)		
Top(n)	Tweet	Indonesia	Tweet+Indonews
1	81.55 (+9.73)	82.51 (+10.69)	82.59 (+10.77)
2	81.21 (+9.39)	**82.91** **(+11.09)**	81.62 (+9.8)
3	81.29 (+9.47)	82.19 (+10.37)	82.11 (+10.29)
5	80.13 (+8.31)	82.70 (+10.88)	81.79 (+9.97)
10	79.12 (+7.30)	82.51 (+10.69)	73.58 (+1.76)

Table 6. Testing F1-Score Results of Scenario 3

	F1-Score (%)		
Top(n)	Tweet	Indonesia	Tweet+Indonews
1	81.08 (+9.38)	82.14 (+10.44)	82.15 (+10.45)
2	80.83 (+9.13)	**82.57** **(+10.87)**	81.24 (+9.54)
3	80.93 (+9.23)	81.76 (+10.06)	81.66 (+9.96)
5	79.68 (+7.98)	82.33 (+10.63)	81.50 (+9.80)
10	78.53 (+6.83)	82.03 (+10.33)	71.53 (-0.17)

The results show that the LSTM model produces optimal accuracy using the Indonews corpus of 82.91% and F1 score of 82.57%, which can combine the top 2 words.

D) Fourth Scenario (Optimizer)

In the fourth scenario, additional Adam optimization is carried out, which is a continuation of the third scenario. In this scenario, there are two learning speed values tested: the default value of 0.01 and the optimal value of 1×10^5 or 0.00001. The results of the fourth scenario can be seen in Table 7.

Table 7. Testing Results of Scenario 4

Top (n)	Default Learning Rate		Optimal Learning Rate	
	Accuracy (%)	F1-Score (%)	Accuracy (%)	F1-Score (%)
2	81.08 (+9.26)	80.41 (+8.71)	**83.04 (+11.22)**	**82.62 (+10.92)**

Learning rate is a parameter that determines how much the model is updated during training. The optimal learning rate value is very important because it can affect the speed of algorithm convergence and the quality of the final results. In the context of research results, using Adam optimization with a learning rate $of\ 1 \times 10^5$ or 0.00001 is proven to provide optimal performance, as documented in Table 7. The fourth scenario test results show that the accuracy reaches 83.04%, and the F1-Score is 82.62%.

4. Discussion

In this research, four test scenarios were implemented using the Long Short-Term Memory (LSTM) model by applying various techniques, including Term Frequency - Inverse Document Frequency (TF-IDF) feature extraction, Word2Vec feature expansion, and optimization with Adam (Adaptive Moment Estimation). In the first scenario (baseline), the results of testing training data and test data with a ratio of 90:10, 80:20, and 70:30 show that the best comparison is 90% training data and 10% test data, resulting in an accuracy of 71.82% and a score F1 71.70%. In the second scenario, we performed TF-IDF feature extraction using a maximum of 5000 features, which resulted in a significant increase in accuracy to 82.25% and F1 score to 81.72%. The third scenario involves expanding the features by applying the Word2Vec model and training data using Corpus Indonews. The results show an accuracy of 82.91% and an F1 score of 82.57%. In the last scenario, namely the fourth scenario, we apply optimization using Adam. In this test, the learning rate was 1×10^5 or 0.00001, resulting in an increase in accuracy to 83.04% and an F1 score of 82.62%. Performance improvements in each scenario can be observed in more detail in Figure 4 and Figure 5.

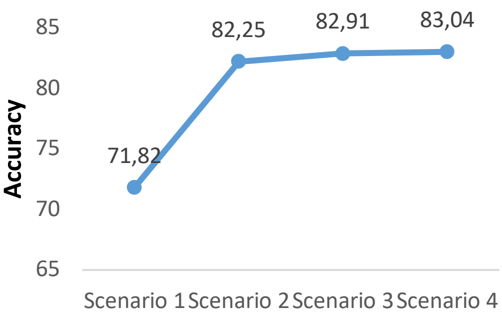

Figure 4 Graph of Performance Improvement Accuracy from each scenario

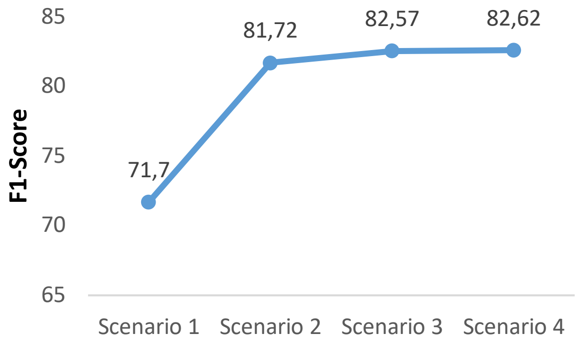

Figure 5 Graph of Performance Improvement F1-Score from each scenario

5. Conclusion

This paper evaluates the application of Adam optimization by utilizing the learning rate to achieve optimal accuracy using the Long-Short Term Memory (LSTM) classification method, feature extraction using TF-IDF with 5000 maximum features, and Word2Vec with the Indonews corpus with 142,545 articles as an extension features through four test scenarios. The dataset comes from Twitter with a total of 37,391 data, with 15,525 data labeled negative and 21,866 data labeled positive. When used together, LSTM achieves the highest accuracy, namely 83.04%, with an F1-Score of 82.62%. These results show that using the optimal learning rate on Adam can produce a significant increase in accuracy, namely 11.22%, and an F1 score of 10.92% from the baseline.

Reference

[1] "Leading countries based on number of X (formerly Twitter) users as of January 2023 (in millions)," 2023. [Online]. Available: https://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/1/4Source:https://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/

[2] I. Zulfa and E. Winarko, "Sentimen Analisis Tweet Berbahasa Indonesia dengan Deep Belief Network," *IJCCS*, vol. 11, no. 2, pp. 187–198, 2017, [Online]. Available: www.search.twitter.com

[3] T. T. Thet, J. C. Na, and C. S. G. Khoo, "Aspect-based sentiment analysis of movie reviews on discussion boards," *J Inf Sci*, vol. 36, no. 6, pp. 823–848, Dec. 2010, doi: 10.1177/0165551510388123.

[4] M. Bouazizi and T. Ohtsuki, "Multi-class sentiment analysis on Twitter: Classification performance and challenges," *Big Data Mining and Analytics*, vol. 2, no. 3, pp. 181–194, Sep. 2019, doi: 10.26599/BDMA.2019.9020002.

[5] H. Gunawan Sulistio and A. Handojo, "Aspect-Based Sentiment Analysis pada Ulasan E-Commerce dengan Metode Support Vector Machine untuk Mendapatkan Informasi Sentimen dari Beberapa Aspek."

[6] North Eastern Hill University. Department of Biomedical Engineering, Institute of Electrical and Electronics Engineers. Kolkata Section, IEEE Industry Applications Society, and Institute of Electrical and Electronics Engineers, *International*

Conference on Computational Performance Evaluation : ComPE 2020 online conference : 2nd-4th July 2020.

[7] B. Liu, "Text sentiment analysis based on CBOW model and deep learning in big data environment," *J Ambient Intell Humaniz Comput*, vol. 11, no. 2, pp. 451–458, Feb. 2020, doi: 10.1007/s12652-018-1095-6.

[8] 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, doi: 10.22146/ijccs.41236.

[9] M. Z. Rahman, Y. A. Sari, and N. Yudistira, "Analisis Sentimen Tweet COVID-19 menggunakan Word Embedding dan Metode Long Short-Term Memory (LSTM)," 2021. [Online]. Available: http://j-ptiik.ub.ac.id

[10] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Dec. 2014, [Online]. Available: http://arxiv.org/abs/1412.6980

[11] Institute of Electrical and Electronics Engineers. Beijing Section and Institute of Electrical and Electronics Engineers, *ICSESS 2018 : proceedings of 2018 IEEE 9th International Conference on Software Engineering and Service Science : November 23-25,2018, China Hall of Science and Technology, Beijing, China*.

[12] D. Irfan, R. Rosnelly, M. Wahyuni, J. T. Samudra, and A. Rangga, "Perbandingan Optimasi SGD, ADADELTA, Dan ADAM Dalam Klasifikasi Hydrangea Menggunakan CNN," 2022. [Online]. Available: http://jurnal.goretanpena.com/index.php/JSSR

[13] E. B. Setiawan, D. H. Widyantoro, and K. Surendro, "Feature expansion for sentiment analysis in Twitter," in *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, Institute of Advanced Engineering and Science, Oct. 2018, pp. 509–513. doi: 10.1109/EECSI.2018.8752851.

[14] G. Berliana and S. T. Shaufiah, "Klasifikasi Posting Tweet mengenai Kebijakan Pemerintah Menggunakan Naive Bayesian Classification."

[15] R. Ahuja, A. Chug, S. Kohli, S. Gupta, and P. Ahuja, "The impact of features extraction on the sentiment analysis," in *Procedia Computer Science*, Elsevier B.V., 2019, pp. 341–348. doi: 10.1016/j.procs.2019.05.008.

[16] R. P. Nawangsari, R. Kusumaningrum, and A. Wibowo, "Word2vec for Indonesian sentiment analysis towards hotel reviews: An evaluation study," in *Procedia Computer Science*, Elsevier B.V., 2019, pp. 360–366. doi: 10.1016/j.procs.2019.08.178.

[17] Manipal University Jaipur. School of Computing and Information Technology and Institute of Electrical and Electronics Engineers, *2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT) : Manipal University, Jaipur, Sep. 28-29, 2019*.

[18] H. R. Alhakiem and E. B. Setiawan, "Aspect-Bas1ed Sentiment Analysis on Twitter Using Logistic Regression with FastText Feature Expansion," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 5, pp. 840–846, Nov. 2022, doi: 10.29207/resti.v6i5.4429.

[19] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdiscip Rev Data Min Knowl Discov*, vol. 8, no. 4, Jul. 2018, doi: 10.1002/widm.1253.

[20] X. Liu, "Research on the forecast of coal price based on LSTM with improved Adam optimizer," *J Phys Conf Ser*, vol. 1941, no. 1, p. 012069, Jun. 2021, doi: 10.1088/1742-6596/1941/1/012069.

[21] 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, Jul. 2019, doi: 10.1016/j.patcog.2019.02.023.

[22] D. Krstinić, M. Braović, L. Šerić, and D. Božić-Štulić, "Multi-label Classifier Performance Evaluation with Confusion Matrix," Academy and Industry Research Collaboration Center (AIRCC), Jun. 2020, pp. 01–14. doi: 10.5121/csit.2020.100801.