ILKOM Jurnal Ilmiah Vol. 15, No. 3, December 2023, pp. 455-464
Accredited 2nd by RISTEKBRIN No. 200/M/KPT/2020; E-ISSN 2548-7779 | P-ISSN 2087-1716

Research Article

Comparative Analysis of Long Short-Term Memory Architecture for Text Classification

Moh. Fajar Abdillah [a,1]; **Kusnawi** [a,2,*]
[a] *Universitas Amikom Yogyakarta, Jl. Ring Road Utara, Yogyakarta and 55281, Indonesia*
[1] *moh.abdillah@students.amikom.ac.id;* [2] *khusnawi@amikom.ac.id*
** Corresponding author*

Article history: Received September 03, 2023; Revised November 13, 2023; Accepted December 01, 2023; Available online December 20, 2023

Abstract

Text classification which is a part of NLP is a grouping of objects in the form of text based on certain characteristics that show similarities between one document and another. One of methods used in text classification is LSTM. The performance of the LSTM method itself is influenced by several things such as datasets, architecture, and tools used to classify text. On this occasion, researchers analyse the effect of the number of layers in the LSTM architecture on the performance generated by the LSTM method. This research uses IMDB movie reviews data with a total of 50,000 data. The data consists of positive, negative data and there is data that does not yet have a label. IMDB Movie Reviews data go through several stages as follows: Data collection, data pre-processing, conversion to numerical format, text embedding using the pre-trained word embedding model: Fastext, train and test classification model using LSTM, finally validate and test the model so that the results are obtained from the stages of this research. The results of this study show that the one-layer LSTM architecture has the best accuracy compared to two-layer and three-layer LSTM with training accuracy and testing accuracy of one-layer LSTM which are 0.856 and 0.867. While the training accuracy and testing accuracy on two-layer LSTM are 0.846 and 0.854, the training accuracy and testing accuracy on three layers are 0.848 and 864.

Keywords: FastText; LSTM; NLP; Text Classification.

Introduction

Natural Language Processing or commonly abbreviated as NLP is one part of computer science that deals with human and computer interaction approaches. NLP itself serves to understand human language to computers [1], [2]. One example of the NLP approach itself is text classification [3], [4]. Classification is a form of pattern recognition or certain characteristics that is grouped into several parts. The object of classification itself varies, it can be text, video, image, or sound [5]. While text classification itself is a grouping of objects in the form of text based on certain characteristics that show similarities between one document and another [6]. There are several methods used for text classification, one of the methods used to classify text is Long-Short Term Memory or often abbreviated as LSTM.

LSTM is a development of Recurrent Neural Network architecture or often abbreviated to RNN [7]. LSTM introduced by Hochreiter and Schmidhuber designed a temporal sequense model with long-range dependencies [8]. Solving problems in text classification, the LSTM architecture can use single layer or multi-layer [9]. Multiple-layer LSTM is superior to single-layer LSTM [8], resulting the accuracy validation comparison between single-layer LSTM and multi-layer LSTM of 0.7243 and 0.8060 consecutively. When comparing single layer LSTM with a normalized root mean squared error (nRMSE) of 10.7% and 5-layer LSTM with a nRMSE of 11.4%, research [10] finds that simple layers perform similarly to more complex layers, while in research [9], three layers have the best results with a comparison of testing accuracy, namely single layer LSTM 0.881, double layer LSTM 0.902, and triple layer LSTM 0.909, which means that more complex layers have better results than simple layers. With an accuracy ratio of 90.75 for double layer LSTM and 87.98 for single layer LSTM, research [11] claims that double layer LSTM is superior than single layer LSTM. The difference in the use of this layer affects the results of text classification. Therefore, in this research the author identifies which layer has the best results between one layer, two layers, or three layers.

Method

This research aims to find out which layer has the best results between layers one, two and three in the LSTM architecture. This research has several stages to achieve the results. The stages consist of: data collection, data preprocessing, conversion to numerical format, embedding text using pre-trained word embedding model: fastText from research [12], training and testing the classification model using LSTM, and finally validating and testing the model so that the results are obtained. The data used in this research is IMDB Movie Reviews data from research [13]. The amount of data used is 50,000 data. The data consists of positive and negative data, meanwhile there is data that does not yet have a label. **Figure 1** is an overview of the overall stages of this research.

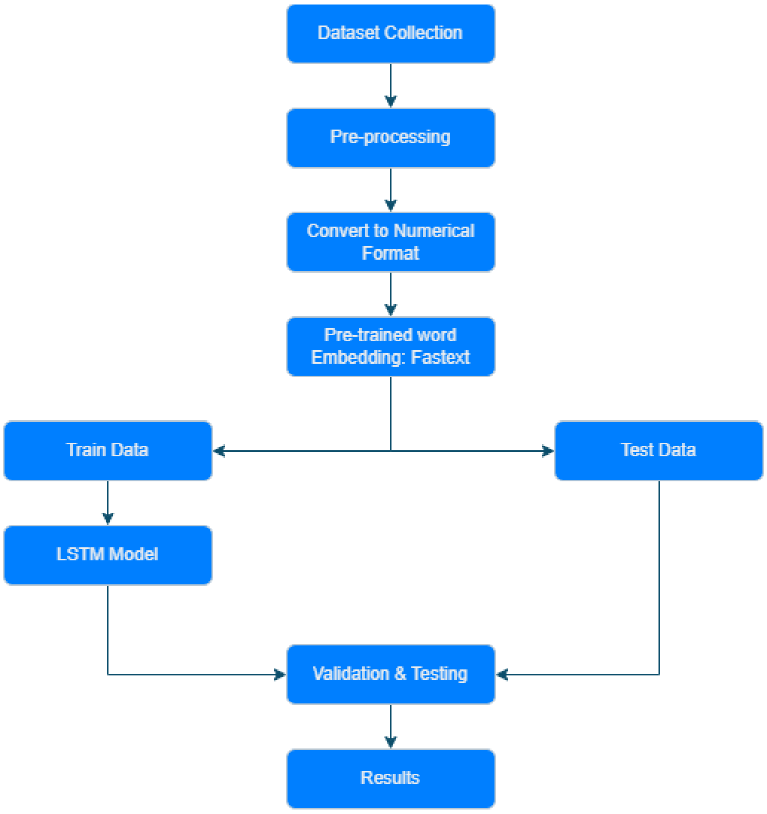


Figure 1. Research flow

A. Data Collection

This stage is a stage to collect or retrieve data that is later used in this study. This research uses the IMDB Movie Reviews dataset taken from research [13]. The amount of data used is 50,000 data. The data contains reviews of many films and consists of positive and negative data, meanwhile there is data that does not yet have a label.

B. Data Preprocessing

Data preprocessing is a very important process in this research because data preprocessing is a stage to ensure the data used for analysis to the next stage is clean so that it is ready to be analyzed [14]–[16]. The data goes through several processes in this stage to produce clean data. The process includes case folding, filtering, tokenization, stop word removal, and stemming.

- *Case Folding*

  Case folding is the process of converting all letters into lowercase or uppercase letters, in this study researchers converted all letters into lowercase letters to make it easier when the text is analyzed for consistency [14], [17].

- *Filtering*

  Filtering is the process of filtering words to remove unwanted values such as non-alphabetic characters or punctuation marks [15], [17], [18].

- *Tokenization*

  Tokenization is the next process performed after the filtering process. Tokenization aims to separate the text into smaller parts [17], [19].

- *Stop Word Removal*

  Stop word removal is a stage to remove stop words so as not to cause the analysis process to become more complex [20].

- *Stemming*

  Stemming is a process that is carried out after removing stop words. Stemming itself aims to get the original word from the text [20]. After the stemming process, clean data is generated and ready to be analyzed to the next stage.

An example of the data preprocessing stage is in **Table 1**.

Table 1. Data Preprocessing

Process	Result
Example Sentence	The first thing that struck me about Oz was its brutality and unflinching scenes of violence, which set in right from the word GO
Case Folding	the first thing that struck me about oz was its brutality and unflinching scenes of violence, which set in right from the word go
Filtering	the first thing that struck me about oz was its brutality and unflinching scenes of violence which set in right from the word go
Tokenization	the,first,thing,that,struck,me,about,oz,was,its,brutality,and,unflinching,scenes,of,violence,which,set,in,r ight,from,the,word,go
Stop Word Removal	first,thing,struck,oz,brutality,unflinching,scenes,violence,set,right,word,go
Stemming	first thing struck oz brutal unflinch scene violenc set right word go

C. Conversion to Numerical Format

The next stage is converting words into numbers. After the data is clean, the next step is to perform the sentence conversion process which is the process of converting words into tokens after being converted into tokens into a group of arrays whose contents are numbers [18]. Next is the provision of padding in the sentence. Giving padding aims to make the sentence have the same pang. Determining the longest sentence or maximum words in a sentence is a way of determining padding in sentences [17]. The padding in this study is based on the maximum length of words in the sentence contained in the dataset, which is 123. An example of sentence conversion is in **Table 2**.

Table 2. Sentence Conversion

Process	Result
Word Token	[first, thing, struck, oz, brutal, unflinch, scene, violenc, set, right, word, go,]
Sequence	[670, 22, 248, 1689, 1826, 1843, 1049, 772, 49, 451, 304, 9]
Padding	[670, 22, 248, 1689, 1826, 1843, 1049, 772, 49, 451, 304, 9]

D. Pre- Trained Word Embedding

Pre-Trained word embedding is a vectorization process to convert words into vectors where the vector represents words that have been changed before [9], [18], [21], [22]. This process is needed because it aims to convert words into vectors so that they can be processed by the model [23]. The dimension of the resulting vector can be determined. The accuracy of the word vector representation is influenced by the length of the sentence. The longer the sentence, the more accurate the resulting representation [18]. The maximum number of words in this study is 123 and the dimensions used have a dimension length of 300. This study uses the FastText framework because it is more efficient and faster than Glove and Word2vec [18], [23]. FastText is a library owned and managed by fecabook that functions to represent words efficiently and support the text classification process [23]. After this process is carried out, 26253 words are found in the wiki vocab and 42760 new words are found in the dataset.

E. Developing Model

The model used in this research is the LSTM method, a popular algorithm among NLP researchers because of its superior and efficient ability when modelling and learning from sequential data [24]. The ability of LSTM to solve various problems or tasks in natural language preprocessing has been proven to be sophisticated and effective, such as in text classification [25], tagging problems [26], [27], and sequence to sequence prediction [28]. LSTM is an extension of the RNN architecture introduced by Hochreiter and Schmidhuber to design temporal sequence models with long-range dependencies [8]. LSTM is designed to store and utilize previously obtained values and information for a certain period [29]. LSTM has 3 gate layers, namely input gate, forgate gate, and output gate [30]. Input gate serves to regulate the entry of new information into memory, forgate gate serves to regulate how long the value or information is in memory. The value or information will be forgotten when the result of the forgate gate is close to zero, but if the value or information is close to 1 it will be retained. The output gate is responsible for setting the size of the value that will be stored in memory [29]. This research also uses softmax to determine the information or value that will come out of the forgate gate. SoftMax is a layer that functions to decide or classify the activity information that will come out of the LSTM [31], [32].

The value of each LSTM gate is calculated with the equation below which takes reference from [18].

$$i_t = \sigma(U_i t_{t-1} + W_i x_t + b_i) \quad (1)$$

$$f_t = \sigma(U_f t_{t-1} + W_f x_t + b_f) \quad (2)$$

$$o_t = \sigma(U_o t_{t-1} + W_o x_t + b_o) \quad (3)$$

$$c_t = f_t \times c_{t-1} \times i_t \times \tanh(U_c t_{t-1} + W_c x_t + b_c) \quad (4)$$

$$h_t = o_t \times \tanh(c_t) \quad (5)$$

Where,

i_t : Input Gate

f_t : Forgate Gate

o_t : Output Gate

t : Time

c_t : Cell State

h_t : Hidden Unite

This research compares the architecture of LSTM, namely layer 1, layer 2, and layer 3. The architecture of the three layers is illustrated in the figure below by taking references from [9], [18] but there are some changes, namely adding a batch normalization layer that serves to normalize activation so as to improve accuracy and speed up training [33]–[35].

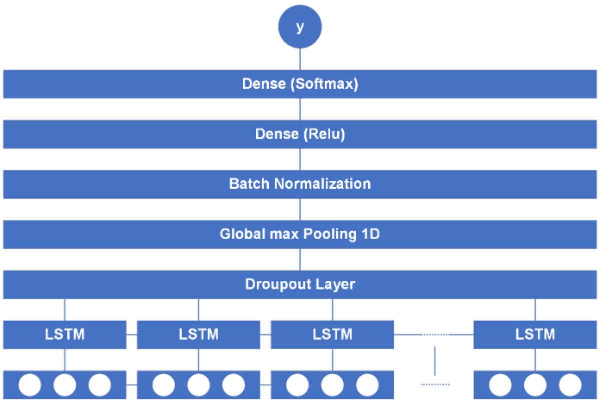


Figure 2. Single Layer LSTM

Figure 2 is an illustration of a one-layer LSTM architecture, starting from the embedding matrix which contains the results of FastText word embedding in the form of a vector. Then followed by the LSTM layer, droupout layer, global max pooling 1D layer, batch normalization, dense (Relu) layer, and ended by the Softmax layer as the activation function in the dense layer. Details of the one-layer LSTM architecture are shown in **Table 3**.

Table 3. Single layer LSTM architecture

Layer (type)	Output Shape	Number of Parameters
Embedding (Embedding)	(None, 123, 300)	20704200
LSTM (LSTM)	(None, 123, 64)	93440
Global_Max_Pooling 1D (GlobalMaxPooling1D)	(None, 64)	0
Batch_Normalizzation (BatchNormalization)	(None, 64)	256
Dense (Desne)	(None, 64)	4160
Droupour (Droupout)	(None, 64)	0
Dense_1 (Dense)	(None, 2)	130

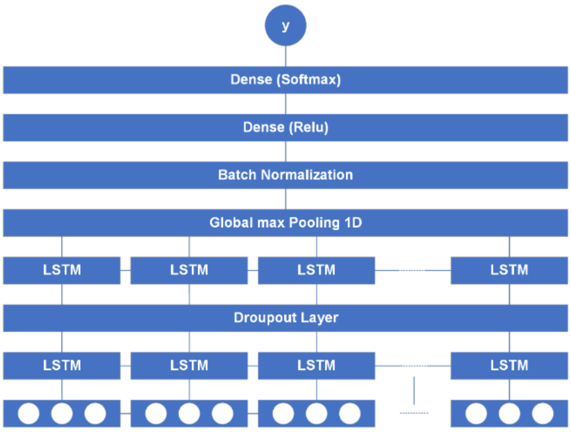


Figure 3. Double Layer LSTM

Figure 3 shows the architecture of the two-layer LSTM, where the steps of the two-layer LSTM architecture are similar to the one-layer LSTM. The difference between the one-layer and two-layer LSTM lies in the additional LSTM layer before the 1D global max pooling layer. The addition of the LSTM layer before global max pooling affects the results of the processed dataset. Details of the two-layer LSTM architecture are in **Table 4**.

Table 4. Double layer LSTM architecture

Layer (type)	Output Shape	Number of Parameters
Embedding (Embedding)	(None, 123, 300)	20704200
LSTM_1 (LSTM)	(None, 123, 64)	93440
LSTM_2 (LSTM)	(None, 123, 64)	33024
Global_Max_Pooling 1D (GlobalMaxPooling1D)	(None, 64)	0
Batch_Normalizzation (BatchNormalization)	(None, 64)	256
Dense (Desne)	(None, 64)	4160
Droupour (Droupout)	(None, 64)	0
Dense_1 (Dense)	(None, 2)	130

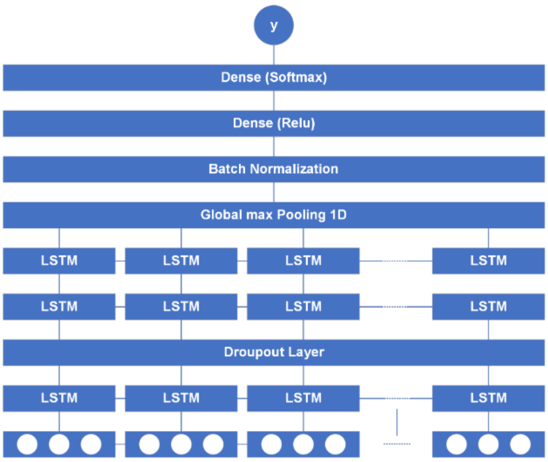


Figure 4. Triple Layer LSTM

Figure 4 illustrates the three-layer LSTM architecture. The main difference between the three-layer LSTM architecture and the one-layer and two-layer LSTM architectures lies in the number of LSTM layers before the global max pooling layer and dense layer. The three-layer LSTM architecture has two LSTM layers before the global max pooling layer and dense layer while the two-layer LSTM architecture has only one LSTM layer before the global max pooling layer and the one-layer LSTM architecture has no new LSTM layer between the global max pooling layer and the dense layer. **Table 5** shows the details of the three-layer LSTM architecture.

Table 5. Triple layer LSTM architecture

Layer (type)	Output Shape	Number of Parameters
Embedding (Embedding)	(None, 123, 300)	20704200
LSTM_3 (LSTM)	(None, 123, 64)	93440
LSTM_4 (LSTM)	(None, 123, 64)	33024
LSTM_5 (LSTM)	(None, 123, 64)	33024
Global_Max_Pooling 1D (GlobalMaxPooling1D)	(None, 64)	0
Batch_Normalizzation (BatchNormalization)	(None, 64)	256
Dense (Desne)	(None, 64)	4160
Droupour (Droupout)	(None, 64)	0
Dense_1 (Dense)	(None, 2)	130

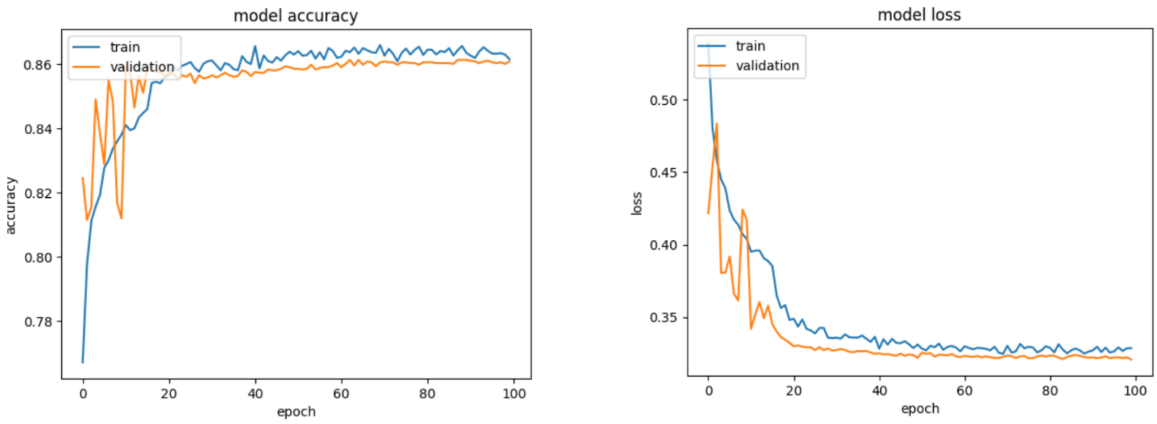
F. Validation and Testing

The testing process uses 20% of the tens of data sets with a total of 10000 data. The data used for validation during the training process is 10%, which is 5000 data.

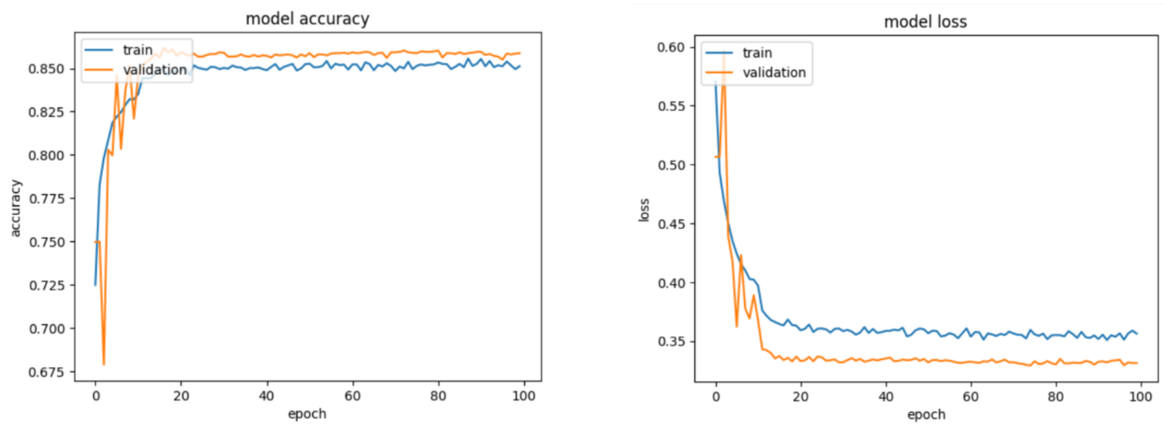
Results and Discussion

A. Model Development

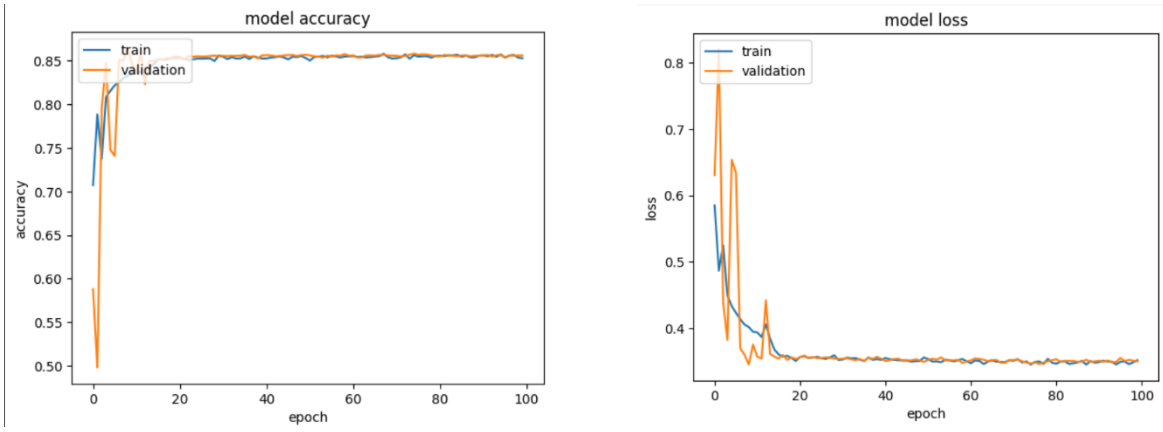
The model developed in this research is LSTM with architecture based on Figures 2, 3, and 4. The batch size used is batch size 16 with 100 epochs. The parameters used in LSTM and Dense Relu are 64. The learning rate used in the LSTM architecture in this study is 0.005. The purpose of this research is to find the best LSTM architecture layer between single layer, double layer, and tier layer. **Figure 5** shows the results of training data and validation where the blue line represents the model accuracy and model loss of training data and the yellow line represents the model accuracy and model loss of validation.



(a)



(b)



(c)

Figure 5. Training accuracy and loss accuracy of LSTM models using: (a) single layer LSTM; (b) double layer LSTM; (c) triple layer LSTM.

Figure 5 shows the distance between training accuracy and validation accuracy on one, two and three layer LSTM architectures. The one-layer LSTM architecture in the figure shows the distance between the blue line and the yellow line is still distant with training accuracy and validation accuracy intersecting at the end of the epoch, the two-layer LSTM architecture has similarities with the one-layer LSTM that is between training accuracy and validation accuracy still has a distance but validation accuracy is higher than training accuracy and does not intersect at the end of the epoch. Meanwhile, the three-layer LSTM architecture has stabilized between training accuracy and validation accuracy as shown by the blue and yellow lines that are not far apart and intersect each other. Details of the results of each LSTM architecture are in **Table 6**.

Table 6. Results of training, validation and testing of LSTM architecture

Layer	Training Accuracy	Validation Accuracy	Testing Accuracy
Single Layer	0,856	0,856	0,867
Double Layer	0,846	0,852	0,854
Triple Layer	0,848	0,846	0,864

Table 6 shows the details of the results from training, validation, and testing of the one-layer, two-layer, and three-layer LSTM architectures. The one-layer LSTM architecture has the highest accuracy compared to the two-layer and three-layer LSTM. The highest accuracy of the one-layer LSTM shows that the best architecture is the one-layer LSTM architecture at the expense of the training and validation process which still has a distance between its accuracy.

Conclusion

This research aims to compare one layer, two layer, and three layer LSTM architectures using the IMDB Movie Reviews dataset of 50,000 data. After going through the preprocessing process to testing, it was found that the one-layer LSTM architecture has the best accuracy results between one-layer, two-layer and three-layer LSTM architectures, namely 0.856 and 0.867 for training accuracy and testing accuracy. With the best accuracy results obtained by the one-layer LSTM architecture, it shows that the number of layers in the LSTM architecture affects the results. However, this study cannot conclude that the level of model complexity can improve accuracy because the results on the two-layer LSTM architecture are lower than the one-layer LSTM and the three-layer LSTM is higher than the two-layer LSTM. This research can be a reference for other researchers to improve future research. Changes to some parameters in the architecture can be applied to improve accuracy. The addition of epochs can also be used because in this study the blue and yellow lines intersect at the end of the epoch.

References

[1] F. B. Putra *et al.*, "Identification of Symptoms Based on Natural Language Processing (NLP) for Disease Diagnosis Based on International Classification of Diseases and Related Health Problems (ICD-11)," *IES 2019 - International Electronics Symposium: The Role of Techno-Intelligence in Creating an Open Energy System Towards Energy Democracy, Proceedings*, pp. 1–5, 2019, doi: 10.1109/ELECSYM.2019.8901644.

[2] Y. Kang, Z. Cai, C. W. Tan, Q. Huang, and H. Liu, "Natural language processing (NLP) in management research: A literature review," *https://doi.org/10.1080/23270012.2020.1756939*, vol. 7, no. 2, pp. 139–172, Apr. 2020, doi: 10.1080/23270012.2020.1756939.

[3] M. K. Nammous and K. Saeed, *Natural language processing: Speaker, language, and gender identification with LSTM*, vol. 883. Springer Singapore, 2019. doi: 10.1007/978-981-13-3702-4_9.

[4] A. M. Alayba, V. Palade, M. England, and R. Iqbal, "A combined CNN and LSTM model for Arabic sentiment analysis," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11015 LNCS, pp. 179–191, 2018, doi: 10.1007/978-3-319-99740-7_12/FIGURES/5.

[5] D. Rohidin, N. A. Samsudin, and M. M. Deris, "Association rules of fuzzy soft set based classification for text classification problem," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 3, pp. 801–812, Mar. 2022, doi: 10.1016/J.JKSUCI.2020.03.014.

[6] D. Bennett, K. E. Dewi, and R. E. Sagala, "You may also like Document Image Extraction System Design N I Widiastuti and K E Dewi-Near Surface Disposal of Radioactive Waste: Safety Requirements (Safety Standards Series No WS-R-1) and Safety Assessment for Near Surface Disposal of Radioactive Waste: Safety Guide (Safety Standards Series No WS-G-1.1) Using Summarization to Optimize Text Classification", doi: 10.1088/1757-899X/407/1/012157.

[7] M. Abidin, M. I. Abidin, I. Nurtanio, and A. Achmad, "Deepfake Detection in Videos Using Long Short-Term Memory and CNN ResNext," *ILKOM Jurnal Ilmiah*, vol. 14, no. 3, pp. 178–185, Dec. 2022, doi: 10.33096/ilkom.v14i3.1254.178-185.

[8] A. G. Salman, Y. Heryadi, E. Abdurahman, and W. Suparta, "Single Layer & Multi-layer Long Short-Term Memory (LSTM) Model with Intermediate Variables for Weather Forecasting," *Procedia Comput Sci*, vol. 135, pp. 89–98, 2018, doi: 10.1016/J.PROCS.2018.08.153.

[9] A. Setyanto *et al.*, "Arabic Language Opinion Mining Based on Long Short-Term Memory (LSTM)," *Applied Sciences 2022, Vol. 12, Page 4140*, vol. 12, no. 9, p. 4140, Apr. 2022, doi: 10.3390/APP12094140.

[10] N. Halpern-Wight, M. Konstantinou, A. G. Charalambides, and A. Reinders, "Training and Testing of a Single-Layer LSTM Network for Near-Future Solar Forecasting," *Applied Sciences 2020, Vol. 10, Page 5873*, vol. 10, no. 17, p. 5873, Aug. 2020, doi: 10.3390/APP10175873.

[11] A. H. Ombabi, W. Ouarda, and A. M. Alimi, "Deep learning CNN–LSTM framework for Arabic sentiment analysis using textual information shared in social networks," *Soc Netw Anal Min*, vol. 10, no. 1, pp. 1–13, 2020, doi: 10.1007/s13278-020-00668-1.

[12] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, "Advances in Pre-Training Distributed Word Representations," *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, pp. 52–55, Dec. 2017, Accessed: Jul. 05, 2023. [Online]. https://arxiv.org/abs/1712.09405v1

[13] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning Word Vectors for Sentiment Analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:*

Human Language Technologies, Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 142–150. [Online]. Available: http://www.aclweb.org/anthology/P11-1015

[14] E. H. Muktafin and P. Kusrini, "Sentiments analysis of customer satisfaction in public services using K-nearest neighbors algorithm and natural language processing approach," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 1, pp. 146–154, Feb. 2021, doi: 10.12928/TELKOMNIKA.V19I1.17417.

[15] M. S. Islam *et al.*, "Machine Learning-Based Music Genre Classification with Pre-Processed Feature Analysis," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 7, no. 3, pp. 491–502, Jan. 2022, doi: 10.26555/JITEKI.V7I3.22327.

[16] I. Irawanto, C. Widodo, A. Hasanah, P. Adhitya, and D. Kusumah, "Sentiment analysis and classification of Forest Fires in Indonesia," vol. 15, no. 1, pp. 175–185, 2023.

[17] 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, pp. 209–218, Jul. 2019, doi: 10.22146/ijccs.41236.

[18] M. Hayaty and A. H. Pratama, "Performance of Lexical Resource and Manual Labeling on Long Short-Term Memory Model for Text Classification," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 9, no. 1, pp. 74–84, Feb. 2023, doi: 10.26555/jiteki.v9i1.25375.

[19] L. A. Mullen, K. Benoit, O. Keyes, D. Selivanov, and J. Arnold, "Fast, Consistent Tokenization of Natural Language Text," *J Open Source Softw*, vol. 3, no. 23, p. 655, 2018, doi: 10.21105/joss.00655.

[20] A. I. Saad, "Opinion Mining on US Airline Twitter Data Using Machine Learning Techniques," *16th International Computer Engineering Conference, ICENCO 2020*, pp. 59–63, 2020, doi: 10.1109/ICENCO49778.2020.9357390.

[21] F. Almeida and G. Xexéo, "Word Embeddings: A Survey".

[22] B. Wang, A. Wang, F. Chen, Y. Wang, and C. C. J. Kuo, "Evaluating word embedding models: methods and experimental results," *APSIPA Trans Signal Inf Process*, vol. 8, p. e19, 2019, doi: 10.1017/ATSIP.2019.12.

[23] K. K. Agustiningsih, E. Utami, and O. M. A. Alsyaibani, "Sentiment Analysis and Topic Modelling of The COVID-19 Vaccine in Indonesia on Twitter Social Media Using Word Embedding," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 8, no. 1, pp. 64–75, Apr. 2022, doi: 10.26555/jiteki.v8i1.23009.

[24] W. K. Sari, D. P. Rini, and R. F. Malik, "Text Classification Using Long Short-Term Memory With GloVe Features," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 5, no. 2, pp. 85–100, Dec. 2019, doi: 10.26555/JITEKI.V5I2.15021.

[25] J.-H. Wang, T.-W. Liu, X. Luo, and L. Wang, "An LSTM Approach to Short Text Sentiment Classification with Word Embeddings," pp. 214–223.

[26] Z. Huang, B. Research, W. Xu, and K. Y. Baidu, "Bidirectional LSTM-CRF Models for Sequence Tagging".

[27] P. Wang, Y. Qian, F. K. Soong, L. He, and H. Zhao, "Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Recurrent Neural Network," Oct. 2015, Accessed: Aug. 04, 2023. [Online]. Available: https://arxiv.org/abs/1510.06168v1

[28] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," *Adv Neural Inf Process Syst*, vol. 4, no. January, pp. 3104–3112, Sep. 2014, Accessed: Aug. 04, 2023. [Online]. Available: https://arxiv.org/abs/1409.3215v3

[29] P. F. Muhammad, R. Kusumaningrum, and A. Wibowo, "Sentiment Analysis Using Word2vec And Long Short-Term Memory (LSTM) For Indonesian Hotel Reviews," *Procedia Comput Sci*, vol. 179, pp. 728–735, Jan. 2021, doi: 10.1016/J.PROCS.2021.01.061.

[30] A. W. Saputra, A. W. Saputra, A. P. Wibawa, U. Pujianto, A. B. P. Utama, and A. Nafalski, “LSTM-based Multivariate Time-Series Analysis: A Case of Journal Visitors Forecasting,” *ILKOM Jurnal Ilmiah*, vol. 14, no. 1, pp. 57–62, Apr. 2022, doi: 10.33096/ilkom.v14i1.1106.57-62.

[31] T. Yu, J. Chen, N. Yan, and X. Liu, “A Multi-Layer Parallel LSTM Network for Human Activity Recognition with Smartphone Sensors,” *2018 10th International Conference on Wireless Communications and Signal Processing, WCSP 2018*, 2018, doi: 10.1109/WCSP.2018.8555945.

[32] X. She and D. Zhang, “Text Classification Based on Hybrid CNN-LSTM Hybrid Model,” *Proceedings - 2018 11th International Symposium on Computational Intelligence and Design, ISCID 2018*, vol. 2, pp. 185–189, 2018, doi: 10.1109/ISCID.2018.10144.

[33] L. Huang, D. Yang, B. Lang, and J. Deng, “Decorrelated Batch Normalization”.

[34] S. Santurkar, D. Tsipras, A. Ilyas, and A. M. ˛ A. Mit, “How Does Batch Normalization Help Optimization?”.

[35] J. Bjorck, C. Gomes, B. Selman, and K. Q. Weinberger, “Understanding Batch Normalization”.