BAREKENG: Journal of Mathematics and Its Applications
December 2023 Volume 17 Issue 4 Page 2235–2242
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol17iss4pp2235-2242

SENTIMENT ANALYSIS WITH LONG-SHORT TERM MEMORY (LSTM) AND GATED RECURRENT UNIT (GRU) ALGORITHMS

M. Nazhif Abda Putera Khano [1], Dewi Retno Sari Saputro[2*], Sutanto[3], Antoni Wibowo[4]

[1,2,3]Department of Mathematics, Faculty of Mathematics and Natural Sciences, Sebelas Maret University
Ir. Sutami Street No. 36, Surakarta, 57126, Indonesia

[4]Master of Information Technology, Bina Nusantara University
K. H. Syahdan Street No. 9, Kemanggisan, Palmerah Jakarta, 11480, Indonesia

*Corresponding author's e-mail: * **dewiretnoss@staff.uns.ac.id***

ABSTRACT

Sentiment analysis is a form of machine learning that functions to obtain emotional polarity values or data tendencies from data in the form of text. Sentiment analysis is needed to analyze opinions, sentiments, reviews, and criticisms from someone for a product, service, organization, topic, etc. Recurrent Neural Network (RNN) is one of the Natural Language Processing (NLP) algorithms that is used in sentiment analysis. RNN is a neural network that can use internal memory to process input. RNN itself has a weakness in Long-Term Memory (LTM). Therefore, this article examines the combination of Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) algorithms. GRU is an algorithm that is used to make each recurrent unit able to record adaptively at different time scales. Meanwhile, LSTM is a network architecture with the advantage of learning long-term dependencies on data. LSTM can remember long-term memory information, learn long-sequential data, and form information relation data in LTM. The combination of LSTM and GRU aims to overcome RNN's weakness in LTM. The LSTM-GRU is combined by adding GRU to the data generated from LSTM. The combination of LSTM and GRU creates a better performance algorithm for addressing the LTM problem.

Article History:

Received: 21st July 2023
Revised: 8th October 2023
Accepted: 6th November 2023

Keywords:

Sentiment Analysis;
RNN;
GRU;
LSTM.

How to cite this article:

M. N. A. P. Khano, D. R. S. Saputro, Sutanto and A. Wibowo., "SENTIMENT ANALYSIS WITH LONG-SHORT TERM MEMORY (LSTM) AND GATED RECURRENT UNIT (GRU) ALGORITHMS," *BAREKENG: J. Math. & App.*, vol. 17, iss. 4, pp. 2235-2242, December, 2023.

Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*

Research Article · Open Access

1. INTRODUCTION

Technological developments cannot be ignored anymore because they are needed to solve several problems. One example of using technology is analyzing data from opinions, sentiments, reviews, and criticisms in the form of text. Opinions are the essence of almost all human activities because it's the main influence of everyday human behavior where when you want to make a decision, you need the opinion of many people [1]. Machine learning is one of the methods used to analyze this data. One of machine learning is sentiment analysis. Sentiment analysis is a form of machine learning that has functions to obtain emotional polarity values or data tendencies, which can be used to analyze data in the form of text [2].

Sentiment analysis is needed to analyze opinions, sentiments, reviews, and criticisms from someone for a product, service, organization, topic, etc. sentiment analysis can be used in product reviews, film recommendations, predicting stock values, and monitoring public opinion [2]. Sentiment analysis or commonly called opinion mining process is used to find user views on several topics or texts put forward by users so that they can determine whether the contents are positive, negative, or neutral [3]. Sentiment analysis itself has various algorithms that can be used because the methods used to analyze sentiment rely heavily on case studies [4]. Recurrent Neural Network (RNN) is an algorithm that can be utilized for sentiment analysis and have time train fastest of all methods used [5].

The RNN is a neural network that is used for processing an input, by using its own memory [6]. The RNN is classified as an artificial neural network system commonly used in areas like handwriting recognition [7] or speech recognition [8]. Memory in the RNN is showing that, depending on all previous computations, it will do the same task for any element from a sequence. RNN itself can be said to remember what has been done so far. RNN is an adaptation of neural networks for data sequences [9]. RNN in sentiment analysis has a weakness in long-term memory (LTM) [2].

A study conducted by [10] used Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) algorithms to improve the effectiveness of sentiment analysis. Although LSTM and GRU can both independently transform any signal into a set of features, working together they enable high-density feature analysis to produce the best representation of retinal reports [11]. LSTM and GRU are expected to solve the LTM problem in RNN. Based on this, this article will examine the combination of the LSTM and GRU algorithms to overcome the RNN problem in LTM.

2. RESEARCH METHODS

The Visualization of Similarities Viewer (VosViewer), LSTM, and GRU have been the three foundations on which this research turned into undertaken

2.1 Visualization of Similarities Viewer (VosViewer)

Visualization of Similarities Viewer (VosViewer), a computer tool called Viewer (VosViewer) allows users to examine bibliographical knowledge network maps for free [12]. VosViewer has an edge compared with other analysis programs because employs text mining to find combinations of mapping of associated phrases and creates clustering techniques for data tests [13]. Furthermore, VosViewer provides users with a number of interactive features and options to make accessing and exploring data in the field of bibliometrics more convenient [14][15]. The words (co-words) used in a document can be utilized to infer scientific notions. The foundation of colingual analysis is co-occurrence analysis. These documents' indexes are made up of terms that come from two or more other documents [16]. By evaluating the terms' strengths, the co-word analysis seeks to understand the content, patterns, and trends of various documents [17][18][19]. The mapping generated by VosViewer from bibliographical data related to sentiment analysis, LSTM, and GRU can be seen in the **Figure 1**. **Figure 1** shows the results of bibliometric visualization through extraction in VOSviewer with title and abstract field.

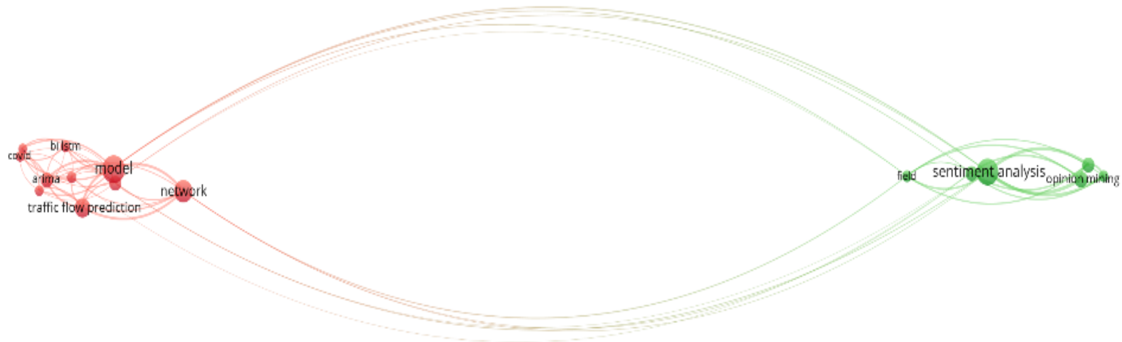

Figure 1. Visualization with title and abstract restriction

Figure 1 is visualized on VosViewer with 16 number of terms. The greater the circle of each variable, indicating that it is widely used. In the above mapping, it is known that the dominating terms are model, network, and sentiment analysis.

2.2 Long-Short Term Memory (LSTM)

LSTM algorithm is a new development of RNN. LSTM can solve long-term memory problems. According to [6], LSTM is a unique type from RNN that can be used to study the long-term dependence on data. LSTM, developed by [20], has been used as a further version of RNN and can overcome the limitations of RNN by using hidden layers or memory cells. Repetitive modules in LSTM are more complicated than RNNs in general, which have four layers with two states, namely the hidden state and the cell state. LSTM has a memory cell that is used to store information. In the calculations, input gates, output gates, and forget gates are used to manage the memory. The input gate functions to determine new data to enter into the cell state. Output gate functions to determine the result of the input that has been entered into the cell state. Forget gate functions to delete data that is not needed anymore in the cell state. The gate structure of the LSTM based on [21] can be seen in Figure 2.

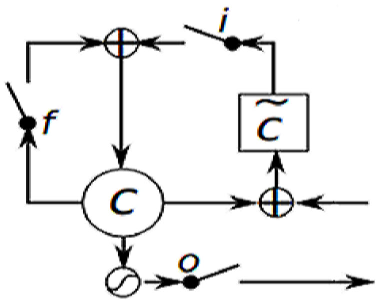

Figure 2. The gate structure of LSTM

Based on [22], unlike recurrent units in general, each j in the LSTM unit has memory c_t^j at time t. The h_t^j output from the LSTM is

$$h_t^j = o_t^j \, tanh \, tanh \left(c_j^t\right) \tag{1}$$

where o_t^j is the output gate, which has some memory content. Equation (2) show the output gate of LSTM

$$o_t^j = \sigma(W_o x_t + U_c h_{t-1} + V_o c_t)^j \tag{2}$$

with σ function of the sigmoid logistic. A diagonal matrix V_o. Existing memory was deleted and added new memory $\tilde{c}_t^j$ to update memory cell c_t^j incrementally

$$c_t^j = f_t^j c_{t-1}^j + i_t^j \tilde{c}_t^j, \tag{3}$$

where new memory

$$\tilde{c}_t^j = tanh\,(W_c x_t + U_c h_{t-1})^j \tag{4}$$

The forget gate f_t^j is modulated degree to which the existing memory is deleted and where the memory cell is added with new memory by the input gate i_t^j. Gates is written with **Equation (5)** and **Equation (6)**

$$f_t^j = \sigma\left(W_f x_t + U_f h_{t-1} + V_f c_t\right)^j \tag{5}$$

$$i_t^j = \sigma(W_i x_t + U_i h_{t-1} + V_i c_t)^j \tag{6}$$

where V_f and V_i are diagonal matrices.

2.3 Gated Recurrent Unit (GRU)

GRU is included in the development of the RNN algorithm which is similar to LSTM and can overcome the problem of using extra memory units in LSTM. According to [21], GRU is another form of RNN that is used to make each recurrent unit able to record adaptively at different times of scale. Gating units in GRU can control data movement or information with units. Even so, this is done without using additional memory cells. GRU also has two gates, that is the update gate and the reset gate which modulate the movement of data or information that passes one another in each hidden unit. The GRU update gate functions to determine how much past information should be stored and to remember new information. Reset gate functions to make decisions by combining new input and past information and determining whether new information should be forgotten or not. The gate structure of the GRU is shown in the **Figure 3**.

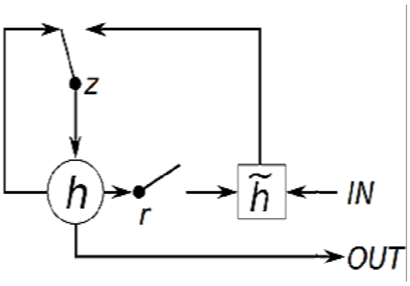

Figure 3. The gate structure of GRU

Based on [21], at some time t on GRU, linear interposition among previous activation h_{t-1}^j and the new activation h_t^j is defined as activation h_t^j with **Equation (7)**

$$h_t^j = \left(1 - z_t^j\right)h_{t-1}^j + z_t^j \underline{h}_t^j \tag{7}$$

The GRU update gate is denoted by z_{ret}^j to find out how many units have been changed. The update gate is written with **Equation (8)**

$$z_t^j = \sigma(W_z x_t + U_z h_{t-1})^j \tag{8}$$

Furthermore, for the activation candidate, h_t^j is written as **Equation (9)**

$$\underline{h}_t^j = tanh\,(W x_t + U(r_t \odot h_{t-1}) \tag{9}$$

where $\odot$ represents the multiplication by element and the set of calculations from the reset gate is r_t. As r_t^j approaches 0, the reset gate operates to make the unit appear to be extracting every symbol in the given sequence to read first and is allowed to clear calculations of the existing state. Reset gate r_t^j is calculated the same as the update gate,

$$r_t^j = \sigma(W_r x_t + U_r h_{t-1})^j \tag{10}$$

3. RESULTS AND DISCUSSION

The RNN was created to overcome the problem of neural networks, which have previously extracted information from current time periods and failed to take into account useful data related to their timing sequence and spatial arrangement [2]. An RNN that can process data sequences is also needed to obtain previous information and reuse existing information. RNN is commonly used for language models because of its ability to remember long-term dependencies [23].

The words are converted into machine-readable vectors in RNN, and a sequence of vectors is processed one at a time. In the RNN process, the vector goes through the previous head state towards the next step in a sequence, in a hidden state the access to the neural network from the memory containing the information from the past data and add the tanh activation. Tanh activation on the RNN is useful to help regulate the spikes in value when passing through the network. Values can't get over between -1 and 1 by the tanh function. Vectors passing through the neural network transform by repeated operations and explode in value causing other values to appear insignificant and an activation of tan is needed to help with the problem.

However, with increasing time lag, the RNN gradient can disappear through the open RNN to become a feed-forward neural network. The gradient is used to renew the neural network's weights. The vanishing gradient problem occurs when the gradient force on return propagates over time, the slope value becomes very small. It does not make a significant contribution to learning. In RNN layers that get small gradient updates are not learned. Depending on past computations or "remembering" what has been processed, the RNN will do the same task to every element of the data sequence. RNN can forget what has been seen and longer sequences have short-term memory.

To overcome this problem, the LSTM and GRU algorithms are used. LSTM and GRU have been successfully applied to forest phenology prediction [24]. According to [25], using a hybrid network for sentiment analysis for Arabic and can achieve peak performance. Internal mechanisms called gates allow LSTM and GRU to regulate information flow. The Gates will discover in the sequence what data is critical for keeping or forgetting. LSTM can spontaneously retain the memory of LTM and has a more complex structure compared to RNN. The structure of the LSTM can overcome the problem of long-term dependence. As another improvement from RNN, the GRU algorithm with a gated structure is formed. GRU combines cell state and hidden state resulting in different results from LSTM. Due to its high efficiency and simplicity of form, GRU is very widespread in natural language processing [2]. Compared to LSTM, GRU is the latest application of RNN, but both are appropriate for text data and because of this have been harmonized to achieve more accurate classification. In a study conducted by [26], using LSTM and GRU in predictions for COVID-19, it was found that LSTM and GRU showed much better robustness and predictions compared to actual figures which described lower prediction errors. The merger of LSTM and GRU aims to overcome RNN's weaknesses in LTM. A hybrid LSTM-GRU can learn to store and predict only relevant data. The production of LSTM will be input to GRU as a result of the merging of LSTM and GRU.

The update gate formula from the information on the forget gate for each LSTM cell is,

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{11}$$

With the information penetration rate represented by $f_t \in [0,1]$ and the sigmoid function is $\sigma(\cdot)$. Then the filtered data will pass through the input gate with **Equation (12), Equation (13),** and **Equation (14)**.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{12}$$

$$\underline{C} = tanh\ (W_c \cdot [h_{t-1}, x_t] + b_c) \tag{13}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \underline{C_t} \tag{14}$$

Input x_t and the previously hidden layer state h_{t-1} are processed at this stage with the sigmoid and tanh functions. Then, **Equation (15)** and **Equation (16)** are the LSTM output gate conversion formula

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{15}$$

$$h_t = o_t \cdot tanh\ tanh\ (C_t) \tag{16}$$

Then the output from the LSTM becomes the network layer of the GRU cell. The formula data forward propagation at equation below.

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{17}$$

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{18}$$

$$\underline{h_t} = tanh\ tanh\ (W \cdot [r_t * h_{t-1}, x_t]) \tag{19}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \underline{h_t} \tag{20}$$

where input at GRU is the hidden layer output from LSTM ($x_t = h_t$).

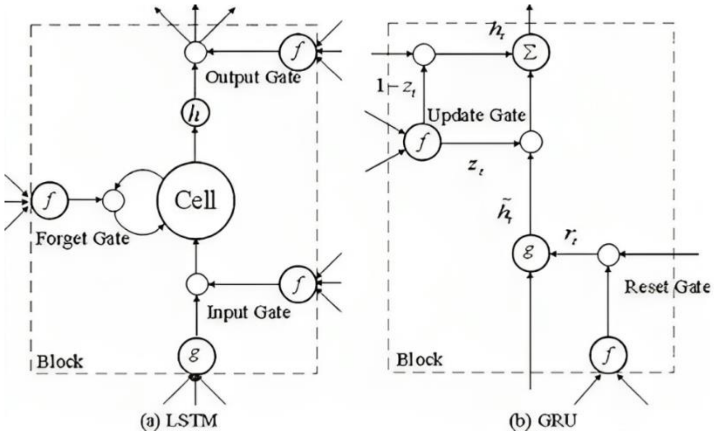

Figure 4. Structure of LSTM-GRU

The structure from LSTM-GRU in **Figure 4** shows the flow of the process data that is input through the LSTM and the output becomes input for the GRU.

In the first stage of a model, input data that can be processed is taken from an LSTM layer. Each neuron in the LSTM on this path creates a weight value when it receives input data. In the next step, data from an LSTM layer will be sent to GRU. A weight value is generated as it travels from LSTM to GRU layers. Input neurons are receiving data from a layer of GRU, and the correct weights have been generated. A weight value is generated as it travels from LSTM to GRU layers. Output neurons are receiving the GRU layer data, which is then converted into appropriate weights. By comparison of output with an initial value, the cost function is then calculated. The weights shall be adjusted to take into account the difference in projected and real values as soon as the minimum point of the cost function is achieved. The weights are kept on the basis of future estimates.

Furthermore, the LSTM-GRU merger is applied using Python by producing the following syntax. Python has some of the widest support for deep learning as a programming language [27]. The syntax below is model creation for LSTM cells and GRU cells.

```
#LSTM Cell
class LSTM_net(nn.Module):
    def __init__(self, input_size, hidden_size, output_size):
        super(LSTM_net, self).__init__()
        self.hidden_size = hidden_size
        self.lstm_cell = nn.LSTM(input_size, hidden_size)
        self.h2o = nn.Linear(hidden_size, output_size)
        self.softmax = nn.LogSoftmax(dim=2)

    def forward(self, input_, hidden):
        out, hidden = self.lstm_cell(input_.view(1, 1, -1), hidden)
        output = self.h2o(hidden[0])
        output = self.softmax(output)
        return output.view(1, -1), hidden

    def init_hidden(self):
        return (torch.zeros(1, 1, self.hidden_size), torch.zeros(1, 1,
self.hidden_size))


n_hidden = 128
net = LSTM_net(n_letters, n_hidden, n_languages)
```

```
train_setup(net, lr=0.0005, n_batches=100, batch_size = 256)
```

```
#GRU cell
class GRU_net(nn.Module):
    def __init__(self, input_size, hidden_size, output_size):
        super(GRU_net, self).__init__()
        self.hidden_size = hidden_size
        self.gru_cell = nn.GRU(input_size, hidden_size)
        self.h2o = nn.Linear(hidden_size, output_size)
        self.softmax = nn.LogSoftmax(dim=2)

    def forward(self, input_, hidden):
        out, hidden = self.gru_cell(input_.view(1, 1, -1), hidden)
        output = self.h2o(hidden)
        output = self.softmax(output)
        return output.view(1, -1), hidden

    def init_hidden(self):
        return torch.zeros(1, 1, self.hidden_size)


n_hidden = 128
net = LSTM_net(n_letters, n_hidden, n_languages)
train_setup(net, lr=0.0005, n_batches=100, batch_size = 256)
```

Based on the explanation above, LSTM-GRU can overcome the RNN problem on LTM where the loop that occurs on the RNN can be overcome by the forget gate owned by LSTM by deleting data from the initial input that is no longer useful and passing through GRU where the update gate can remember past information and add new information and reset gate generate data that has been combined with new information. The combination of LSTM and GRU creates an algorithm with better performance for solving LTM problems compared to RNN. In RNN if the weight value is too small, the gradient will disappear and if the gradient is too large it can make the gradient explode then it means that RNN is sensitive to time step and doesn't have long-term memory. LSTM adds input gate and forget gate to solve the problem of gradient disappear and gradient explode, so that long-term information can be captured, and it can have better performance in long sequence. Then, GRU can reduces matrix multiplication and can save a lot of time without sacrificing performance with one gate less than LSTM. Refered to [28] the more neural network layers can improve the abstract processing ability.

4. CONCLUSIONS

According to the results of the study, RNN has weaknesses in dealing with LTM problems. LSTM and GRU are used to solve LTM problems in RNN. LSTM has four layers with two states, namely hidden state and cell state. There are three gates in the memory cell from LSTM, namely, input gate, output gate, and forget gate. GRU will allow each recurring unit to be recorded in a variety of time periods. Gating units in GRU are able to control data movement between units. There are two gates in GRU, namely, the update gate and the reset gate.

The LSTM-GRU combination can overcome the LTM problem by overcoming loops that occur in the RNN with forget gates and bypassing update gates and reset gates in GRU. LSTM-GRU merging creates an algorithm with better performance than RNN to solve LTM problems.

ACKNOWLEDGMENT

Thank you to the Softcomputing Mathematics Research Group for their moral support and constructive comments in improving this paper. We also express our gratitude to the Institute for Research and Community Service (LPPM) Universitas Sebelas Maret (UNS) for the research group grant through the letter of approval for research implementation of non-APBN funds No. 228/UN27.22 /PT.01.03/2023.

REFERENCES

[1] B. Liu, *Sentiment Analysis: A Fascinating Problem*. 2012. doi: 10.1007/978-3-031-02145-9_1.
[2] 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.
[3] Muljono, D. Putri Artanti, A. Syukur, A. Prihandono, and D. I. Rosal Moses Setiadi, "Analisis Sentimen Untuk Penilaian Pelayanan Situs Belanja Online Menggunakan Algoritma Naïve Bayes," 2018. [Online]. Available: http://twitter.com
[4] Y. A. Singgalen, "Pemilihan Metode dan Algoritma dalam Analisis Sentimen di Media Sosial : Sistematic Literature Review," *J. Inf. Syst. Informatics*, vol. 3, no. 2, pp. 278–302, 2021, doi: 10.33557/journalisi.v3i2.125.
[5] M. N. Alim, "Pemodelan Time Series Data Saham LQ45 dengan Algoritma LSTM, RNN, dan Arima," *Pros. Semin. Nas. Mat.*, vol. 6, pp. 694–701, 2023, [Online]. Available: https://journal.unnes.ac.id/sju/index.php/prisma/
[6] 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, pp. 1–25, 2018, doi: 10.1002/widm.1253.
[7] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 855–868, 2009, doi: 10.1109/TPAMI.2008.137.
[8] A. M. and G. H. Alex Graves, "Speech Recognition with Deep Recurrent Neural Networks , Department of Computer Science, University of Toronto," *Dep. Comput. Sci. Univ. Toronto*, vol. 3, no. 3, pp. 45–49, 2013, [Online]. Available: https://ieeexplore.ieee.org/stampPDF/getPDF.jsp?tp=&arnumber=6638947&ref=aHR0cHM6Ly9pZWVleHBsb3JlLmllZWUub3JnL2Fic3RyYWN0L2RvY3VtZW50LzY2Mzg5NDc/Y2FzYV90b2tlbj1OQUo1VFJxWk5JRUFBQUFBOmtPZmdDbS00NGhqaGI2N3dMd2JrU3lSaEdJREhBWnpMSkxoT201Um5YMXR0S0poUDAtM2hkbT
[9] S. Sachin, A. Tripathi, N. Mahajan, S. Aggarwal, and P. Nagrath, "Sentiment Analysis Using Gated Recurrent Neural Networks," *SN Comput. Sci.*, vol. 1, no. 2, 2020, doi: 10.1007/s42979-020-0076-y.
[10] N. Aslam, F. Rustam, E. Lee, P. B. Washington, and I. Ashraf, "Sentiment Analysis and Emotion Detection on Cryptocurrency Related Tweets Using Ensemble LSTM-GRU Model," *IEEE Access*, vol. 10, no. April, pp. 39313–39324, 2022, doi: 10.1109/ACCESS.2022.3165621.
[11] D. M. Kotambkar and P. M. Wankhede, "Hybrid LSTM/GRU-based Domain Adaptation Model for Correlation Analysis to Detect Glaucoma," *Int. J. Electr. Electron. Eng.*, vol. 10, no. 1, pp. 168–175, 2023, doi: 10.14445/23488379/ijeee-v10i1p116.
[12] R. W. Yatscoff and J. Hayter, "Bibliometric evaluations of modern Clinical Chemistry are needed," vol. 29, no. 10, pp. 1982–1983, 1983.
[13] D. Retno *et al.*, "BIBLIOMETRIC ANALYSIS OF NEURAL BASIS EXPANSION ANALYSIS FOR INTERPRETABLE TIME SERIES (N-BEATS) FOR RESEARCH TREND MAPPING," vol. 17, no. 2, pp. 1103–1112, 2023.
[14] L. Waltman, N. J. van Eck, and E. C. M. Noyons, "A unified approach to mapping and clustering of bibliometric networks," *J. Informetr.*, vol. 4, no. 4, pp. 629–635, 2010, doi: 10.1016/j.joi.2010.07.002.
[15] N. J. van Eck and L. Waltman, "Software survey: VOSviewer, a computer program for bibliometric mapping," *Scientometrics*, vol. 84, no. 2, pp. 523–538, 2010, doi: 10.1007/s11192-009-0146-3.
[16] V. P. Diodato and P. Gellatly, *Dictionary of Bibliometrics*. 1994. doi: 10.4324/9780203714133.
[17] M.-A. De Looze and J. Lemarie, "Corpus Relevance Through Co-Word Analysis :," *Scientometrics*, vol. 39, no. 3, pp. 267–280, 1997.
[18] S. COULTER, Neal; MONARCH, Ira; KONDA, "Software Engineering as Seen through Its Research Literature : A Study in Co- Word Analysis . CMU / SEI-95-TR-019 ESC-TR-95-019 An Evolutionary Perspective of Software Engineering Research Through Co-Word Analysis Neal Coulter Ira Monarch Suresh Konda Mar," *J. Am. Soc. Inf. Sci.*, vol. 4571, no. January, 1998, doi: 10.1002/(SICI)1097-4571(1998)49.
[19] J. Du and Y. Wu, "A bibliometric framework for identifying 'Princes' who wake up the 'sleeping beauty' in challenge-type scientific discoveries," *J. Data Inf. Sci.*, vol. 1, no. 1, pp. 50–68, 2016, doi: 10.20309/jdis.201605.
[20] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
[21] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," pp. 1–9, 2014, [Online]. Available: http://arxiv.org/abs/1412.3555
[22] A. Graves, "Generating Sequences With Recurrent Neural Networks," pp. 1–43, 2013, [Online]. Available: http://arxiv.org/abs/1308.0850
[23] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," *Proc. - 2016 31st Youth Acad. Annu. Conf. Chinese Assoc. Autom. YAC 2016*, pp. 324–328, 2017, doi: 10.1109/YAC.2016.7804912.
[24] G. Peng and Z. Yili, "Research on Forest Phenology Prediction Based on LSTM and GRU Model," *J. Resour. Ecol.*, vol. 14, no. 1, pp. 25–34, 2022, doi: 10.5814/j.issn.1674-764x.2023.01.003.
[25] N. Habbat, H. Anoun, and L. Hassouni, "A Novel Hybrid Network for Arabic Sentiment Analysis using fine-tuned AraBERT model," *Int. J. Electr. Eng. Informatics*, vol. 13, no. 4, pp. 801–812, 2021, doi: 10.15676/ijeei.2021.13.4.3.
[26] F. Shahid, A. Zameer, and M. Muneeb, "Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM," *Chaos, Solitons and Fractals*, vol. 140, p. 110212, 2020, doi: 10.1016/j.chaos.2020.110212.
[27] J. Heaton, *Applications of Deep Neural Networks with Keras*. 2020. [Online]. Available: http://arxiv.org/abs/2009.05673
[28] 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. April 2021, pp. 98–101, 2020, doi: 10.1109/IWECAI50956.2020.00027.