Recursive Journal of Informatics
Vol. 2, No. 1, March 2024
p-ISSN 2963-5551 https://journal.unnes.ac.id/journals/index.php/rji e-ISSN 2986-6588

Hyperparameter Tuning of Long Short-Term Memory Model for Clickbait Classification in News Headlines

Grace Yudha Satriawan[1], Budi Prasetiyo[2]
[1,2]Computer Science Department, Faculty of Mathematics and Natural Sciences,
Universitas Negeri Semarang, Indonesia

Abstract. The information available on the internet nowadays is diverse and moves very quickly. Information is becoming easier to obtain by the general public with the numerous online media outlets, including news portals that provide up-to-date information insights. Various news portals earn revenue from advertising using pay-per-click methods that encourage article writers to use clickbait techniques to attract visitors. However, the negative effects of clickbait include a decrease in journalism quality and the spread of hoaxes. This problem can be prevented by using text classification to classify clickbait in news titles. One method that can be used for text classification is a neural network. Artificial neural networks use algorithms that can independently adjust input coefficient weights. This makes this algorithm highly effective for modeling non-linear statistical data. The artificial neural network algorithm, especially the Long Short-Term Memory (LSTM), has been widely used in various natural language processing fields with satisfying results, including text classification. To improve the performance of the neural network model, adjustments can be made to the model's hyperparameters. Hyperparameters are parameters that cannot be obtained through data and must be defined before the training process. In this research, the Long Short-Term Memory (LSTM) model was used in clickbait classification in news titles. Sixteen neural network models were trained with different hyperparameter configurations for each model. Hyperparameter tuning was carried out using the random search algorithm. The dataset used was the CLICK-ID dataset published by William & Sari, 2020[1], with a total of 15,000 annotated data. The research results show that the developed LSTM model has a validation accuracy of 0.8030, higher than William & Sari's research, and a validation loss of 0.4876. Using this model, researchers were able to classify clickbait in news titles with fairly good accuracy.
Purpose: The study was to develop and evaluate a LSTM model with hyperparameter tuning for clickbait classification on news headlines. The thesis also aims to compare the performance of simple LSTM and bidirectional LSTM for this task.
Methods: This study uses CLICK-ID dataset and applies different text preprocessing techniques. The dataset later was used to build and train 16 LSTM models with different hyperparameters and evaluates them using validation accuracy and loss. This study uses random search for hyperparameter tuning.
Result: The results of the study show that the best model for clickbait classification on news headlines is a bidirectional LSTM model with one layer, 64 units, 0.2 dropout rate, and 0.001 learning rate. This model achieves a validation accuracy of 0.8030 and a validation loss of 0.4876. The results also show that hyperparameter tuning using random search can improve the performance of the LSTM models by avoiding zero probabilities and finding the optimal values for the hyperparameters.
Novelty: This study compares and analyzes the different preprocessing methods on text and the different configurations of the models to find the best model for clickbait classification on news headlines. The study also uses hyperparameter tuning to tune the model into the best model and finding the optimal values for the hyperparameters.

Keywords: Hyperparameter Tuning, Neural Network, Clickbait, Natural Language Processing
Received July 27, 2024 / **Revised** November 14, 2024 / **Accepted** March 13, 2024

This work is licensed under a Creative Commons Attribution 4.0 International License.

INTRODUCTION

The information available on the internet today is diverse and moves very quickly. Information is becoming increasingly easy to obtain by the general public with the proliferation of online media, one of which is news portals that provide insight into current information both locally and internationally. Most digital news portals make profits from advertisers who pay them to display ads on their website. This business model is commonly known as pay-per-click, where the news portal will receive payment every time an ad is clicked by a visitor [2], [3]. This requires writers to create news articles that are easily found with

[1]*Corresponding author.
Email addresses: graceyudha@students.unnes.ac.id (Satriawan)
DOI: 10.15294/rji.v2i1.71831

attractive headlines, so that the news portal can display ads they have to generate profits, including using clickbait techniques in their headlines [4].

Clickbait is content on a website that is intentionally created and designed to make readers interested in clicking on the displayed link. Clickbait content is usually spread on social media [5]. The working mechanism of clickbait is by arousing the curiosity of potential readers about something specific [6]. The lack of something specific creates a phenomenon called information gap and as long as the curiosity is not satisfied, it will continue to increase until it is satisfied [7]. This information gap is then utilized by news writers by creating headlines that do not reveal all the information from the main news completely. However, articles with clickbait headlines also create a bad reading experience for their readers and make them dissatisfied with the content of the article. In addition, clickbait can also be used to spread fake news [4].

The phenomenon of clickbait that often occurs is also supported by search engine algorithms based on the click-through rate (CTR), which is a comparison of how many people click on a link to how often the link is displayed [8], [9]. This algorithm can influence how high the ranking of a search result is on the search engine, and because news articles with headlines that use clickbait techniques usually have a high CTR, the news articles made with this clickbait model will get a high-ranking on-search results [10].

Clickbait can also cause a decrease in the quality of journalism because it prioritizes headlines that increase the number of visitors rather than the information contained in the news article itself. This can also trigger the spread of fake news in society, which can cause unrest in society. Therefore, a way to classify clickbait is needed to prevent the negative impacts of clickbait.

Clickbait classification in news headlines can use text classification. Text classification is a method of building a machine learning model that can classify new documents into pre-defined labels [11]. The text classification process also requires a data preprocessing stage before entering the model building stage [12]. The model used in text classification can use various algorithms. One algorithm that can be used for text classification is the neural network [12], [13]. Neural network or artificial neural network is a computational model inspired by the pattern of human brain neurons [14]–[16]. Artificial neural network uses algorithms that can independently adjust input coefficient weights. This makes the algorithm highly effective for modeling non-linear statistical data [17].

According to Sherstinsky (2020), the artificial neural network algorithm, especially Long Short-Term Memory (LSTM), has been widely used in various natural language processing fields with satisfactory results, ranging from language modeling, speech-to-text transcription, machine translation, and including text classification [18]. Naeem et al. (2020) applied the LSTM model to classify clickbait titles from articles with satisfactory results [19]. LSTM itself is divided into two different methods, namely simple LSTM and bidirectional LSTM. Simple LSTM is a type of LSTM that learns input from the beginning of the input to the end, and bidirectional LSTM learns input from two directions, namely from the beginning to the end and from the end to the beginning [20]. This learning method enables bidirectional LSTM to better understand the context of the input text.

To improve the performance of the neural network model, adjustments can be made to the model's hyperparameters [21]. Hyperparameters are parameters that cannot be obtained through data and must be defined before the training process [22]. To find the optimal hyperparameter model, the model can be trained using a search algorithm. The hyperparameter search algorithm commonly used is Random Search and Grid Search [21], [22]. Grid Search performs a brute-force hyperparameter search by searching for all combinations, while random search performs a random search and has a limited number of combinations. Neural networks can only understand numerical inputs, so title inputs must be converted into numbers. Titles are converted into a sequence of numbers that represent each word, then from those numbers, a word representation in the form of a vector that contains the meaning of the word is obtained so that the model can understand the word. This process is called word embedding [23]. Word embedding can be done using various methods, one of which is by training word embedding simultaneously with the model and finding word vectors through the backpropagation process. Another method is by using a pretrained word embedding model that has been trained with more words beforehand. One of the commonly used pretrained word embedding methods is the word2vec model [24]. Word2vec is a word embedding model developed by Google for language model analysis.

The datasets used in previous research were news headline datasets in English. Therefore, it is not yet known whether they can be applied to Indonesian news headline datasets. In 2020, William & Sari published CLICK-ID, which is the largest public dataset containing Indonesian news headlines and their experiments in making a classification model with the CNN and LSTM algorithms [1]. The CLICK-ID dataset consists of 46,517 news headlines divided into two parts, 15,000 annotated data and 31,517 unannotated data. The data is taken from 12 Indonesian news portals such as detikNews, Liputan6, Kapanlagi, Kompas, Tribunnews, Republika, Sindonews, Tempo, Wowkeren, Okezone, Fimela, dan Posmetro-Medan. From the problem above, the researcher wants to investigate the classification of Indonesian language news headlines using the Long Short-Term Memory (LSTM) model. The LSTM model will be created by searching for a combination of hyperparameters that can produce the best model. The best model will be determined by evaluation results using several metrics such as precision, recall, f1-score, accuracy, confusion matrix, and ROC and AUC values. The dataset used in this study is the CLICK-ID dataset provided publicly, and the results of previous studies are used as benchmarks. The dataset will be processed again with text preprocessing functions and transformed into word vectors using word embedding with the backpropagation process or using pre-trained word2vec according to the combination of hyperparameters. Based on the above explanation, the researcher decided to conduct research titled "Hyperparameter Tuning of Long Short-Term Memory Model for Clickbait Classification in News Headlines".

METHODS

In this study, a model is constructed using LSTM as the primary algorithm for clickbait classification. The model is divided into 16 different configurations, and the model's hyperparameters are adjusted using the random search algorithm. Subsequently, these 16 models are evaluated and compared to identify the one with the optimal hyperparameters.

The initial steps of the research involve preparing the dataset and corpus of the Indonesian language for training the word2vec model. The dataset used is in CSV format, containing text with clickbait and non-clickbait labels. Additionally, a separate dataset of Indonesian sentences is prepared for training the word2vec model. The next step is text preprocessing, which includes several techniques such as case folding, punctuation normalization, text normalization, stopword removal, and stemming. This preprocessing stage produces four different text preprocessing functions, which will be compared to understand their impact on the model training results.

Next, the data is duplicated to match the 33 text preprocessing functions and is then applied to each function. The dataset is subsequently divided into training and testing data in a 3:1 ratio. Tokenizers are created based on the training data to convert words into numerical values. Four different tokenizers are generated, corresponding to the text preprocessing functions. The following step involves applying the text preprocessing functions to the data used to train the word2vec model. The word2vec model is obtained from the training results and pre-trained models. During the training of the word2vec model, various configurations are used as part of the hyperparameter tuning process. A total of 8 word2vec models are utilized in this research, 6 of which are generated from the training results, and 2 are pre-trained models. These word2vec models are used in the model with the word2vec word embedding configuration.

For the clickbait classification model, 16 models are created based on different hyperparameter configurations. These models are trained using the random search algorithm, which searches for the optimal combination of hyperparameters with a maximum of 20 trials. Once the models are obtained, they are evaluated using the test data, and evaluation metrics such as precision, recall, F1-score, accuracy, and ROC-AUC are used. The evaluation results are then compared to identify the optimal hyperparameter configuration for effectively classifying clickbait.

Case Folding

The steps included in the text preprocessing process encompass several techniques, such as case folding, punctuation normalization, text normalization, stopword removal, and stemming. In this study, the process of case folding is employed as an essential step in the text preprocessing phase. The dataset contains text that is not consistently written in capital letters, which can significantly influence the model to be constructed due to its case-sensitive nature. For instance, if a title contains the word "NasDem" with both "N" and "D" capitalized, and another title includes the word "Nasdem" with only the initial "N" in capital letters, the model would treat "NasDem" and "Nasdem" as two distinct words, even though they have the

same meaning. To address this issue, case folding is applied to convert all data to lowercase, thereby standardizing the writing style of the titles within the dataset.

Punctuation Normalization

Punctuation normalization is also a crucial preprocessing technique applied to the text data. Typically, punctuation normalization involves removing punctuation marks from the text. However, an alternative approach is adopted here, where punctuation marks, specifically "!" and "?," are replaced with corresponding phrases "tanda seru" (exclamation mark) and "tanda tanya" (question mark). This replacement is based on the frequency of "!" and "?" occurrences in the labeled clickbait and non-clickbait dataset, as shown in Table 1.

Tabel 1. Distribution of exclamation marks "!" and question marks "?" in clickbait and non-clickbait data.

Data	Number of titles with exclamation mark "!"	Number of titles with exclamation mark "!"
Data with clickbait label	361	623
Data with non-clickbait label	23	12

The table clearly demonstrates that there is a significantly higher occurrence of "!" and "?" in the titles labeled as clickbait. By replacing these punctuation marks with corresponding phrases, the aim is to introduce new features that can act as discriminative indicators for clickbait classification. On the other hand, all other punctuation marks, such as " " #$%&'()*+,-./:;<=>@[]^_`{|}~," are removed from the text in this research. This approach to punctuation normalization is designed to streamline the text and create a consistent representation of the titles across the dataset.

Stopword Removal

In this study, the researchers utilized a list of 758 Indonesian stop words called "ID-Stopwords." However, it was observed that some words from the stop word list could actually be helpful in classifying clickbait content. For instance, the word "ternyata" (which means "apparently" in English) appeared 80 times in the word count data of clickbait, whereas it appeared only 10 times in the word count data of non-clickbait. This indicates that certain stop words should be removed from the list to retain essential features that aid in clickbait classification.

To address this, the researchers employed a method based on the probability of word occurrences in both datasets to select the stop words for removal. This method calculates the probability of word occurrences in the stop word list in both the clickbait and non-clickbait datasets. If the probability of a word's occurrence exceeds a threshold value (set at 0.7 in this study), the word is removed from the stop word list. Equations 1 and 2 represent the method used for selecting stop words based on this probability approach.

$$P(s \in c) = \frac{P(s \in c)}{(P(s \in c) + P(s \in n))} \quad (1)$$

$$P(s \in n) = \frac{P(s \in n)}{(P(s \in c) + P(s \in n))} \quad (2)$$

where:
P = the probability of a word.
s = a stop word.
c = clickbait dataset.
n = non-clickbait dataset.
After the selection process, the number of stop words in the list reduced to 673 words.

Text Preprocessing Application

the text preprocessing stage will generate four distinct datasets, each corresponding to the specific text preprocessing functions applied. All four functions will consistently involve lowercasing, punctuation normalization, and text normalization. However, they will differ in terms of the additional steps they employ. The first function will not remove stop words or apply stemming. The second function will omit stopword removal but will apply the stemming process. The third function will utilize stopword removal

but skip stemming. Finally, the fourth function will encompass both stopword removal and stemming processes.

Word2vec

The models in this study utilizes an embedding layer consisting of both Keras embedding layer and word2vec. In this research, the word2vec model is trained separately from the main model. Later on, the models are combined to search for an optimal method for clickbait classification. The corpus used to train the word2vec model is obtained from the Leipzig corpora collection [25] and the Wikipedia article corpus. The Leipzig corpora collection consists of one million sentences extracted from news portals in the year 2020. The corpus needs to undergo text preprocessing before training. Text preprocessing is necessary to ensure that the indices of vectors generated by the word2vec model align with the input from the clickbait title data. As the model's input utilizes four text preprocessing functions, the same text preprocessing functions are used for the word2vec corpus. After the preprocessing stage, the corpus is trained to create the word2vec model. In this study, the corpus is trained to create multiple models with different hyperparameter configurations. The hyperparameter configurations used include the vector dimensions and learning algorithms of the model. The vector dimensions for the model are selected from the set {100, 200, 300} [26], while the learning algorithms used are skip-gram and Continuous Bag-of-Words (CBOW).

Model Training

The research employs an LSTM-based model. It comprises 16 individual models, each with distinct configurations. These models undergo training using random search, with a maximum of 20 iterations (trials) to explore and find the best hyperparameters for each specific configuration. Based on the configurations, Table 2 displays the models and their respective configurations that will be trained during the study, while Table 3 shows the values used to fine-tune the hyperparameters.

Tabel 2. Model Configuration

Model	Dataset Preprocessing	LSTM	Embedding
Model #1	Without Stopword Removal, Without Stemming	Bidirectional LSTM	Keras Embedding
Model #2	Without Stopword Removal, Without Stemming	Bidirectional LSTM	Word2vec
Model #3	Without Stopword Removal, Without Stemming	Simple LSTM	Keras Embedding
Model #4	Without Stopword Removal, Without Stemming	Simple LSTM	Word2vec
Model #5	Without Stopword Removal, Stemming	Bidirectional LSTM	Keras Embedding
Model #6	Without Stopword Removal, Stemming	Bidirectional LSTM	Word2vec
Model #7	Without Stopword Removal, Stemming	Simple LSTM	Keras Embedding
Model #8	Without Stopword Removal, Stemming	Simple LSTM	Word2vec
Model #9	Stopword Removal, Without Stemming	Bidirectional LSTM	Keras Embedding
Model #10	Stopword Removal, Without Stemming	Bidirectional LSTM	Word2vec
Model #11	Stopword Removal, Without Stemming	Simple LSTM	Keras Embedding
Model #12	Stopword Removal, Without Stemming	Simple LSTM	Word2vec
Model #13	Stopword Removal, Stemming	Bidirectional LSTM	Keras Embedding
Model #14	Stopword Removal, Stemming	Bidirectional LSTM	Word2vec
Model #15	Stopword Removal, Stemming	Simple LSTM	Keras Embedding
Model #16	Stopword Removal, Stemming	Simple LSTM	Word2vec

Tabel 3 Hyperparameter Value

LSTM output dimension	{8k| k∈{1,2,...,8}}
LSTM layer count	{1,2,3}
LSTM regularizer	{0,005; 0,0005; 0,00005}
LSTM dropout	{0; 0,05; 0,1; 0,25; 0,5}
Embedding dimension	{100, 200, 300}
Learning rate	{0,001; 0,0001}

The model is created using Tensorflow and Keras, and the training process is conducted using the Keras Tuner library with the random search algorithm. The model will be trained with 10,050 data points and tested with 4,950 data points, using 15 epochs per trial.

RESULT AND DISCUSSION

The model is evaluated by examining its precision, recall, f1-score, accuracy, ROC, and AUC scores, as well as the confusion matrix, using the testing data. The confusion matrix, also known as the error matrix, is a table that displays the accuracy and correctness of the model's predictions [27][28][29]. Table 4 below is a more detailed evaluation of each of the 16 models that have been trained and adjusted with their respective hyperparameters.

The best model obtained in this research had a validation accuracy of 0.8030 and a validation loss of 0.4876. It used a bidirectional LSTM layer with 64 units, an embedding layer with 300 dimensions, a dropout rate of 0.2, a batch normalization layer, a global max pooling layer, and an Adam optimizer with a learning rate of 0.0001. The best model outperformed the previous studies using the same dataset, such as CNN (0.7893), LSTM (0.7913), and BiLSTM (0.7993). It also achieved comparable results with state-of-the-art models such as BERT (0.8087) and XLNet (0.8113).

Tabel 4. Configuration and Evaluation Results of the Model

Model	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16
Stopword removal									✓	✓	✓	✓	✓	✓	✓	✓
Stemming					✓	✓	✓	✓					✓	✓	✓	✓
LSTM Layer	Bidirectional	Bidirectional	Simple LSTM	Simple LSTM	Bidirectional	Bidirectional	Simple LSTM	Simple LSTM	Bidirectional	Bidirectional	Simple LSTM	Simple LSTM	Bidirectional	Bidirectional	Simple LSTM	Simple LSTM
LSTM Layer Count	1	3	1	3	1	3	1	2	1	2	1	1	1	1	1	2
LSTM Output Dimension	- Layer 1: 16	- Layer 1: 32 - Layer 2: 24 - Layer 3: 32	- Layer 1: 48	- Layer 1: 56 - Layer 2: 64 - Layer 3: 32	- Layer 1: 8	- Layer 1: 48 - Layer 2: 8 - Layer 3: 8	- Layer 1: 56	- Layer 1: 16 - Layer 2: 8	- Layer 1: 56	- Layer 1: 48 - Layer 2: 16	- Layer 1: 16	- Layer 1: 40	- Layer 1: 32	- Layer 1: 40	- Layer 1: 24	- Layer 1: 16 - Layer 2: 48
LSTM Regularizer	- Layer 1: 0,0005	- Layer 1: 0,00005 - Layer 2: 0,00005 - Layer 3: 0,00005	- Layer 1: 0,005	- Layer 1: 0,005 - Layer 2: 0,005 - Layer 3: 0,005	- Layer 1: 0,0005	- Layer 1: 0,0005 - Layer 2: 0,005 - Layer 3: 0,005	- Layer 1: 0,0005	- Layer 1: 0,005 - Layer 2: 0,005	- Layer 1: 0,0005	- Layer 1: 0,005 - Layer 2: 0,0005	- Layer 1: 0,0005	- Layer 1: 0,00005	- Layer 1: 0,00005	- Layer 1: 0,00005	- Layer 1: 0.0005	- Layer 1: 0,0005 - Layer 2: 0,005
LSTM Dropout	- Layer 1: 0,5	- Layer 1: 0,5 - Layer 2: 0,1 - Layer 3: 0,0	- Layer 1: 0,1	- Layer 1: 0,5 - Layer 2: 0,5 - Layer 3: 0,25	-Layer 1: 0,05	- Layer 1: 0,25 - Layer 2: 0,25 - Layer 3: 0,0	- Layer 1: 0,05	- Layer 1: 0,0 - Layer 2: 0,0	- Layer 1: 0,05	- Layer 1: 0,25 - Layer 2: 0,0	- Layer 1: 0,05	- Layer 1: 0,5	- Layer 1: 0,05	- Layer 1: 0,1	- Layer 1: 0,5	- Layer 1: 0,1 - Layer 2: 0,5
Embedding Dimension	300	300	300	300	100	200	100	100	300	300	200	100	300	200	200	100
Word2vec Corpus		Leipzig Corpora Collection		Leipzig Corpora Collection		Leipzig Corpora Collection		Leipzig Corpora Collection		Leipzig Corpora Collection		Leipzig Corpora Collection		Leipzig Corpora Collection		Wikipedia
Word2vec Training Algorithm		Skip-gram		CBOW		CBOW		Skip-gram		Skip-gram		Skip-gram		Skip-gram		CBOW
Learning Rate	0,0001	0,001	0,0001	0,001	0,001	0,001	0,001	0,001	0,0001	0,001	0,001	0,001	0,0001	0,001	0,001	0,001
Validation Accuracy	≈0,7781	≈0,8020	≈0,7900	**≈0,8030**	≈0,7881	≈0,7881	≈0,7980	≈0,7930	≈0,7672	≈0,7771	≈0,7632	≈0,7811	≈0,7632	≈0,7761	≈0,7711	≈0,7761
Validation Loss	≈0,5133	≈0,5526	≈0,5704	≈0,4876	≈0,607	≈0,4944	≈0,6331	≈0,4945	≈0,5982	≈0,516	≈0,62	≈0,4859	≈0,5496	≈0,5133	≈0,5392	≈0,5116
Best Threshold	≈0,448	≈0,3611	≈0,4603	≈0,3472	≈0,4712	≈0,3723	≈0,4716	≈0,4354	≈0,4382	≈0,4259	≈0,4777	≈0,4568	≈0,4343	≈0,4204	≈0,4302	≈0,4268
AUC	≈0,8515	≈0,8435	≈0,8573	≈0,8610	≈0,8550	≈0,8357	≈0,8593	≈0,8426	≈0,8220	≈0,8300	≈0,8267	≈0,8400	≈0,8227	≈0,8253	≈0,8145	≈0,8053

CONCLUSION

This study develops and evaluates a LSTM model with hyperparameter tuning for clickbait classification on news headlines. The study compares and analyzes the different preprocessing methods on text and the different configurations of the models. The study also uses random search algorithm for hyperparameter tuning. The best model is a bidirectional LSTM model with one layer, 64 units, 0.2 dropout rate, and 0.001 learning rate. This model achieves a validation accuracy of 0.8030 and a validation loss of 0.4876. The model can help readers to avoid misleading news and improve journalism quality. Compared to previous studies using the same dataset, the evaluation results obtained by the researchers demonstrate a satisfactory performance. Although the outcomes of the research are already promising, there is still room for further improvement to achieve even more optimal results. This study can serve as a valuable benchmark for future research endeavors, providing a solid reference point for comparative analysis and enhancing the understanding of clickbait classification methods.

REFERENCES

[1] A. William and Y. Sari, "CLICK-ID: A Novel Dataset for Indonesian Clickbait Headlines," *Data Brief*, vol. 32, p. 106231, Oct. 2020, doi: 10.1016/J.DIB.2020.106231.

[2] I. Beleslin and B. R. Njegovan, "Clickbait Titles: Risky Formula for Attracting Readers and Advertisers," in *XVII International Scientific Conference on Industrial Systems*, 2017.

[3] H. T. Zheng, J. Y. Chen, X. Yao, A. K. Sangaiah, Y. Jiang, and C. Z. Zhao, "Clickbait Convolutional Neural Network," *Symmetry (Basel)*, vol. 10, no. 5, May 2018, doi: 10.3390/sym10050138.

[4] C. I. Coste and D. Bufnea, "Advances in Clickbait and Fake News Detection Using New Language-Independent Strategies," *Journal of Communications Software and Systems*, vol. 17, no. 3, 2021, doi: 10.24138/jcomss-2021-0038.

[5] M. Potthast, S. Kopsel, B. Stein, and M. Hagen, "Clickbait Detection," pp. 810–817, 2016, doi: 10.1007/978-3-319-30671-1.

[6] A. Chakraborty, B. Paranjape, S. Kakarla, and N. Ganguly, "Stop Clickbait: Detecting and Preventing Clickbaits in Online News Media," in *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2016*, 2016. doi: 10.1109/ASONAM.2016.7752207.

[7] G. Loewenstein, "The Psychology of Curiosity: A Review and Reinterpretation.," *Psychol Bull*, vol. 116, no. 1, pp. 75–98, Jul. 1994, doi: 10.1037/0033-2909.116.1.75.

[8] J. Kuiken, A. Schuth, M. Spitters, and M. Marx, "Effective Headlines of Newspaper Articles in a Digital Environment," *Digital Journalism*, vol. 5, no. 10, 2017, doi: 10.1080/21670811.2017.1279978.

[9] W. Wang, F. Feng, X. He, H. Zhang, and T. S. Chua, "Clicks Can be Cheating: Counterfactual Recommendation for Mitigating Clickbait Issue," in *SIGIR 2021 - Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021. doi: 10.1145/3404835.3462962.

[10] P. Biyani, K. Tsioutsiouliklis, and J. Blackmer, "'8 Amazing Secrets for Getting More Clicks': Detecting Clickbaits in News Streams Using Article Informality," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, Feb. 2016, doi: 10.1609/aaai.v30i1.9966.

[11] M. M. Mirończuk and J. Protasiewicz, "A Recent Overview of the State-of-the-Art Elements of Text Classification," *Expert Systems with Applications*, vol. 106. 2018. doi: 10.1016/j.eswa.2018.03.058.

[12] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text Classification Algorithms: A Survey," *Information (Switzerland)*, vol. 10, no. 4, 2019, doi: 10.3390/info10040150.

[13] H. Hassani, C. Beneki, S. Unger, M. T. Mazinani, and M. R. Yeganegi, "Text Mining in Big Data Analytics," *Big Data and Cognitive Computing*, vol. 4, no. 1, 2020, doi: 10.3390/bdcc4010001.

[14] U. Güçlü and M. A. J. van Gerven, "Modeling the Dynamics of Human Brain Activity with Recurrent Neural Networks," *Front Comput Neurosci*, vol. 11, 2017, doi: 10.3389/fncom.2017.00007.

[15] IBM Cloud Education, "What are Neural Networks? | IBM," *Ibm*. 2020.

[16] S. Sharma, S. Sharma, and A. Athaiya, "Activation Functions in Neural Networks," *International Journal of Engineering Applied Sciences and Technology*, vol. 04, no. 12, 2020, doi: 10.33564/ijeast.2020.v04i12.054.

[17] O. Agasi, J. Anderson, A. Cole, M. Berthold, M. Cox, and D. Dimov, “What is an Artificial Neural Network (ANN)? - Definition from Techopedia,” *Techopedia*. 2018.
[18] A. Sherstinsky, “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network,” *Physica D*, vol. 404, 2020, doi: 10.1016/j.physd.2019.132306.
[19] B. Naeem, A. Khan, M. O. Beg, and H. Mujtaba, “A Deep Learning Framework for Clickbait Detection on Social Area Network Using Natural Language Cues,” *J Comput Soc Sci*, vol. 3, no. 1, 2020, doi: 10.1007/s42001-020-00063-y.
[20] S. Cornegruta, R. Bakewell, S. Withey, and G. Montana, “Modelling Radiological Language with Bidirectional Long Short-Term Memory Networks,” 2016.
[21] S. K. Palaniswamy and R. Venkatesan, “Hyperparameters Tuning of Ensemble Model for Software Effort Estimation,” *J Ambient Intell Humaniz Comput*, vol. 12, no. 6, 2021, doi: 10.1007/s12652-020-02277-4.
[22] L. Yang and A. Shami, “On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice,” Jul. 2020, doi: 10.1016/j.neucom.2020.07.061.
[23] B. Wang, A. Wang, F. Chen, Y. Wang, and C. C. J. Kuo, “Evaluating Word Embedding Models: Methods and Experimental Results,” *APSIPA Transactions on Signal and Information Processing*, vol. 8. 2019. doi: 10.1017/ATSIP.2019.12.
[24] Google Code Archive, “word2vec,” Jul. 30, 2013. https://code.google.com/archive/p/word2vec/ (accessed Nov. 02, 2022).
[25] D. Goldhahn, T. Eckart, and U. Quasthoff, “Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages,” 2012. [Online]. Available: http://corpora.uni-leipzig.de
[26] M. Basaldella, E. Antolli, G. Serra, and C. Tasso, “Bidirectional LSTM Recurrent Neural Network for Keyphrase Extraction,” in *Communications in Computer and Information Science*, Springer Verlag, 2018, pp. 180–187. doi: 10.1007/978-3-319-73165-0_18.
[27] C. Sammut and G. I. Webb, *Encyclopedia of Machine Learning*. Boston, MA: Springer US, 2010. doi: 10.1007/978-0-387-30164-8.
[28] S. V Stehman, “Selecting and Interpreting Measures of Thematic Classification Accuracy,” OElsevier Science Inc, 1997.
[29] M. Vakili, M. Ghamsari, and M. Rezaei, “Performance Analysis and Comparison of Machine and Deep Learning Algorithms for IoT Data Classification,” 2020.