Fake News Detection Using a Hybrid CNN-LSTM Deep Learning Model

Ferhati Bouchra[a1], Kadek Indah Permata[a2], Putu Rheya Ananda Savitri[a3]

[a]Program Studi Teknologi Informasi, Fakultas Teknik, Universitas Udayana
Jl. Raya Kampus Udayana, Bali, Indonesia
[1]ferhatibouchra@gmail.com
[2]indahpermatakadek83@gmail.com
[3]rheya.ananda@gmail.com

Abstract

Hoax or fake news refers to information that is deliberately created or presented without considering the truth in order to attract readers' attention. Its distribution really depends on the response of readers, whether they spread it without verifying the truth or deliberately repost it via social media. As technology and the internet grow, social media users become more vulnerable to fake news. This research proposes a system to detect the truth of news using a Deep Learning algorithm combining CNN and LSTM with GloVe word representation. Implementing this model in website applications makes it easy for users to detect fake news. The evaluation obtained shows that the use of a combination of CNN and LSTM as well as GloVe produces an accuracy of 98% with precision, recall and F-Score values of around 98% each, which shows the model is able to classify fake and genuine news.

Keywords: *Fake News Detection, Deep Learning, CNN, LSTM, Glove.*

1. Introduction

Fake news, also known as hoaxes, refers to information that is deliberately created or presented to attract readers' attention without considering the truth, with the intention of convincing readers. The spread of hoaxes is highly dependent on the response of the individuals who read it, whether they spread it without verifying the truth or deliberately resend it to many people through various social media platforms. With the development of technology and the internet, people are increasingly using social media as a means of communication. However, unknowingly, the phenomenon of fake news or hoaxes has also become rampant among social media users over time (Anisa et al., 2023).

Some cases include the spread of news in 2019 about aftershocks and the potential for a tsunami in Palu. This news never stopped and even managed to disturb the community, especially for the victims affected by the disaster. Even the hoax about the issue of aftershocks in Palu is included in the data of the Ministry of Communication and Information Technology as the 2nd most impactful hoax content spread in Indonesia (Poso, 2020). Another example of fake news is the circulation of news about fake eggs. The news stated that the eggshell was made of paper and the yolk was made of silicone. This condition can certainly cause anxiety to those who believe it (Wicaksono, 2023). Another example is a video upload showing a man inserting plastic sheets into a machine and then producing white flakes resembling rice. The upload was also accompanied by a narration "Rice factory from plastic". However, in fact, the claim that shows a man making plastic rice using a shredder is wrong. Reported from kompas.com, the video about fake rice claimed to be sourced from a factory in China is not true. The machine that appears in the video is a plastic shredder commonly used to recycle plastic. The plastic is shredded so that it is easy to store, move, and recycle (Fakta, 2023).

According to Kominfo data, hoaxes continue to increase from year to year. This is reinforced by data from During the first quarter of 2023, the Ministry of Communication and Information

Technology has identified 425 hoax issues circulating on websites and digital platforms. This number is higher than in the first quarter of 2022 which reached 393 hoax issues (Kominfo, 2023).

The government has implemented various steps to overcome the spread of fake news. One step is the establishment of the Anti-Hoax Community Movement by the Ministry of Communication and Information, with the aim of educating all levels of society to be more vigilant against invalid information. Even so, fake news is still widespread in the digital space, including through short messages, news articles, and social media platforms. A survey conducted by the Indonesian Telematics Society (Mastel) shows that only 16.20% of respondents were able to directly distinguish the existence of fake news. In addition, around 21.80% of respondents felt difficulty in verifying the truth of news (Kurniawan & Mustikasari, 2021).

Several studies have been conducted to analyze factual news and fake news. The study conducted by Antonius & Metty developed research to determine fake news in the Indonesian language using the CNN and LSTM methods. The results of this study show that the CNN method has an accuracy test, precision and recall rate of 0.88, while the LSTM method has an accuracy test, precision of 0.84 and recall of 0.83. the prediction results obtained by CNN are all true, while the prediction results obtained by LSTM there is 1 wrong prediction. Based on the evaluation and testing results, the CNN method shows better performance than the LSTM method (Tama & Sibaroni, 2023).

The next study by Fauzaan & Yuliant analyzed fake news on social media Twitter content using the CNN method by comparing the weighting features used from Term Frequency Inverse Document Frequency (TF-IDF) and Term Frequency-Relevance Frequency (TF-RF). The highest accuracy is obtained in the Term Frequency-Relevance Frequency (TF-RF) weighting feature with an accuracy of 84.11%, while in the Term Frequency Inverse Document Frequency (TF-IDF) weighting feature with an accuracy of 80.29% (Tama & Sibaroni, 2023).

The next study by Dwi et al detected online hoax news Covid-19 using the Hybrid LSTM and SVM method. The results of this study are that LSTM produces representative text features which are then used for the news classification process by SVM which results in a percentage accuracy value of 94% (Anisa et al., 2023). Based on the problem description and previous research, this study aims to implement deep learning to detect fake news using English.

2. Research Methods

This research aims to identify fake news from various news sources and provide accurate information to users. Our research uses a combination of two Deep Learning models, namely Convolutional Neural Network (CNN) and Long-short term Memory (LSTM), to perform fake news identification. Additionally, text data will be represented using GloVe embedding.

The stages involved in creating the fake news detection application are data collection, data preprocessing, word embedding loading, model training, model saving, and model implementation on a website interface. This will make it easier for users to utilize the fake news detection application. An illustration of this overview can be seen in Figure 1.

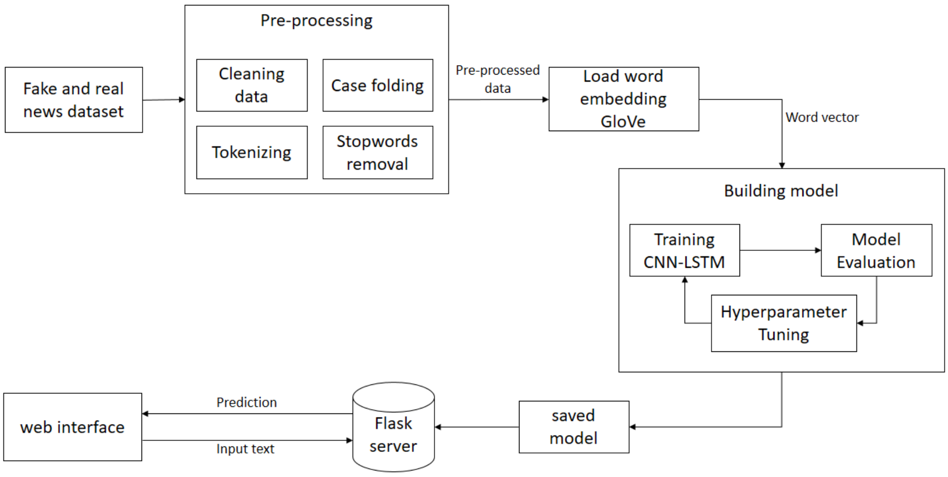


Figure 1. System Design

2.1. Data Preprocessing

The data analyzed in this research comes from the WELFake dataset. The WELFake dataset is a news data collection consisting of 72,134 articles, of which 35,028 are real news and 37,106 are fake news. This dataset consists of 4 combined news data namely from Kaggle, McIntire, Reuters, BuzzFeed Political. This is intended to prevent overfitting of the classifier. The collected data is then processed to make it easier for the model to learn. The preprocessing process consists of 4 stages, namely data cleaning, case folding, tokenizing, and stopword removal. All of these processes are run sequentially.

2.2. Word Embedding

Word embedding is one of the natural language processing techniques to represent words in the form of numerical vectors. So in short, this word embedding will change words into numbers so that the meaning of words in text can be understood and processed. This research uses Global Vectors for Word Representation (GloVe) for word representation based on statistical relationships between words in a large corpus text. GloVe tries to understand how words relate to each other by seeing how often they appear together in text. So, not only trying to guess the words that appear together, but also trying to create a numerical representation (vector) for words that can help in guessing how often those words appear together in large text. So, GloVe creates word vectors that help in understanding how words are related in real text (Fauzy & Erwin Budi Setiawan, 2023).

2.3. LSTM

Long Short-Term Memory (LSTM) is a deep learning algorithm designed to overcome the gradient descent problem that arises in traditional RNNs. LSTM is capable of processing, predicting, and classifying data based on time series, such as speech recognition, machine translation, and healthcare. The fundamental difference between LSTM and RNN is that LSTM can predict data based on information that has been stored for a long time [19]. LSTM can understand the context and order of words in a sentence, resulting in better translations. So, LSTM is an important tool in the development of complex NLP models that can handle sequential data well. The three gates that regulate features in the LSTM model are the input gate, the forget gate, and the output gate. The input gate serves to allow new data to enter the cell state. The forget gate deletes irrelevant information from the cell state. The output gate, which controls the information taken from the cell state, then determines what will be the next hidden state. The LSTM model can automatically store or delete recorded memory using these gates (Widayat, 2021).

2.4. CNN

CNNs are a special type of neural network that's particularly good at finding patterns in data with a layered structure, like images. They achieve this by using small filters that learn to recognize basic features, which are then combined to form more complex patterns. This layered approach makes CNNs simpler and more efficient compared to other neural networks.

A CNN is built with different layers: an input layer for the data, hidden layers that process the information, and an output layer for the final results. The hidden layers contain convolutional layers, where a filter slides across the data, creating a "feature map" that highlights important details. Additional layers like pooling layers can further refine the information, and fully connected layers allow the network to make final decisions (Kukuh Jaluwana et al., 2022).

2.5. Website Application

A website application is built to make it easier for users to utilize this fake news detection application. The tools used to construct the website are Flask as the back-end and JavaScript as the front-end. Flask is a Python-based web framework. Flask is known as one of the lightweight and minimalist frameworks. JavaScript, on the other hand, is a programming language used to create website interfaces. JavaScript runs on the client-side and is used to add interactivity to web pages.

The application has 2 features that can be used. Users can input news that they want to predict in the form of text. In addition, users can input in the form of a URL to check the news, with the help of the Newspaper library. Newspaper or Newspaer3k is a Python library for extracting articles that uses an intelligent parser and Natural Language Processing techniques to parse keywords, summaries, and so on from an article or news item. In this application, Newspaper will extract news from a URL into text for further prediction.

3. Result and Discussion

3.1. Data Preprocessing

Newspaper or Newspaer3k is a Python library for extracting articles that uses an intelligent parser and Natural Language Processing techniques to parse keywords, summaries, and so on from an article or news item. In this application, Newspaper will extract news from a URL into text for further prediction.

3.2. Preprocessing

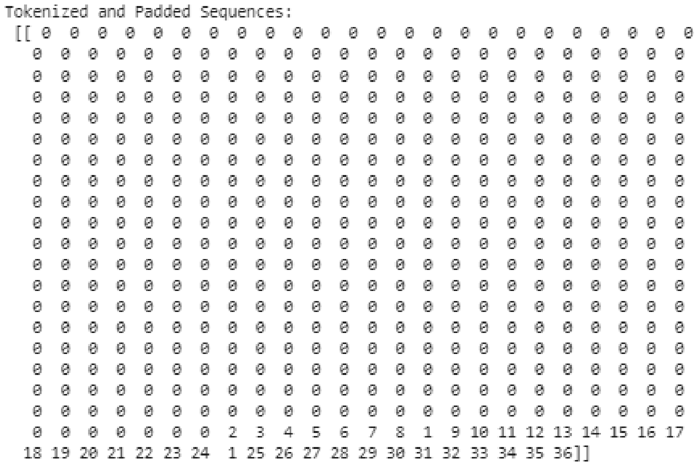
The preprocessing that has been carried out consists of four techniques, namely filtering, casefolding, tokenizing, and stopwords. The first step in the preprocessing process is data cleaning. The purpose of this stage is to clean the raw data or remove elements that can reduce the meaning of a text. Elements that are removed such as punctuation, numbers, double spaces, URLs, and mentions (@) are elements that are removed. Case folding is the process of converting all capital letters in a document to lowercase letters. The accepted letters are the letters 'a' to 'z'. Characters other than letters (punctuation and numbers) will be removed and considered delimiters which are sequences of one or more characters used to determine the separator. Tokenizing or parsing is the stage of cutting the input string based on each word that makes it up. At this stage, a sentence is cut into a word by analyzing a group of data by separating the word and determining the structure of each word. This stage also includes the process of removing numbers, removing punctuation such as symbols and punctuation, and removing whitespace. Stop Word is the stage of taking important words from the tokenizing results. The algorithm used is stoplist (removing unimportant words) or wordlist (storing important words) which are words that are not descriptive. Examples of stopwords are "and", "which", "on", and so on. The example of the preprocessing result can be seen in Table 1.

Table 1. Preprocessing Example

Text	Preprocessing Step	Result
Social Media Erupts After Influencer Posts Controversial Photo with Endangered Species - Wildlife Experts Urge Fans to @RespectNature	Filering	social media erupts after influencer posts controversial photo with endangered species wildlife experts urge fans to
	Case Folding	social media erupts after influencer posts controversial photo with endangered species wildlife experts urge fans to
	Tokenization	['social', 'media', 'erupts', 'after', 'influencer', 'posts', 'controversial', 'photo', 'with', 'endangered', 'species', 'wildlife', 'experts', 'urge', 'fans', 'to']
	Stopwords Removal	social media erupts influencer posts controversial photo endangered species wildlife experts urge fans

3.3. Word Embedding

The word embedding used in this research utilizes the GloVe model. The GloVe model employed is a pre-trained model. The GloVe model is trained using 60 billion tokens and has a dimension of 50, so that each word in the vocabulary is represented as a 50-dimensional vector. The GloVe model is applied after the data has undergone the preprocessing stage. The use of GloVe begins with tokenization and padding of the data, which aims to tokenize each word and equalize the length of the data after it becomes a sequence by adding "0" to the sequence. After tokenization and padding, the GloVe implementation proceeds by creating an embedding matrix according to the vocabulary that has been created and applying it to the data. An example of the appearance of data with a maximum length of 500 can be seen in Figure 2.

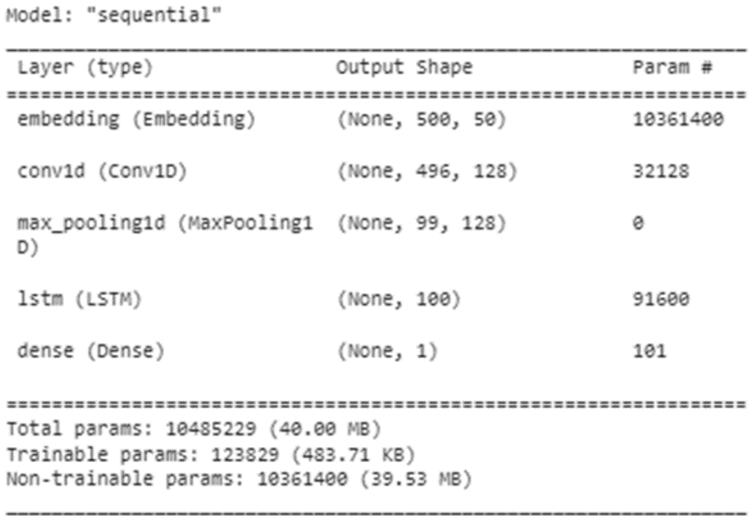


```
Tokenized and Padded Sequences:
 [[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    0  0  0  0  0  0  0  2  3  4  5  6  7  8  1  9 10 11 12 13 14 15 16 17
   18 19 20 21 22 23 24  1 25 26 27 28 29 30 31 32 33 34 35 36]]
```

Figure 2. Tokenized and Padded Sequences

3.4. Model Training and Evaluation

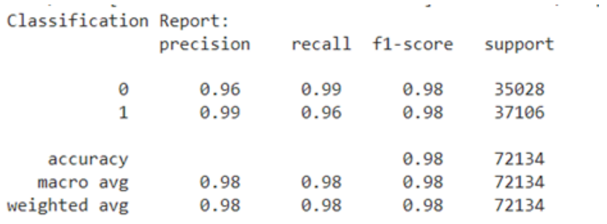
The model is built using Keras and has several layers that play an important role in processing and understanding text. Figure 3 illustrates the architecture of the model used. First, the Embedding layer converts words into vectors with a specific dimension. Then, the Convolutional Neural Network (CNN) layer is used to extract features from the vectors generated by the embedding layer. The MaxPooling Layer follows to reduce the data dimension by taking the maximum value in a specific window. Next, the Long Short-Term Memory (LSTM) layer functions to capture temporal patterns and relationships between words in the text sequence. Finally, the Dense layer, as the output layer, generates predictions using the sigmoid activation function for binary classification tasks. The model is compiled with the 'adam' optimizer, 'binary_crossentropy' loss function, and accuracy metric. After training with training data, model evaluation is performed using test data to measure model performance in terms of loss and accuracy.



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 500, 50)           10361400

 conv1d (Conv1D)             (None, 496, 128)          32128

 max_pooling1d (MaxPooling1  (None, 99, 128)           0
 D)

 lstm (LSTM)                 (None, 100)               91600

 dense (Dense)               (None, 1)                 101

=================================================================
Total params: 10485229 (40.00 MB)
Trainable params: 123829 (483.71 KB)
Non-trainable params: 10361400 (39.53 MB)
_________________________________________________________________
```

Figure 3. Model Architecture

The classification model shows very good performance. The model is able to correctly classify 98% of all samples, including 99% positive samples and 96% negative samples. The F-score of the model is also high, namely 98% for both classes. The evaluation can be seen in Figure 4.



```
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     35028
           1       0.99      0.96      0.98     37106

    accuracy                           0.98     72134
   macro avg       0.98      0.98      0.98     72134
weighted avg       0.98      0.98      0.98     72134
```

Figure 4. Model Evaluation

For more evaluation, this study utilizes LIME Explainer (Local Interpretable Model-agnostic Explanations) on a text document with ID 2 for model analysis. In the LIME analysis results in Figure 5, it is observed that the feature "breitbart" has a negative weight of -0.2325. This indicates that the presence of the word "breitbart" in the text has a negative impact on the model's prediction for the "real" class. The negative weight on "breitbart" suggests that in the training data, texts that include the word "breitbart" were more often associated with the "fake" category. Consequently, the model views "breitbart" as a strong feature in differentiating between news that is likely to be "fake" rather than "real". Overall, the model is more likely to predict the "fake" class for this document, and provides information about the keywords that have a significant impact on the model's prediction, such as "breitbart" and "iger".

3.

Ne cles that uses an intelligent
pa vords, summaries, and so on
fro xtract news from a URL into
te

3.

Th ebsite application utilizes HTML and CSS. The
ch sers with a clean and responsive interface. The

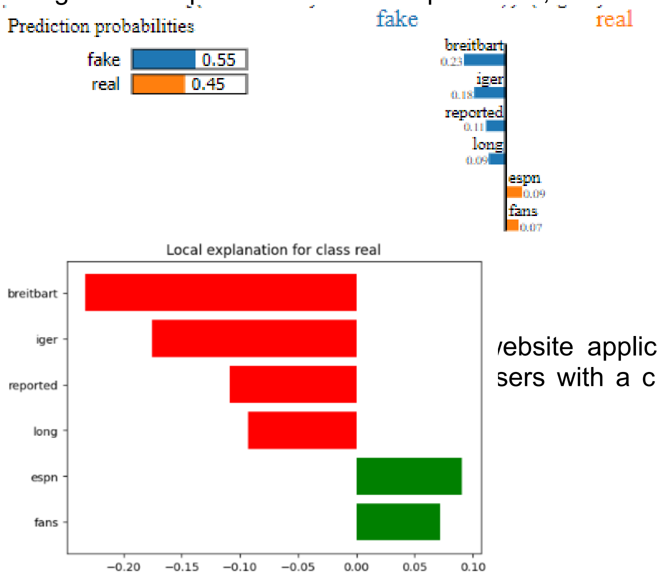


Figure SEQ Figure * ARABIC 5. LIME Analysis Example

provided features allow users to perform news detection by selecting either to input text or the news URL. The application will then directly predict and display whether the news is fake or not and show some excerpts of the news.

3.5.2. Back-end

The application employs the Flask framework as its back end. It utilizes several functions, namely preprocess_text(), scrape_news(), and predict(), which are loaded in a Python file. Additionally, the pre-trained model used in the application is loaded in the same file. The load_model function from the TensorFlow library and pickle.load from the Pickle library are used for model deployment. The preprocess_text() function cleanses the news text from unnecessary characters and converts the text to lowercase. The scrape_news() function downloads or extracts news text from the URL inputted by the user using the newspaper3k library. The predict() function is used for the prediction process of the extracted or directly inputted news text. The process involves receiving input data, validating the input, preprocessing the news text using the previously initialized functions and the loaded tokenizer to convert the text into a sequence of numeric tokens, and adjusting the text length to the maximum length of 500. Subsequently, the text is predicted using the pre-loaded model based on a threshold (0.5) and displayed back to the application's front end.

4. Conclusion

This research employs four text preprocessing techniques: filtering, case folding, tokenization, and stopwords removal to prepare the data for further analysis. The model construction, utilizing a combination of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM), demonstrates good performance, achieving an accuracy rate of 98%. The developed web application, using Flask as the backend, facilitates user-friendly detection of fake news through text or URL input. The analysis using LIME Explainer on a specific document reveals several keywords that significantly impact the model's prediction, indicating the model's tendency to associate these words with fake news. Overall, this research successfully combines various text processing techniques to develop an effective fake news detection application.

References

Anisa, D. F. N., Mukhlash, I., & Iqbal, M. (2023). Deteksi Berita Online Hoax Covid-19 Di Indonesia Menggunakan Metode Hybrid Long Short Term Memory dan Support Vector Machine. *Jurnal Sains Dan Seni ITS*, *11*(3). https://doi.org/10.12962/j23373520.v11i3.83227

Fakta, T. C. (2023, October 9). [HOAKS] Video Pabrik Beras Palsu Terbuat dari Plastik. *KOMPAS TV*. https://www.kompas.com/cekfakta/read/2023/10/09/185900082/-hoaks-video-pabrik-beras-palsu-terbuat-dari-plastik

Fauzy, A. R. I., & Erwin Budi Setiawan. (2023). Detecting Fake News on Social Media Combined with the CNN Methods. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *7*(2), 271–277. https://doi.org/10.29207/resti.v7i2.4889

Kominfo. (2023, April 6). Triwulan Pertama 2023, Kominfo Identifikasi 425 Isu Hoaks. *Kominfo*. https://www.kominfo.go.id/content/detail/48363/siaran-pers-no-50hmkominfo042023-tentang-triwulan-pertama-2023-kominfo-identifikasi-425-isu-hoaks/0/siaran_pers

Kukuh Jaluwana, G. A. M., Gusti Made Arya Sasmita, & I Made Agus Dwi Suarjaya. (2022). Analysis of Public Sentiment Towards Goverment Efforts to Break the Chain of Covid-19 Transmission in Indonesia Using CNN and Bidirectional LSTM. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *6*(4), 511–520. https://doi.org/10.29207/resti.v6i4.4055

Kurniawan, A. A., & Mustikasari, M. (2021). Implementasi Deep Learning Menggunakan Metode CNN dan LSTM untuk Menentukan Berita Palsu dalam Bahasa Indonesia. *Jurnal Informatika Universitas Pamulang*, *5*(4), 544. https://doi.org/10.32493/informatika.v5i4.6760

Poso, P. (2020, January 8). Gempa Susulan di Palu Jadi Hoaks Paling Berdampak di Indonesia. *Kumparan*. https://kumparan.com/paluposo/gempa-susulan-di-palu-jadi-hoaks-paling-berdampak-di-indonesia-1sbOfy3oz1j/full

Tama, F. R., & Sibaroni, Y. (2023). Fake News (Hoaxes) Detection on Twitter Social Media Content through Convolutional Neural Network (CNN) Method. *JINAV: Journal of Information and Visualization*, *4*(1), 70–78. https://doi.org/10.35877/454ri.jinav1525

Wicaksono, P. E. (2023, September 2). Kumpulan Hoaks Telur Palsu, Simak Faktanya. *Liputan6*. https://www.liputan6.com/cek-fakta/read/5269695/kumpulan-hoaks-telur-palsu-simak-faktanya

Widayat, W. (2021). Analisis Sentimen Movie Review menggunakan Word2Vec dan metode LSTM Deep Learning. *Jurnal Media Informatika Budidarma*, *5*(3), 1018. https://doi.org/10.30865/mib.v5i3.3111