JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 4, December 2025, Pp. 3124-3136

IMPLEMENTATION OF WORD EMBEDDING IN DETECTING POLITICAL FAKE NEWS IN INDONESIA USING LONG SHORT-TERM MEMORY ALGORITHM

Anggit Rianansyah*[1)], Ema Utami[2)], Dhani Ariatmanto[3)]
1. Universitas Amikom Yogyakarta, Indonesia
2. Universitas Amikom Yogyakarta, Indonesia
3. Universitas Amikom Yogyakarta, Indonesia

Article Info
Keywords: Long Short-Term Memory, Word Embedding, Hoax, K-fold cross-validation.

Article history:
Received 18 September 2024
Revised 10 October 2024
Accepted 31 October 2024
Available online 1 December 2025

DOI :
https://doi.org/10.29100/jipi.v10i4.6680

* Corresponding author.
Anggit Rianansyah
E-mail address:
anggit.rianansyah2276@students.amikom.ac.id

ABSTRACT
This research investigates the use of word embedding techniques to detect political fake news in Indonesia by utilizing the Long Short-Term Memory (LSTM) algorithm. The spread of fake news, particularly in the political realm, poses significant challenges to public trust and the integrity of information. To address these challenges, we employed a dataset of political news articles and applied word embedding to convert the text into a numerical format that represents the semantic relationships between words. The LSTM algorithm, known for its ability to process and learn from sequential data, was then used to identify patterns indicative of fake news. Our model demonstrated satisfactory accuracy, with the LSTM algorithm without word embedding achieving an average accuracy of 67%, while the application of word embedding (Word2Vec, Glove, and FastText) resulted in average accuracies of 84%, 81%, and 86%, respectively. These findings confirm that combining word embedding with LSTM is effective in detecting fake news. This research contributes to ongoing efforts to combat misinformation in Indonesia by providing a robust tool to enhance the reliability of news in the digital age. Further developments, such as the integration of additional linguistic features and the expansion of the dataset, are expected to improve the model's performance and adaptability across various contexts.

.

I. INTRODUCTION

Indonesia is one of the countries with the largest number of internet users in the world, with 171 million people having access to the internet. Of that number, around 95% use the internet to access various social media platforms [1]. In the continuously evolving digital era, information can be easily disseminated through various social media and online news platforms. Technology plays a crucial role in addressing this social issue, particularly in detecting and combating fake news. Technologies like deep learning and text classification are essential in identifying complex patterns in text data, helping to distinguish fake news from legitimate news. These technological innovations are becoming increasingly vital in maintaining the integrity and accuracy of information in society.

The Indonesian government has taken decisive action in addressing the threat of fake news, such as restricting access to social media during mass protests in front of the Bawaslu office on May 22, 2019. These protests ended in riots, resulting in casualties and damage in several locations [2]. In this context, research utilizing technology to detect fake news is not only academically relevant but also practically significant. As the presidential and vice-presidential elections approach, tackling fake news is crucial to maintaining social stability and harmony.

This research is relevant both academically and practically, as it contributes to the literature on developing deep learning models for fake news classification. The findings of this study can also be directly applied in a practical context in Indonesia, where the spread of fake news presents a significant challenge. Surveys show that around 60% of over 1,000 respondents in Indonesia receive at least one fake news story every day. The government, including the president, actively urges the public to stop the spread of fake news on social media and to participate in combating it [3].

The scope of this research focuses on testing and analyzing the impact of various word embedding methods in detecting fake news using the Long Short-Term Memory (LSTM) algorithm. This study is limited by the dataset used and the scope of the LSTM model evaluated, specifically comparing embedding methods such as Word2Vec,

GloVe, and FastText. By understanding the role of technology in detecting fake news, this research aims to provide new insights into how deep learning can be leveraged to improve detection accuracy.

This study also builds upon and expands previous research that reviewed the use of LSTM for text classification. While there have been several studies on fake news classification, this research seeks to address certain shortcomings of prior work, particularly in comparing the performance of various embedding methods. By exploring these gaps, this study aims to enhance both the effectiveness and efficiency of more accurate fake news detection.

Previous research has evaluated the use of LSTM methods for text classification. The initial step in the study involved collecting tweet data, followed by text mining preprocessing. Lexicon-based features were then added, and a Long Short-Term Memory (LSTM) model was developed. The model was evaluated using a confusion matrix. Google Collaboratory was utilized as the primary tool in this research. Sentiment analysis was conducted on 858 processed data points, with an 80% training and 20% testing split. The results indicated that 52.2% of the public responded positively to the 2024 elections, 37% gave negative responses, and 10% remained neutral. Additionally, the training and validation accuracy graphs showed good results without overfitting or underfitting. The LSTM model achieved an accuracy of 78%. However, the study had some limitations, such as conducting only one test, focusing solely on Twitter as the data source, and lacking further exploration of the obtained data [4].

The research discusses a comparison of word embedding methods (WORD2VEC, GLOVE, and FASTTEXT) in text classification. The performance of each word embedding method is compared in classifying news from the 20 Newsgroups and Reuters Newswire datasets. Based on the experimental results, CNN demonstrated performance in classifying text using word embeddings such as word2vec, GloVe, and FastText. The F-Measure values for the 20 Newsgroups dataset are 0.925, 0.958, and 0.979, respectively. For the Reuters News dataset, the F-Measure values are 0.694, 0.688, and 0.715 [5].

Based on previous reviews and studies, this research aims to assess whether word embedding impacts the Long Short-Term Memory (LSTM) method for detecting fake news with the goal of improving accuracy. This effort is expected to make a significant contribution to the advancement of knowledge in the field of Deep Learning.

II. RESEARCH METHODS

To provide a clear overview of the research process, the steps to be undertaken are outlined as shown in Figure 1.

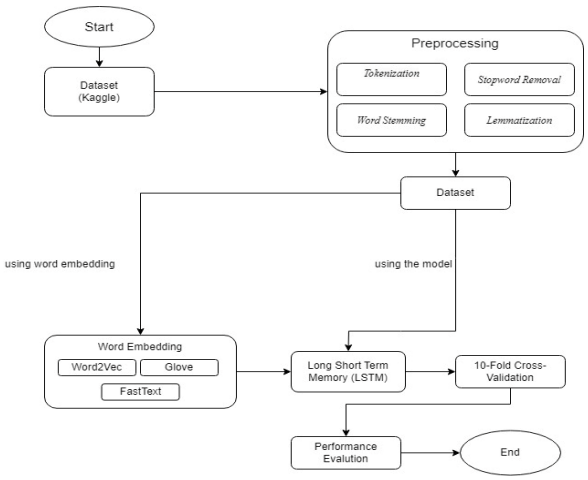


Figure 1 Research Workflow of Long Short-Term Memory (LSTM)

In the research process, the first step is data collection from Kaggle. The data will then undergo identification and correction or removal of any dirty or inaccurate data. Following this, the data will be preprocessed to convert raw data into a cleaner format suitable for analysis or modeling. After preprocessing, the dataset will be trained using two different approaches. In the first approach, the dataset will be directly trained with the Long Short-Term Memory (LSTM) algorithm model. In the second approach, the dataset will undergo feature extraction (Word Embedding) to identify and convert raw data into more informative and relevant features, and then trained using the Long Short-Term Memory (LSTM) model. Subsequently, the dataset will be divided into 10 parts for cross-validation. The testing results will include accuracy, precision, recall, and F1 score for each type of training data.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 4, December 2025, Pp. 3124-3136

A. *Text Classification*

Traditional text classification methods use machine learning. This approach requires a large amount of manually labeled training data and human involvement [6]. Text classification is used in various fields, including e-commerce platforms, blogs, content curation, directories, and documents and text classification in news agencies [7]. Over the past decade, the field has seen a significant increase in research, driven by the remarkable success of deep learning. Numerous methods, datasets, and evaluation metrics have been proposed in the literature, creating a need for a comprehensive and up-to-date survey[8]. The conventional framework for text classification involves several stages: preprocessing, feature extraction, feature selection, and classification. During the preprocessing stage, tasks typically performed include tokenization, stop-word removal, conversion of text to lowercase, and stemming.[9].

B. *Fake News Analysis*

The analysis of fake news detection is a process that involves examining and evaluating information to identify fake news or inaccurate information. The primary goal of this analysis is to separate true and verified information from false or intentionally misleading information. According to [10] , the spread of false information can have negative impacts on national security and individual safety. The detection of false information has become a common issue involving various applications in the field of cybersecurity. The spread of false information can have negative impacts on national security and individual safety. The detection of false information has become a common issue involving various applications in the field of cybersecurity. Therefore, there is a growing need for technology capable of providing insights into the accuracy of internet resources [11]. To mitigate these risks, various Natural Language Processing (NLP) methods have been designed and developed [12]. The aim of this research is to identify the most effective methods for detecting fake news, recognize the limitations of existing approaches, and provide recommendations to address these weaknesses [12].

C. *Dataset Collection*

Data collection from Kaggle involves a systematic and methodical process to ensure the integrity and quality of the data produced. Datasets available on Kaggle typically undergo curation by contributors who gather and clean the data before uploading it to the platform. This process includes identifying relevant data sources, such as news sites, social media, and other online platforms, and scraping the data using automated tools like Python or specialized APIs. Once the data is collected, it is cleaned to remove duplicate entries, correct errors, and categorize the data as needed. The cleaned and structured data is then uploaded to Kaggle with adequate documentation to help users understand the context and how to use the dataset. Through this process, users can access high-quality datasets for various analyses and further research. The dataset used in this study consists of 7,075 news articles classified as Valid and Hoax. This data includes a range of news sources and a broad time period, providing a comprehensive view of the patterns in the spread of fake news.

D. *Data Preprocessing*

Natural Language Processing (NLP) is a crucial technique that enables computers to mimic human brain capabilities. NLP operates within the domain of Artificial Intelligence (AI), with the goal of training computers to understand human language [13]. According to [14] human language is considered a medium for conveying information but often contains inaccuracies. This issue becomes particularly important in mathematics and, more broadly, in computer science. Therefore, in the context of using Natural Language Processing (NLP), a crucial initial step is to re-encode human language into a logical structure. This process must be completed before information can be processed for extraction, translation, speech-to-text conversion, or other related language processing objectives.

Labeling involves assigning values by identifying each piece of data. This process helps the model understand the data based on the predefined labels [15]. "Valid" indicates that the news has been thoroughly checked and supported by reliable sources, while "Hoax" signifies that the news cannot be trusted and may be false or misleading information. [14] Outlining several aspects of data processing in NLP, including:

a. Tokenization

Tokenization is the ability to divide a sentence into specific parts for further analysis; it is a fundamental process in NLP.

TABLE 1
TOKENIZATION

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 4, December 2025, Pp. 3124-3136

Data	Result
Juru bicara muda Partai Amanat Nasional (PAN), Dimas prakoso akbar, meminta ketua umum Partai Solidaritas Indonesia (PSI) tidak buru-buru memainkan peran sebagai korban atau playing victim terkait akun instagram, @giring, yang mendadak hilang.	['juru', 'bicara', 'muda', 'partai', 'amanat', 'nasional', 'pan', 'dimas', 'prakoso', 'akbar', 'meminta', 'ketua', 'umum', 'partai', 'solidaritas', 'indonesia', 'psi', 'tidak', 'buruburu', 'memainkan', 'peran', 'sebagai', 'korban', 'atau', 'playing', 'victim', 'terkait', 'akun', 'instagram', 'yang', 'mendadak', 'hilang']

b. Stopword Removal

Stopword Removal is a preprocessing step in text processing aimed at removing irrelevant words from a sentence based on a stopword list.

TABLE II
STOPWORD REMOVAL

Data	Result
Juru bicara muda Partai Amanat Nasional (PAN), Dimas prakoso akbar, meminta ketua umum Partai Solidaritas Indonesia (PSI) tidak buru-buru memainkan peran sebagai korban atau playing victim terkait akun instagram, @giring, yang mendadak hilang.	['juru', 'bicara', 'muda', 'partai', 'amanat', 'nasional', 'pan', 'dimas', 'prakoso', 'akbar', 'ketua', 'partai', 'solidaritas', 'indonesia', 'psi', 'buruburu', 'memainkan', 'peran', 'korban', 'playing', 'victim', 'terkait', 'akun', 'instagram', 'mendadak', 'hilang']

c. Word Stemming dan Lemmatization

After labeling, word stemming functions to group data with similar meanings. For example, words like "Perkembangan," "Kemajuan," "Progresif," and "berkembang" share the same root. Lemmatization is the process of transforming words into their "dictionary form," which is more recognizable.

E. Long Short-Term Memory Model

Long Short-Term Memory (LSTM) is a variant of the Recurrent Neural Network (RNN) algorithm that introduces memory cells to store information over extended periods [16]. RNNs face the vanishing gradient problem; LSTM addresses this issue by using memory cells and gating units (input gate, forget gate, output gate), allowing LSTM to read, store, and update information effectively.[17]. This algorithm has an architecture as shown in Figure 2.[18].

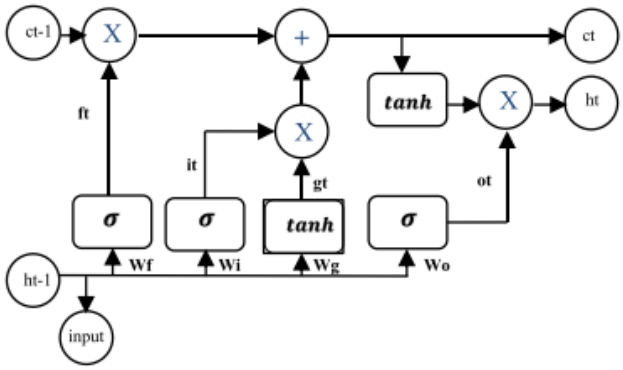


Figure 2 Algorithm LSTM Architecture

C_{t-1} represents the state of the neuron from the previous time step, while h_{t-1} is the output from that neuron. X_t is the current input, and the Sigmoid function works alongside it to determine the output h_t o of the current neuron. The gate is controlled by the Sigmoid function, which ensures the gate's output ranges between 0 and 1 [19]. When the forget gate's value is 0, it discards all previous information, and when the output gate's value is 0, it ignores newly computed states. The node's feature representation is initialized through word embeddings, and the feature matrix contain-ing these embeddings is denoted as $X \epsilon R^{nxm}$, where m represents the dimension of the feature vector. This matrix $X \epsilon R^{nxm}$ serves as input to the LSTM, which processes it to produce a feature matrix $H \epsilon R^{nxm}$ containing the sequence information of the text. [20].

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 4, December 2025, Pp. 3124-3136

F. *Word Embedding*

Word embedding is a technique used to represent words in documents, where each word in the vocabulary is transformed into a vector with numerical values. In this representation, words with similar meanings will have similar vectors [21]. Several methods for word embedding have been proposed by researchers, including:

1. *Word2Vec*

   Word2Vec is a word embedding method used to represent words as vectors. [22]. In Word2Vec, it is necessary to combine word vectors within a sentence using various techniques to create an overall representation of a paragraph. One method for combining word vectors is to stack the vectors of subsequent words beneath the vectors of previous words; this method results in a two-dimensional matrix for each paragraph.[23].

2. Glove

   GloVe is a model that creates meaningful word vector representations using a least-squares model based on the global frequency of word occurrences. Thus, the model efficiently leverages statistics for training[24].

3. FastText

   The algorithm developed by Facebook assumes that each word consists of n-gram characters, allowing for vector representation of words not present in the vocabulary. In this context, FastText embedding is used to generate 300-dimensional token vectors. Vectors associated with tweets are produced by averaging the token vectors [25]. FastText is indeed an extension of the existing Skip-gram model [26]. It is noted that the use of Softmax in the Skip-gram model has limitations because it can only predict one context. As a solution, the authors propose the FastText method, which is explained in the function below [27].

$$\sum_{t=1}^{T}\left[\sum_{c\in C_t} l\left(s(w_t,w_c)\right)+\sum_{n\in N_{t,c}} l\left(-s(w_t,n\ )\right)+\ \right] \quad (1)$$

Dimana:
s = *scoring function*
w = *weight*
ℓ = log(1 + e − x)
n = vocabulary size

G. *K-fold cross-validation*

K-fold cross-validation is a statistical method used to evaluate the performance of a developed model or algorithm. During the training phase, the dataset is divided into two parts: training data and validation data. The model is then trained on the training data and evaluated on the validation data across k iterations[28].

H. *Model Evaluation*

Classifiers can be evaluated using various scores depending on the classification goals, problem size, and desired accuracy level. To assess the effectiveness of the proposed method, metrics such as accuracy, precision, recall, and F-measure are used, which are commonly employed to measure classification performance [29]. Accuracy, Precision, and Recall are calculated using equations 2, 3, and 4. The F1-score, on the other hand, is the harmonic mean of Precision and Recall, expressed in equation 5 [30].

$$Accuracy = \frac{True\ Positive+True\ Negative}{True\ Positive+False\ Positive+False\ Negative+True\ Negative} \quad (2)$$

$$Precision = \frac{True\ Positive}{True\ Positive+False\ Positive} \quad (3)$$

$$Recall = \frac{True\ Positive}{True\ Positive+False\ Negative} \quad (4)$$

$$F1-score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (5)$$

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 4, December 2025, Pp. 3124-3136

III. RESULT AND DISCUSSION

This section of the paper presents the experimental results obtained through various trials and analysis of those results. All experiments were conducted on a laptop with an AMD Ryzen 5 processor and 8GB of RAM, running the Microsoft Windows 10 operating system. Colaboratory was used as the IDE for simulations. Python was employed as the programming language for implementation and generating simulation results.

1. Dataset

The data obtained from Kaggle consists of 7,075 news articles that have undergone a cleaning process, with 2,693 classified as valid and 4,382 as hoaxes. Submissions to the competition were assessed based on the accuracy score, as outlined in the official evaluation metrics [31]. The composition of this dataset is illustrated in Figure 3.

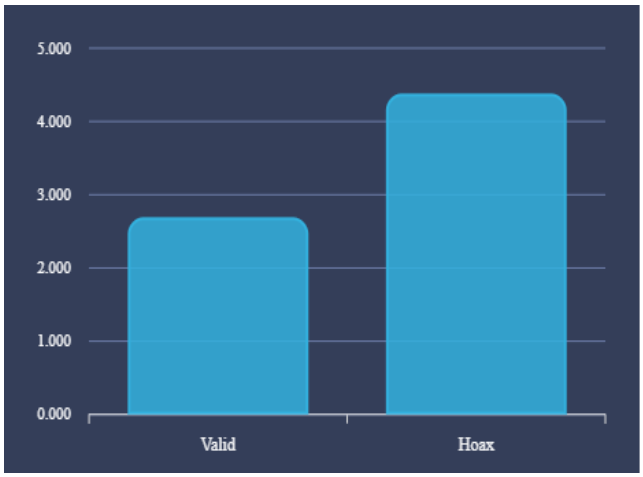


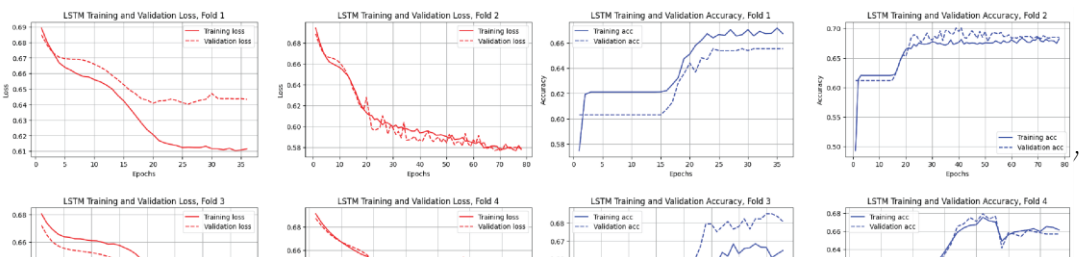
Figure 3 Dataset Composition.

2. Model Training and Parameters

In the author's experiments, they adapted two deep learning models for classification, namely LSTM and LSTM with Word Embedding. These classification models are considered a starting point due to their widespread use and relatively few trainable parameters. For common hyperparameters, the author used 10-fold cross-validation to evaluate and validate the machine learning model. In cross-validation, the dataset is divided into several parts or folds, where each fold is used alternately as the test data, while the remaining folds are used for training, with 100 epochs. Dropout with *p* 0.5 is applied across all layers of the downstream network to prevent overfitting. The performance of the candidate model can be assessed using various available evaluation metrics. To evaluate the accuracy in classifying news articles as Valid or Hoax, accuracy is used as the evaluation metric. Accuracy is measured as the ratio of the number of correctly predicted samples to the total number of samples in the given dataset.

3. Results of Long Short-Term Memory (LSTM)

The validation method used to test the results in this study is K-Fold Cross Validation. The K-Fold Cross Validation test on the LSTM algorithm was conducted with a k value of 10 folds. The validation results can be seen in Figure 4 below.



Implementation of Long Short-Term Memory (LSTM) Algorithm

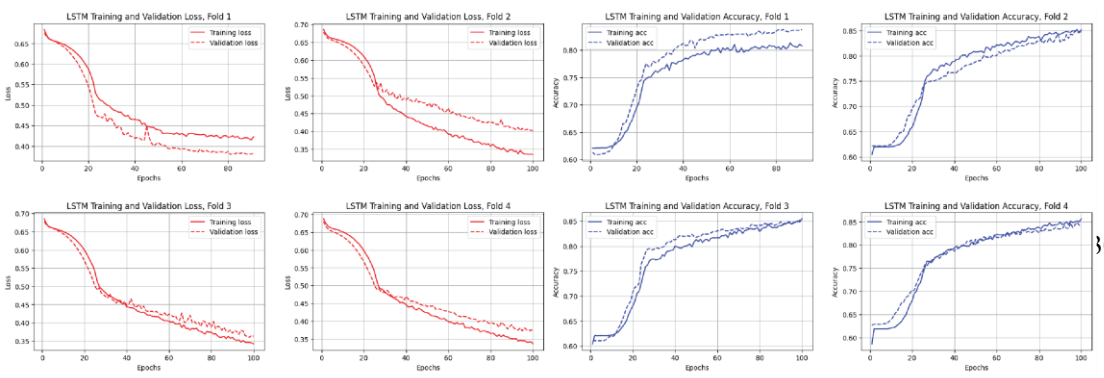
JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 4, December 2025, Pp. 3124-3136

Figure 4 LSTM Training and Validation Loss (red) and LSTM Training and Validation Accuracy (blue).

TABLE III
RESULTS OF LONG SHORT-TERM MEMORY

Performance Evalution	10-Fold Cross-Validation										Average scores
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	
Accuracy	0.65	0.68	0.68	0.67	0.66	0.67	0.69	0.67	0.67	0.68	0.67
Precision	0.64	0.69	0.68	0.66	0.67	0.68	0.68	0.68	0.70	0.69	0.68
Recall	0.92	0.86	0.90	0.93	0.91	0.85	0.91	0.89	0.84	0.84	0.89
F1 score	0.76	0.77	0.78	0.77	0.77	0.76	0.78	0.77	0.76	0.76	0.77

The consistent results across all folds indicate that the model has been well-trained and validated. However, the stagnation of accuracy around 0.67 suggests limitations in the model's ability to understand the data's characteristics, possibly due to feature extraction methods (such as text representation via embeddings or TF-IDF) or model architecture (the LSTM might not be fully optimized). The much higher recall compared to precision shows that the model is more focused on capturing all hoax examples but isn't always accurate in doing so. This likely means the model tends to make more positive predictions, aiming to avoid missing hoaxes, even if some of those predictions are incorrect. In the next stage, the author incorporated word embeddings (Word2Vec, GloVe, and FastText) to help enhance the model's performance, with the expectation of achieving higher accuracy compared to the initial tests. In this study, the LSTM model with Word2Vec word embedding was used, and K-Fold Cross Validation testing was conducted with a k value of 10 folds. The results are shown in Figure 5.



JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 4, December 2025, Pp. 3124-3136

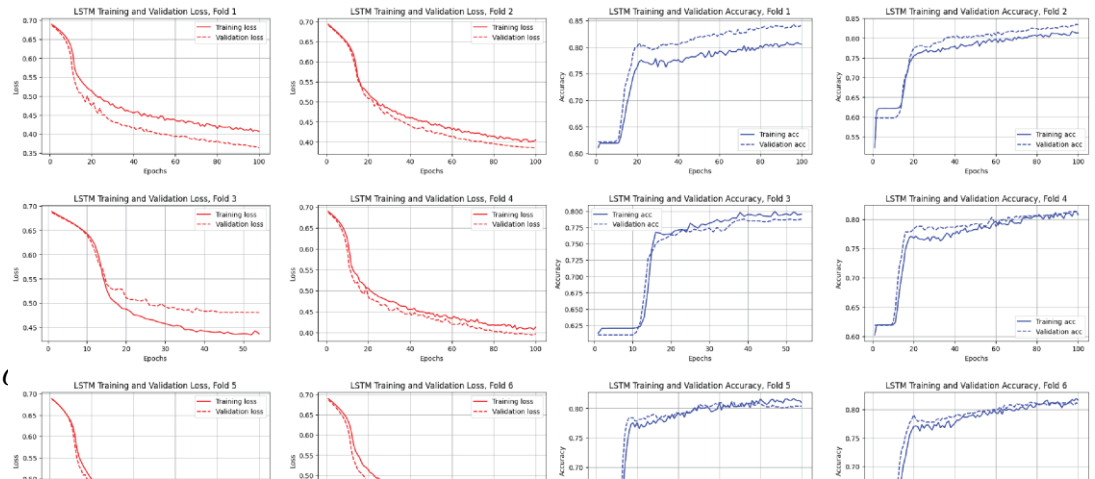
Figure 5 LSTM + Word2Vec Training and Validation Loss (red) and LSTM + Word2Vec Training and Validation Accuracy (blue).

TABLE IV
RESULTS OF LONG SHORT-TERM MEMORY WITH WORD2VEC

As shown in Table 4, The results show high and consistent performance across all 10 folds, with accuracy ranging from 0.81 to 0.87. This slight variation may be due to differences in data distribution or complexity between folds, which is typical in real-world datasets. High precision and recall (ranging from 0.83 to 0.90) indicate that the model effectively distinguishes between classes, maintaining a balance between false positives and false negatives. The consistent F1 score (average of 0.87) confirms that the model is stable and performs well in classification tasks without significant trade-offs between precision and recall. The next step will involve applying the same approach

Performance Evalution	**10-Fold Cross-Validation**										**Average scores**
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	
Accuracy	0.83	0.84	0.84	0.84	0.87	0.86	0.81	0.83	0.81	0.86	0.84
Precision	0.84	0.87	0.85	0.88	0.89	0.89	0.84	0.87	0.83	0.90	0.87
Recall	0.89	0.87	0.90	0.87	0.90	0.89	0.85	0.85	0.84	0.88	0.87
F1 score	0.86	0.87	0.87	0.87	0.90	0.89	0.84	0.86	0.84	0.89	0.87

to Word embedding (GloVe) using the K-Fold Cross Validation method on the LSTM algorithm. In this study, the LSTM model with GloVe word embedding was used, and K-Fold Cross Validation testing was conducted with a k value of 10 folds. The results are shown in Figure 6.



Implementation (
Algorithm

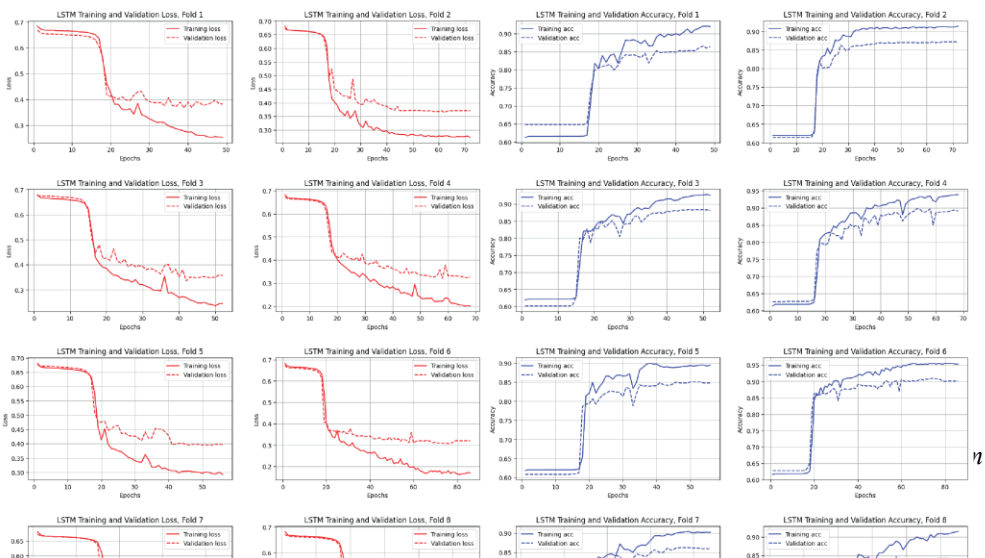
JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 4, December 2025, Pp. 3124-3136

Figure 6 LSTM + GloVe Training and Validation Loss (red) and LSTM + GloVe Training and Validation Accuracy (blue).

TABLE IV
RESULTS OF LONG SHORT-TERM MEMORY WITH GLOVE

As seen in Table 5, The performance evaluation using 10-fold cross-validation shows fairly consistent results, with an average accuracy of 0.81, precision of 0.86, recall of 0.82, and an F1 score of 0.84. Although accuracy varies between 0.78 and 0.84, precision remains high and consistent, indicating the model's ability to avoid false positives. However, recall is more variable, suggesting the model does not always capture all positive instances in each fold. Overall, the model demonstrates a good balance between precision and recall, despite slight variations across folds due to differences in data distribution. In this study, the LSTM model with FastText word embedding was used, and K-Fold Cross Validation testing was conducted with a k value of 10 folds. The results are shown in Figure 7.

Performance Evalution	**10-Fold Cross-Validation**										**Average scores**
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	
Accuracy	0.84	0.83	0.78	0.81	0.80	0.81	0.84	0.78	0.82	0.81	0.81
Precision	0.88	0.86	0.82	0.88	0.85	0.87	0.88	0.85	0.87	0.88	0.86
Recall	0.85	0.85	0.82	0.80	0.82	0.81	0.85	0.79	0.83	0.81	0.82
F1 score	0.87	0.86	0.82	0.84	0.83	0.84	0.87	0.82	0.85	0.84	0.84



JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 4, December 2025, Pp. 3124-3136

Figure 7 LSTM + FastText Training and Validation Loss (red) and LSTM + FastText e Training and Validation Accuracy (blue).

TABLE VI
RESULTS OF LONG SHORT-TERM MEMORY WITH FASTTEXT.

Table 6 demonstrates that using word embedding (FastText) with the K-Fold Cross Validation method on the LSTM algorithm led to an increase in average accuracy to 0.86. The results of the 10-fold cross-validation show consistent model performance across different folds, with an average accuracy of around 0.86. Precision is high at 0.93, indicating that the model effectively reduces false positives. However, the average recall is slightly lower at 0.84, meaning the model occasionally misses true positive instances. The average F1 score of 0.88 reflects a good balance between precision and recall, suggesting that the model handles both false positives and false negatives fairly well. While most folds show strong performance, Fold 9 has the lowest scores, particularly in recall, which slightly pulls down the overall average.

Performance Evalution	**10-Fold Cross-Validation**										**Average scores**
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	
Accuracy	0.85	0.87	0.88	0.89	0.85	0.90	0.86	0.83	0.80	0.84	0.86
Precision	0.94	0.91	0.93	0.93	0.91	0.92	0.92	0.93	0.95	0.92	0.93
Recall	0.81	0.86	0.85	0.90	0.82	0.93	0.84	0.79	0.69	0.82	0.83
F1 score	0.87	0.89	0.89	0.91	0.87	0.92	0.88	0.85	0.80	0.87	0.88

4. Algorithm Comparison Evaluation

TABLE 7
RESULTS OF AVERAGE PERFORMANCE EVALUATION

Average Performance Evaluation	LSTM	LSTM+ Word2Vec	LSTM+Glove	LSTM+FastText
Accuracy	0.67	0.84	0.81	0.86
Precision	0.68	0.87	0.86	0.93
Recall	0.89	0.87	0.82	0.83
F1 score	0.77	0.87	0.84	0.88

In the evaluation results presented in Table 7, there are notable differences in performance metrics between the standard LSTM and LSTM with word embeddings such as Word2Vec, GloVe, and FastText. In terms of accuracy, the use of word embeddings showed a clear improvement, with LSTM+FastText achieving the highest accuracy of 0.86, compared to the standard LSTM, which only reached 0.67. Precision also improved significantly, especially

for LSTM+FastText, which reached 0.93, indicating that this model was more accurate in identifying hoax news compared to the other models. However, the recall for LSTM+FastText was slightly lower (0.83) compared to the standard LSTM (0.89). This indicates that while the standard LSTM was better at identifying all hoax news overall, it might have produced more false positives. The F1 score, which balances precision and recall, also showed an improvement with the use of embeddings, with LSTM+FastText achieving a score of 0.88, the highest among the models. The differences in precision and recall between the standard LSTM and LSTM with word embeddings, particularly for FastText and Word2Vec, highlight the need for deeper analysis. LSTM+FastText has a higher precision, meaning it made fewer false classifications of hoax news, but its recall was slightly lower, suggesting that the model was more selective in deciding whether a piece of news was hoax but did not capture all hoax cases. LSTM+Word2Vec, with its more balanced precision and recall, may be more effective if the goal is to detect as many hoaxes as possible without too many false positives.

A more in-depth discussion on the reasons behind the performance differences of embedding methods would enrich the analysis, especially regarding data distribution and the text patterns captured by these embeddings. However, the article has yet to address significant limitations that may affect model performance, such as the relatively small dataset size (7,075 articles, consisting of 2,693 valid news and 4,382 hoax news) and the imbalance in data composition. A limited dataset may restrict the model's ability to identify more complex patterns, while data imbalance could lead to bias toward the majority class (hoax), resulting in higher recall for hoax news but suboptimal precision or recall for valid news. Furthermore, with a small dataset, deep learning models like LSTM may potentially overfit the training data, making it difficult to classify unseen test data. This imbalance could also impact overall accuracy, where the model may appear "good" by frequently classifying news as hoax, but it is less effective in detecting valid news. Including a discussion on these limitations and possible solutions, such as oversampling, undersampling, class balancing using methods like SMOTE, or incorporating additional data, would enhance the credibility of the research findings and demonstrate awareness of the factors that could influence model performance.

By comparing the results of this study with previous research, the authors can provide a stronger context for the performance of the model used and demonstrate how this study improves or expands upon existing approaches. research by [4] using the LSTM algorithm, a previous study achieved an accuracy of 78%, whereas the proposed model in this research, which incorporates LSTM with word embedding, successfully improved the average accuracy to 86%. This improvement highlights that the use of word embeddings, such as FastText, plays a significant role in enhancing the model's ability to more accurately detect fake news, improving upon the performance of previous approaches. This comparison also underscores that the proposed method not only refines existing techniques but also delivers superior results in the context of news classification.

IV CONCLUSION

The evaluation of different LSTM models with and without word embeddings reveals notable improvements in performance metrics when word embeddings are applied. Initially, the standard LSTM algorithm achieved a baseline accuracy of 67%. The introduction of Word2Vec, GloVe, and FastText embeddings resulted in significant performance enhancements. Specifically, LSTM with Word2Vec increased the accuracy to 84%, while LSTM with GloVe reached 81%. The most substantial improvement was observed with LSTM using FastText, which achieved an accuracy of 86%. Precision, Recall, and F1-Score metrics further highlight the benefits of incorporating word embeddings. LSTM with FastText demonstrated the highest Precision (93%) and F1-Score (88%), although its Recall (83%) was slightly lower compared to the baseline LSTM (89%). Despite the variations in Recall, the overall F1-Score and Precision improvements with FastText indicate a better balance between avoiding false positives and correctly identifying true positives. The consistent high performance across different metrics with LSTM and FastText validates the effectiveness of using advanced word embeddings. The use of K-Fold Cross Validation across tests ensured that the model's performance was robust and not subject to overfitting or underfitting. In conclusion, integrating word embeddings, especially FastText, significantly enhances the LSTM algorithm's ability to classify hoax news, achieving superior accuracy and balanced performance metrics. The study underscores the importance of embedding techniques in improving model precision, recall, and overall classification efficacy.

REFERENCES

[1] M. R. Ramadhani and A. R. Pratama, "Analisis Kesadaran Cybersecurity Pada Pengguna Media Sosial Di Indonesia," *Journal.Uii.Ac.Id*, vol. 1, no. 2, pp. 1–8, 2020, [Online]. Available: https://journal.uii.ac.id/AUTOMATA/article/download/15426/10219

[2] Suliyansyah, "Menepis hoax media sosial di tahun politik: pendekatan systematic literature review," *J. Adhyasta Pemilu*, vol. 6, no. 1, pp. 1–14,

2023.
[3] A. Rusli, J. C. Young, and N. M. S. Iswari, "Identifying fake news in indonesian via supervised binary text classification," *Proc. - 2020 IEEE Int. Conf. Ind. 4.0, Artif. Intell. Commun. Technol. IAICT 2020*, pp. 86–90, 2020, doi: 10.1109/IAICT50021.2020.9172020.
[4] D. A. Firdlous, R. Andrian, and S. Widodo, "Sentiment Analysis Public Twitter on 2024 Election using the Long Short Term Memory Model," *Sistemasi*, vol. 12, no. 1, p. 52, 2023, doi: 10.32520/stmsi.v12i1.2145.
[5] A. Nurdin, B. Anggo Seno Aji, A. Bustamin, and Z. Abidin, "Perbandingan Kinerja Word Embedding Word2Vec, Glove, Dan Fasttext Pada Klasifikasi Teks," *J. Tekno Kompak*, vol. 14, no. 2, p. 74, 2020, doi: 10.33365/jtk.v14i2.732.
[6] C. Li, G. Zhan, and Z. Li, "News Text Classification Based on Improved Bi-LSTM-CNN," *Proc. - 9th Int. Conf. Inf. Technol. Med. Educ. ITME 2018*, pp. 890–893, 2018, doi: 10.1109/ITME.2018.00199.
[7] M. Zhang, "Applications of Deep Learning in News Text Classification," *Sci. Program.*, vol. 2021, 2021, doi: 10.1155/2021/6095354.
[8] Q. Li *et al.*, "A Survey on Text Classification: From Traditional to Deep Learning," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 2, 2022, doi: 10.1145/3495162.
[9] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," *Inf. Process. Manag.*, vol. 50, no. 1, pp. 104–112, 2014, doi: 10.1016/j.ipm.2013.08.006.
[10] V. V. Hirlekar and A. Kumar, "Natural language processing based online fake news detection challenges - A detailed review," *Proc. 5th Int. Conf. Commun. Electron. Syst. ICCES 2020*, no. Icces, pp. 748–754, 2020, doi: 10.1109/ICCES48766.2020.09137915.
[11] K. R. Sabarmathi, K. Gowthami, and S. Sanjay Kumar, "Fake news detection using machine learning and Natural Language Inference (NLI)," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1084, no. 1, p. 012018, 2021, doi: 10.1088/1757-899x/1084/1/012018.
[12] N. Hoy and T. Koulouri, "A Systematic Review on the Detection of Fake News Articles," no. Ml, 2021, [Online]. Available: http://arxiv.org/abs/2110.11240
[13] M. Vinodkumar Sadhuram and A. Soni, "Natural Language Processing based New Approach to Design Factoid Question Answering System," *Proc. 2nd Int. Conf. Inven. Res. Comput. Appl. ICIRCA 2020*, pp. 276–281, 2020, doi: 10.1109/ICIRCA48905.2020.9182972.
[14] P. H. Chen, "Essential Elements of Natural Language Processing: What the Radiologist Should Know," *Acad. Radiol.*, vol. 27, no. 1, pp. 6–12, 2020, doi: 10.1016/j.acra.2019.08.010.
[15] Q. Bi, K. E. Goodman, J. Kaminsky, and J. Lessler, "Running head : Machine Learning for Epidemiologists SC," *Am. J. Epidemiol.*, vol. 21231, 2019.
[16] I. Cholissodin, "Social Computing to Create Government Public Policy Document Blueprint Draft Based on Social Media Data About Covid-19 Using LSTM and MMR Hybrid Algorithms," *Proc. Int. Conf. Green Technol.*, vol. 11, no. 1, p. 6, 2021, doi: 10.18860/icgt.v11i1.1394.
[17] M. A. Riza and N. Charibaldi, "Emotion Detection in Twitter Social Media Using Long Short-Term Memory (LSTM) and Fast Text," *Int. J. Artif. Intell. Robot.*, vol. 3, no. 1, pp. 15–26, 2021, doi: 10.25139/ijair.v3i1.3827.
[18] A. Hasiholan, I. Cholissodin, and N. Yudistira, "Analisis Sentimen Tweet Covid-19 Varian Omicron pada Platform Media Sosial Twitter menggunakan Metode LSTM berbasis Multi Fungsi Aktivasi dan GLOVE," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 10, pp. 4653–4661, 2022, [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/11648
[19] P. Liu, X. Qiu, and H. Xuanjing, "Recurrent neural network for text classification with multi-task learning," *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2016-Janua, pp. 2873–2879, 2016.
[20] H. Wang and F. Li, "A text classification method based on LSTM and graph attention network," *Conn. Sci.*, vol. 34, no. 1, pp. 2466–2480, 2022, doi: 10.1080/09540091.2022.2128047.
[21] H. A. Almuzaini and A. M. Azmi, "Impact of Stemming and Word Embedding on Deep Learning-Based Arabic Text Categorization," *IEEE Access*, vol. 8, pp. 127913–127928, 2020, doi: 10.1109/ACCESS.2020.3009217.
[22] M. A. Nurrohmat and A. SN, "Sentiment Analysis of Novel Review Using Long Short-Term Memory Method," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 13, no. 3, p. 209, 2019, doi: 10.22146/ijccs.41236.
[23] L. Singh, "Fake News Detection: a comparison between available Deep Learning techniques in vector space," *2020 IEEE 4th Conf. Inf. Commun. Technol.*, pp. 5–8, 2020.
[24] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.
[25] N. Badri, F. Kboubi, and A. H. Chaibi, "Combining FastText and Glove Word Embedding for Offensive and Hate speech Text Detection," *Procedia Comput. Sci.*, vol. 207, no. Kes, pp. 769–778, 2022, doi: 10.1016/j.procs.2022.09.132.
[26] T. T. Mengistie and D. Kumar, "Deep Learning Based Sentiment Analysis On COVID-19 Public Reviews," in *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2021, pp. 444–449.
[27] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguist.*, vol. 5, pp. 135–146, 2017.
[28] Y. N. FUADAH, I. D. UBAIDULLAH, N. IBRAHIM, F. F. TALININGSING, N. K. SY, and M. A. PRAMUDITHO, "Optimasi Convolutional Neural Network dan K-Fold Cross Validation pada Sistem Klasifikasi Glaukoma," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 10, no. 3, p. 728, 2022, doi: 10.26760/elkomika.v10i3.728.
[29] E. Gundogan and M. Kaya, "Research paper classification based on Word2vec and community discovery," *2020 Int. Conf. Decis. Aid Sci. Appl. DASA 2020*, pp. 1032–1036, 2020, doi: 10.1109/DASA51403.2020.9317101.
[30] M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi, and B. W. On, "Fake news stance detection using deep learning architecture (CNN-LSTM)," *IEEE Access*, vol. 8, pp. 156695–156706, 2020, doi: 10.1109/ACCESS.2020.3019735.
[31] L. Triyono, R. Gernowo, Prayitno, M. Rahaman, and T. R. Yudantoro, "Indonesian Fake News Detection Using Various Machine Learning Technique," *Int. J. Informatics Vis.*, vol. 7, no. 3, pp. 726–732, 2023, doi: 10.30630/joiv.7.3.1243.