Scientific Journal of Informatics
Vol. 11, No. 3, Aug 2024

p-ISSN 2407-7658 https://journal.unnes.ac.id/journals/sji/index e-ISSN 2460-0040

Feature Expansion with GloVe and Particle Swarm Optimization for Detecting the Credibility of Information on Social Media X with Long Short-Term Memory (LSTM)

Famardi Putra Muhammad Raffly[1*], Erwin Budi Setiawan[2]
[1,2]Department of Informatics, Faculty of Informatics, Telkom University, Indonesia

Abstract.
Purpose: This research aims to develop a system for detecting the credibility of information on social media X by classifying tweets as credible or non-credible. Additionally, it seeks to improve the accuracy of classification and prediction of information credibility using feature extraction methods, semantic features, feature expansion, and optimization.
Methods: The system is built using a deep learning approach with Long Short-Term Memory (LSTM), Term Frequency-Inverse Document Frequency (TF-IDF), Robustly optimized BERT Approach (RoBERTa), Global Vector (GloVe), and Particle Swarm Optimization (PSO). The dataset consists of 54,766 Indonesian tweets from social media X, focusing on the 2024 General Election and using several keywords such as 'Pemilu 2024', 'Pilpres 2024', 'anies baswedan', 'Prabowo', '#GanjarPranowo', and '#debatCapres'.
Result: The results of this study show that the highest accuracy achieved is 89.09% using LSTM with an 80:20 data split, baseline unigram, RoBERTa, Top1 corpus IndoNews, and PSO of the LSTM model's hyperparameters, resulting in a highly significant statistical improvement of 0.96% over the baseline model.
Novelty: This research contributes to information credibility classification research using RoBERTa to add semantic features and GloVe to expand features by utilizing a built corpus and finding similar words to connect with these expanded features. Additionally, PSO is applied to find the optimal hyperparameters, thereby improving the performance and accuracy of the LSTM classification model.

Keywords: Information credibility, Social media X, GloVe, LSTM, PSO
Received June 2024 / **Revised** July 2024 / **Accepted** July 2024

INTRODUCTION

Currently, online social networks play a crucial role in managing information by facilitating interaction and information sharing [1]. This has made social networks a recognized source of information [2]. However, the risk of inaccurate information is a significant drawback that must be addressed [3], especially on the X platform, which allow the spread of erroneous information due to inadequate content oversight [4], [5], [6]. As a social media and information source, the X platform enables users to easily access information through tweets, including the latest news, trends, and opinions on current global topics. However, it is important to note that not all information on the X platform can be considered accurate. The credibility of information on this platform is a primary concern in current research, for example after the 2010 earthquake in Chile, when many rumors and unofficial information on the X platform caused anxiety and insecurity among the local population [7].

A study [8] found that users' assessments of information accuracy on the X platform tend to be low. Several studies have been carried out to classify the credibility of information on social media. One of the first studies was done in 2011 by Castillo et al., using machine learning approaches such as Support Vector Machine (SVM), Decision Tree, Bayes Networks and Decision Rules. The focus was to classify the trustworthiness of information on social media X as either credible or not credible. The best results were obtained using the J48 Decision Tree algorithm, achieving an accuracy of 86% [7]. In 2020, Erwin et al. performed similar research to Castillo et al., but they enhanced the analysis by adding combined features

[*]Corresponding author.
Email addresses: muhammadraffly@student.telkomuniversity.ac.id (Raffly)[*], erwinbudisetiawan@telkomuniversity.ac.id (Setiawan)
DOI: 10.15294/sji.v11i3.7839

from User Profile and Message Content. This study used a dataset from social media X, involving 115 accounts and 19,401 tweets in Indonesian. By incorporating combined features from User Profile and Message Content, Erwin et al. achieved the best results using the J48 Decision Tree algorithm, with an accuracy of 88.42% [9]. The following year, Marina et al. compared algorithms such as SVM, Naïve Bayes, KNN, Random Forest and Logistic Regression to detect information credibility on the X platform. This study used two feature extraction approaches: Content-Based and User-Based. According to the results, the highest accuracy of 83.4% was attained by the Random Forest algorithm [2].

In the next iteration, more studies used machine learning approaches to classify information credibility on social media, particularly the X platform. In 2020, Vyas et al. implemented a deep learning approach using Long Short-Term Memory (LSTM) to assess the performance of the LSTM model in differentiating between trustworthy and untrustworthy news on the X platform. The LSTM approach was found helpful in understanding the continuous representation of microblog events to assess the credibility of information in tweets, classifying them as credible or not. This study used GloVe (Global Vectors) for feature extraction to create word vector representations that include tagging like URLs, hashtags (#), and the "@" symbol. These word vector representations were then used as features in the LSTM model to assess news credibility on the X platform. By using GloVe, this study achieved more accurate word vector representations for tagging features, improving the LSTM model's performance in evaluating news credibility on the X platform. The results showed an accuracy of 81% [10]. In 2023, Fadhli et al. implemented a hybrid deep learning model to detect conversation credibility on the X platform. The employed method was known as CreCDA, a combination of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models by incorporating post and user features to improve the performance of credibility detection. This approach is aimed to recognize credibility levels in online social interactions efficiently. Additionally, this study used Word2Vec for feature extraction, a neural network model used to learn word vector representations in documents. The accuracy of the hybrid deep learning model reached 81% [11].

In a sentiment analysis study on social media X, Diaz et al. found a 1.1% increase in accuracy by adding semantic features using the Robustly optimized BERT Approach (RoBERTa) [12]. Another study by Anindika et al. focused on detecting hoaxes on social media X, using TF-IDF for feature extraction and GloVe for feature expansion. GloVe was used to find similar words and link them to existing expanded features using a built-in corpus. The results showed that the baseline (TF-IDF N-gram + BERT) combined with the GloVe-expanded corpus achieved a higher accuracy of 98.57%, with a 4.69% improvement over the previous baseline [13]. Previous studies have shown that using methods like TF-IDF, RoBERTa, and GloVe results in better accuracy, even though deep learning approaches are rarely applied. In a study [10], LSTM was used for its ability to evaluate the credibility of information in tweets, classifying them as credible or not. LSTM's advantage lies in its memory cells, outperforming conventional recurrent neural networks [14]. Additionally, because optimization methods have not been applied in research on information credibility classification, optimization methods like Particle Swarm Optimization (PSO) can be used, as they are known to achieve optimal solutions more quickly and easy to implement [15][16]. PSO can enhance the model's accuracy and performance by finding optimal hyperparameters. This was demonstrated in a study by Regina et al., who applied CNN-PSO for sentiment analysis on the social media X. PSO was used to find the optimal hyperparameters for the CNN model, resulting in a 10.07% increase in accuracy [17].

Based on the issues outlined, this research aims to detect the credibility of information on social media X by classifying tweets as credible or non-credible. The approach combines methods proven effective in previous studies to achieve optimal accuracy. The combination includes LSTM as the classification model, TF-IDF as the feature extraction and a baseline, RoBERTa for semantic features, GloVe for feature expansion, and PSO as the optimization method.

METHODS

Design system

The Information Credibility Detection System developed in this research is presented in Figure 1, with five experimental scenarios that can be observed in Table 1.

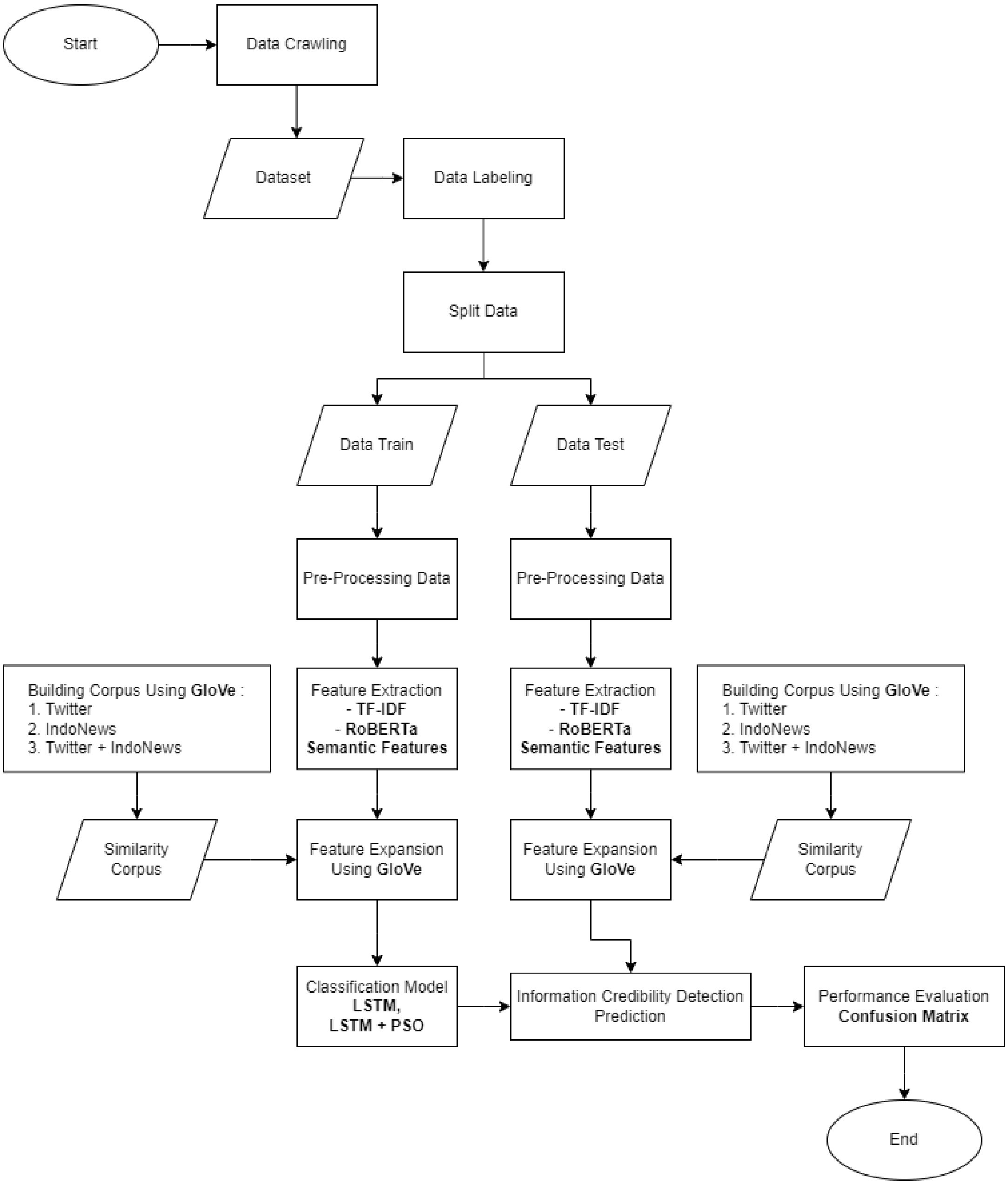

Figure 1. Information credibility detection flowchart

Table 1. Experimental scenarios description

Scenario	Description
1	Conducting data split testing as the baseline, maximum TF-IDF feature testing, and TF-IDF parameter testing. The results with the best accuracy are used for the subsequent scenario.
2	Conducting baseline combination testing (Unigram, Bigram, Trigram, and the combinations). The results with the best accuracy are used for the following scenario.
3	Adding RoBERTa semantic features and compare them with the previous baseline. The results with the best accuracy are used for the next scenario.
4	Implementing feature expansion with Global Vectors (GloVe). The results with the best accuracy are used for the following scenario.
5	Implementing the Particle Swarm Optimization (PSO) method to find optimal hyperparameters (learning rate, dropout rate, units, and layers) for the LSTM model.

Data crawling

Data was collected from social media X using the Tweet-Harvest tool, which is available in Python. A total of 54,766 tweets in Indonesian related to the 2024 general election topic were gathered using six keywords: 'Pemilu 2024', 'Pilpres 2024', 'anies baswedan', 'Prabowo', '#GanjarPranowo', and '#debatCapres'. Next, the data was processed in the preprocessing stage. Table 2 represents the number of data for each keyword used in the data crawling.

Table 2. Number of data by keywords

Keyword	Data Amount
Pemilu 2024	11,672
Pilpres 2024	2,688
anies baswedan	13,557
Prabowo	16,282
#GanjarPranowo	9,345
#debatCapres	1,222
Total	54,766

Data labeling

In this study, the data labeling process was performed manually by three analysts. The final label was determined by a majority vote. The three analysts agreed and shared the same perception of the criteria for tweets considered credible or non-credible. A tweet was considered credible if it was informative, factual, and not a personal opinion. Additionally, the account user posting the tweet had to be competent on the relevant topic. The detailed criteria for credible tweets can be seen in Table 3.

Table 3. Criteria for credible tweets

Criteria	**Description**	**Ref**
Informative and Factual	The tweet contains informative content based on factual events	Castillo *et al.* [7]
Not a Personal Opinion	The tweet contains content that is not a personal opinion	Castillo *et al.* [7]
Competent Account	An account that is competent in discussing the topic of the post	Erwin *et al.* [9]

Data tweets labeled as credible are assigned a value of 1, while those labeled as non-credible are assigned a value of 0. The data of tweets labeled as credible and non-credible are presented in Table 4. This study used Fleiss' Kappa to measure data labeling consistency and agreement among the three individuals. According to Fleiss, Kappa values are categorized into three levels: poor agreement ($k < 0.40$), good agreement ($0.40 \leq k < 0.75$), and excellent agreement ($k \geq 0.75$) [18]. This study achieved a Kappa value of 0.89, indicating that the data labeling by the three individuals had excellent agreement and consistency.

Table 4. Credible and non-credible data labels

Label	Number of Data
Credible	27,277
Non-Credible	27,489

The data balance also needs attention because it can affect the results [13]. The balanced data has almost equal class proportions; so, no class significantly dominates over the others. Conversely, imbalanced number of data has uneven class distributions, with one class having significantly fewer or more instances than the others [19]. In this study, the number of data used was set to be balanced between credible and non-credible categories. Therefore, data balancing methods was no longer required. Figure 2 illustrates the data balance between the credible and non-credible categories.

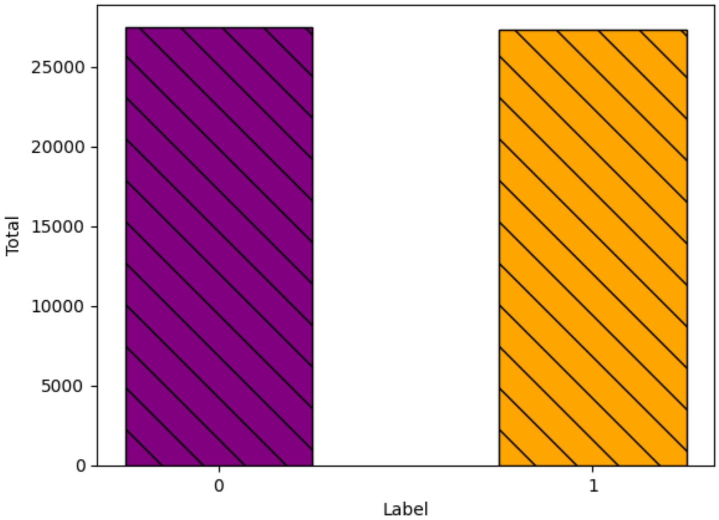

Figure 2. The balance of credible and non-credible data categories

Preprocessing

Preprocessing data was performed to address errors, tidy up word forms, and reduce the number of words in the data before it was fed into the classification process. An example of data preprocessing is presented in Table 5. The data preprocessing steps in this study include:

1) Cleaning: In the cleaning stage, unnecessary characters such as punctuation marks, hashtags, URLs, and numbers in the dataset were removed or cleaned.
2) Case Folding: In this stage, all words in the dataset were converted from uppercase to lowercase.
3) Stopword Removal: In the stopword removal stage, irrelevant words such as pronouns, prepositions, and conjunctions that do not have specific meanings were eliminated. This stage used the Natural Language Toolkit (NLTK) library with a list of Indonesian stopwords.
4) Stemming: In this stage, words were transformed into their root forms by removing affixes from the words. The algorithm applied in the stemming stage was the Sastrawi stemming algorithm.
5) Tokenization: In the tokenization stage, sentences were broken down into a sequence of words.

Table 5. Data preprocessing example

Preprocessing	Text
Original Tweet	“Waspada Anies Baswedan didukung oleh bibit2 teroris https://t.co/18BBoy78ae”
Cleaning	“Waspada Anies Baswedan didukung oleh bibit teroris”
Case Folding	“waspada anies baswedan didukung oleh bibit teroris”
Stopword Removal	“waspada anies baswedan didukung bibit teroris”
Stemming	“waspada anies baswedan dukung bibit teroris”
Tokenization	[‘waspada’, ‘anies’, ‘baswedan’, ‘dukung’, ‘bibit’, ‘teroris’]

N-Gram

In this study, N-Gram was used for feature extraction, including Unigram, Bigram, and Trigram for tweet representation. The aim of using the N-Gram model for feature extraction was to understand the sequence of words so that the system can better comprehend the context of the data [20]. N-Gram can isolate desired words, such as predicting the correctness of limited word spellings. In this context, N-Gram referred to a set of consecutive words of a specified length, presented as N-word sequences [13].

Term frequency-inverse document frequency (TF-IDF)

A widely used technique in feature extraction is TF-IDF. Feature extraction consists of choosing and extracting important details from data to create feature vectors. TF-IDF assesses the importance of a word in a document corpus. This process reflects how significant a term is within the context of a document [21]. TF-IDF plays a crucial role in machine learning and text mining to weight features. The weight increases with the word’s frequency in the document but is balanced by the word's frequency across the entire dataset. This mechanism eliminates common words from consideration while assigning higher weights to rare words. TF-IDF achieves high weight when a word frequently appears in a specific document but rarely across the entire dataset [22]. The formula for TF-IDF weighting used in this study is as follows:

$$W_{ij} = tf_{ij} \times IDF_j,\ IDF_j = \left(\log\frac{N}{df}\right) \tag{1}$$

W_{ij} represents the document weight for word-j, tf_{ij} represents the frequency of the word in the document, IDF_j represents the Inverse Document Frequency, N represents the total number of documents and df represents the number of documents containing the word.

Robustly optimized BERT approach (RoBERTa)

The Robustly optimized BERT Approach (RoBERTa) is a word embedding method that represents each word in the text as a dense numerical vector. This method is a modification and improvement of the previous approach known as Bidirectional Encoder Representations from Transformers (BERT) [23]. By implementing dynamic masking patterns and utilizing larger data batches, RoBERTa strengthens the model by differentiating each epoch using different masking patterns, thus improving speed and model performance. To significantly enhance the model's performance, the Next Sentence Prediction (NSP) feature is removed in RoBERTa [24]. This study uses RoBERTa as a semantic feature to represent words as vectors combined with the baseline.

Global vector (GloVe)

Feature expansion aims to assess the possibility of replacing words which do not present in the tweet representation with words that have semantic relationships [25]. One feature expansion method that can be used is Global Vector (GloVe). GloVe is an unsupervised learning approach that excels in creating word representations, particularly in terms of word similarity and name identification [26]. This method was developed through research at Stanford University. The structure of GloVe leverages the main advantage of performing data computations simultaneously to capture meaningful linear substructures, similar to the approach used by Word2Vec [13]. In this study, a corpus was built using GloVe from tweets, news articles, and a combination of tweets and news articles to generate hundreds of vectors for each word in the dictionary. The news includes 95,664 articles on the 2024 General Election, sourced from Indonesian media outlets such as Detik, CNBC Indonesia, Antara, MetroTV, and Tirto. The news articles were collected using five keywords: 'pemilu 2024', 'prabowo', 'anies', 'ganjar', and 'pilpres'. The corpus built using GloVe can be observed in Table 6.

Table 6. GloVe corpus

Corpus	Number of Vocabulary
Tweet	40,465
IndoNews	131,580
Tweet + IndoNews	150,943

The GloVe corpus also generates word similarities. For example, Table 7 represents words are synonymous to "damai" in the similarity corpus built from tweets. Columns 'Rank 1' to 'Rank 10' and their values reflect how closely each word is related to "damai".

Table 7. Word similarity of "damai"

Word	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
damai	sejuk	aman	wujud	tentram	mari
	0.8857	0.8109	0.7944	0.7466	0.7385
	Rank 6	Rank 7	Rank 8	Rank 9	Rank 10
	sintong	cipta	uii	nyaman	sukses
	0.7335	0.7187	0.7158	0.7088	0.7025

The word similarity corpus is used to expand features in the vector representation obtained from TF-IDF extraction. The feature expansion process involves replacing words with zero values in the feature vector with similar words from the GloVe list that appear in the tweet. The feature expansion process with T as a tweet is as follows [25]:

1. Let $f_v = \{t_1, t_2, \ldots, t_n\}$ denote the feature vector of tweet T.
2. For each $t_i \in f_v$ and $t_i = 0$
    a. Retrieve words W that are semantically similar from the GloVe list.
    b. If at least one of the words in W is present in tweet T, the corresponding feature vector will be assigned a value of 1, i.e., $t_i \leftarrow 0$.

As an illustration, consider the sentence in the tweet "Indonesia wujudkan pemilu 2024 dengan aman". Suppose the word "damai" has a feature value of 0 in the tweet representation, with similar words generated by the GloVe listed in Table 7. Since the word "aman" appears in the tweet content and is considered similar to "damai" according to Table 7, the feature value corresponding to "damai" in the tweet representation is changed to 1.

Long short-term memory (LSTM)

The classification model used in this study is Long Short-Term Memory (LSTM). Long Short-Term Memory (LSTM) is an advancement of the Recurrent Neural Network (RNN) technique [27], effectively utilizing memory cells [28]. LSTM was introduced as an architecture capable of providing excellent performance in machine translation [29]. The advantage of LSTM in understanding long-term temporal dependencies with gradient-based optimization makes it highly effective in text classification [10]. There are four main components of LSTM: forget gate, input gate, memory state, and output gate. When the forget gate operates, information data is analyzed and evaluated to determine whether it should be retained or discarded in the memory cell. The input gate then determines the value to be updated. The new value vector is processed through the Tanh function, and the result is stored in the memory cell. Subsequently, the memory cell replaces the previous data with the new value. This new value results from the combination of the forget gate and input gate outputs. In the final step, the output gate decides whether the value in the memory cell will be used. Using the Tanh function, the output gate transforms the memory cell value to assess its relevance [30]. Through these gates, LSTM can store initial input data units in the memory cell, retain forgotten data units in the existing memory cell, and store new output data units in the memory cell [10]. The architecture of LSTM can be seen in Figure 3.

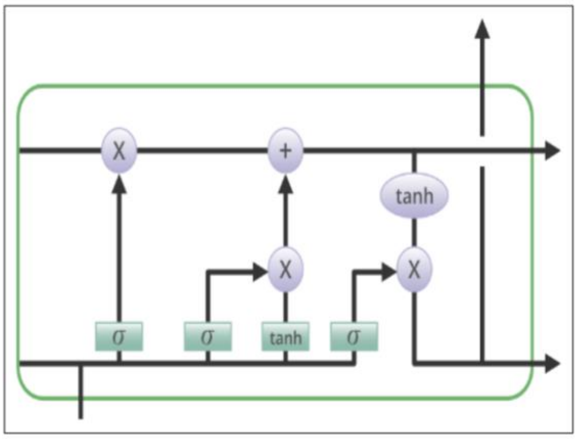

Figure 3. Long short-term memory architecture [27]

In this study, the LSTM model for classification was implemented using the TensorFlow library in Python. The LSTM model consisted of three LSTM layers, a dropout layer, and a dense layer. The first LSTM layer was the input layer (units=32, activation='tanh', recurrent_activation='sigmoid', return_sequences=True). The following two LSTM layers were used as hidden layers (units=32, activation='tanh', recurrent_activation='sigmoid', return_sequences=True). The dropout layer was applied to prevent overfitting with a dropout rate of 0.5. The dense layer (units=1, activation='sigmoid') acted as the output layer, producing a single output for binary classification to classify the credibility of information.

Particle swarm optimization (PSO)

Particle Swarm Optimization (PSO) utilizes a group of particles, with each particle's position and velocity being modified in reference to the best-performing particle to reach a solution [16]. This optimization method is known for its tendency to reach optimal solutions faster than other methods and is easy to implement [15][16]. PSO mimics the social behavior of a flock of birds or a school of fish. For example, consider a group of birds flying randomly in search of food in an area where only one piece of food is present. Although none of the birds know the exact location of the food, they know how far the food is from their respective search positions. The best strategy for finding the food is to follow the bird closest to the food's location. In PSO, each bird represents a solution in the problem space and is called a "particle". Each particle has a fitness value that needs to be optimized and a velocity that adjusts as it moves, influenced by the search area and is stored as the best position to be achieved. The fitness value is referred to as p_{best}, and when a particle considers the entire population as its neighbor, the best position is called g_{best}. Thus, the particle updates itself to generate the next generation of the swarm [15]. In this study, PSO is used to optimize the LSTM classification model. PSO can help find the best parameters for the LSTM model, including learning rate, dropout, neurons, and layers. The value intervals used for each parameter are learning rate (0.0001 to 0.1), dropout rate (0.1 to 0.5), units (8 to 128), and layers (1 to 5). The working system of PSO in optimizing these hyperparameters is illustrated in Figure 4.

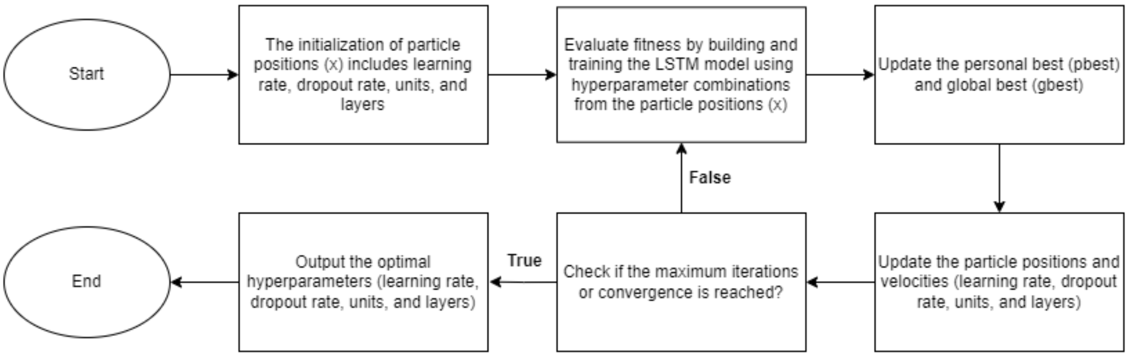
Start
The initialization of particle positions (x) includes learning rate, dropout rate, units, and layers
Evaluate fitness by building and training the LSTM model using hyperparameter combinations from the particle positions (x)
Update the personal best (pbest) and global best (gbest)
Update the particle positions and velocities (learning rate, dropout rate, units, and layers)
Check if the maximum iterations or convergence is reached?
False
True
Output the optimal hyperparameters (learning rate, dropout rate, units, and layers)
End

Figure 4. Particle swarm optimization work system

Performance evaluation

The performance of the LSTM classification method is assessed using the Confusion Matrix method in this research. The Confusion Matrix is a visualization table that measures the performance of a learning algorithm in accomplishing its task [31]. The Confusion Matrix includes four outputs: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). True Positive refers to positive data correctly predicted as positive. False Positive refers to negative data predicted as positive. True Negative is negative data correctly predicted as negative. False Negative is positive data predicted as negative. Performance evaluation is conducted by calculating the values of recall, precision, F1-score, and accuracy. Recall is the ratio of correctly classified positive data to the total positive data. Precision is the ratio of correctly classified positive data to the total predicted positive data. The F1-Score, also known as the F1-measure, compares recall and precision simultaneously. Accuracy is the ratio of correctly classified data to the total data. In simple terms, accuracy reflects how accurately a model classifies data correctly. The recall, precision, F1-score, and accuracy formulas are as follows.

$$Recall = \frac{TP}{TP+FN} \tag{2}$$

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

$$F1 - Score = 2 \times \frac{precision \times recall}{precision+recall} \tag{4}$$

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{5}$$

RESULTS AND DISCUSSIONS

This study employs LSTM as the classification model, TF-IDF as the baseline, RoBERTa for semantic features, GloVe for feature expansion, and PSO as the optimization method. The data is shuffled five times for each data split ratio. The credibility detection system is built with five scenarios to achieve the highest accuracy. The resulting accuracy is the average of 5 data sets. The first scenario determines the data split ratio and tests parameters on the baseline for subsequent scenarios. The second scenario compares the best baseline for the following scenario. The third scenario adds semantic features from RoBERTa and combines them with the baseline. The fourth scenario compares and determines the best accuracy from the system constructed using the GloVe corpus from tweets, IndoNews, and a combination of tweets and IndoNews. The final scenario applies the PSO method to the LSTM classification model to find the optimal hyperparameters, thus achieving the best accuracy.

In the first scenario, tests were conducted to determine the best data split ratio using the LSTM model and TF-IDF as the baseline with Unigram and feature thresholds (min_df = 5, max_df = 0.7), using all data features. The compared data split ratios were 90:10, 80:20, and 70:30. Table 8 represents the results of this scenario, with the 80:20 split ratio using the minimum and maximum feature values yielding the highest accuracy of 88.24%. Consequently, the 80:20 ratio was used to test the parameter for the number of data features in TF-IDF. The feature counts compared were 3000, 5000, 7000, and all data features. Table 9 presents the results of this parameter testing, where the highest accuracy of 88.24% was consistently achieved using all features. Additionally, tests were conducted on the minimum and maximum feature values (min_df and max_df) for TF-IDF, with a range of 1-5 for min_df and 0.5-0.9 for max_df. These tests resulted in the highest accuracy of 88.32%, with values of 4 for min_df and 0.7 for max_df. All the results from the first scenario were applied to the subsequent scenarios.

Table 8. Data ratio testing result in scenario 1

Data Split	Accuracy (%)	
	With Min and Max Features	Without Min and Max Features
90:10	88.03	87.90
80:20	**88.24**	87.83
70:30	87.79	87.50

Table 9. Feature count parameter testing result in scenario 1

Feature Amount	Accuracy (%)
3000	88.16
5000	88.20
7000	88.23
All Features	**88.24**

The second scenario compares various baselines to obtain the highest accuracy. The baselines used are Unigram, Bigram, Trigram, and a combination of them. Table 10 shows the results of the baseline comparison. According to Table 10, Unigram consistently achieves the highest accuracy at 88.32% compared to the other baselines. This result will be used for the following scenario.

Table 10. Comparison of baseline result in scenario 2

Baseline	Accuracy (%)
Unigram	**88.32**
Bigram	83.84
Trigram	77.29
Unigram + Bigram	87.48
Unigram + Trigram	87.84
Unigram + Bigram + Trigram	83.59

In the third scenario, RoBERTa was used to add semantic features. The results from RoBERTa's semantic features were then combined with the baseline. Table 11 shows the combination of these baselines and RoBERTa's semantic features. This combination led to an enhanced accuracy of 88.40%, marking a 0.18% improvement over the accuracy achieved by the baseline model alone. Thus, this combination was applied in the following scenario.

Table 11. Comparison results in scenario 3

Combination	Accuracy (%)
LSTM + Baseline	88.32 (+0.09)
LSTM + Baseline + RoBERTa	**88.40 (+0.18)**

Feature expansion was applied in the fourth scenario. In this scenario, the results from the previous scenario were enhanced by expanding features using GloVe and leveraging a corpus built by identifying similar words to connect them with the features. The corpus was constructed from tweets, IndoNews, and a combination of tweet+IndoNews, with word similarity rankings, including the top one, top five, top ten, and top fifteen similar words. This scenario compared combinations of the baseline with the tweet corpus, baseline with the IndoNews corpus, and baseline with the tweet+IndoNews corpus. The comparison results are shown in Table 12, indicating that combining the baseline and the IndoNews corpus with the top one similar word resulted in the highest accuracy of 88.85%.

Table 12. Comparison results in scenario 4

Rank	Accuracy (%)		
	Baseline + Tweet	Baseline + IndoNews	Baseline + Tweet IndoNews
Top 1	88.84 (+0.68)	**88.85 (+0.69)**	88.68 (+0.5)
Top 5	88.57 (+0.37)	88.59 (+0.4)	88.55 (+0.34)
Top 10	88.63 (+0.44)	88.45 (+0.24)	88.43 (+0.22)
Top 15	88.44 (+0.23)	88.46 (+0.25)	88.48 (+0.27)

In the last scenario, the Particle Swarm Optimization (PSO) method was applied for optimization. This method is aimed to optimize the LSTM model's hyperparameters to improve accuracy and performance. The optimized hyperparameters consisted of the learning rate (0.0001 to 0.1), dropout rate (0.1 to 0.5), units (8 to 128), and layers (1 to 5). For the PSO process, ten populations and five iterations were used. Table 13 presents the results of this assessment. It demonstrates that the LSTM model, with its hyperparameters optimized using PSO, achieved a higher accuracy of 89.09% compared to its performance without optimization. The optimal hyperparameters from the testing encompass a learning rate of 0.0001, a dropout rate of 0.143, 8 units, and five layers comprising of an input layer, two hidden layers, a dropout layer, and a dense layer as the output layer.

Table 13. Comparison results in scenario 5

Model	Hyperparameter	Accuracy (%)
LSTM Without Optimization	Learning Rate: 0.0001 Dropout Rate: 0.5 Units: 32 Layer: 5	88.85 (+0.69)
LSTM With PSO	**Learning Rate: 0.0001** **Dropout Rate: 0.143** **Units: 8** **Layer: 5**	**89.09 (+0.96)**

The results align with the researchers' expectations by balancing the data distribution between credible and non-credible categories and applying the five scenarios mentioned above. The combination of LSTM as the classification model, determining the best data split ratio, selecting the best baseline, adding semantic features with RoBERTa, constructing various corpus with GloVe, and optimizing LSTM hyperparameters with PSO resulted in the best accuracy. Figure 5 is the Confusion Matrix, and Table 14 presents the evaluation metrics results derived from the Confusion Matrix in the best testing. Additionally, the graph of accuracy improvement for each scenario can be observed in Figure 6.

Table 14. Evaluation metrics results from the best testing

Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
88.99	90.05	89.52	89.09

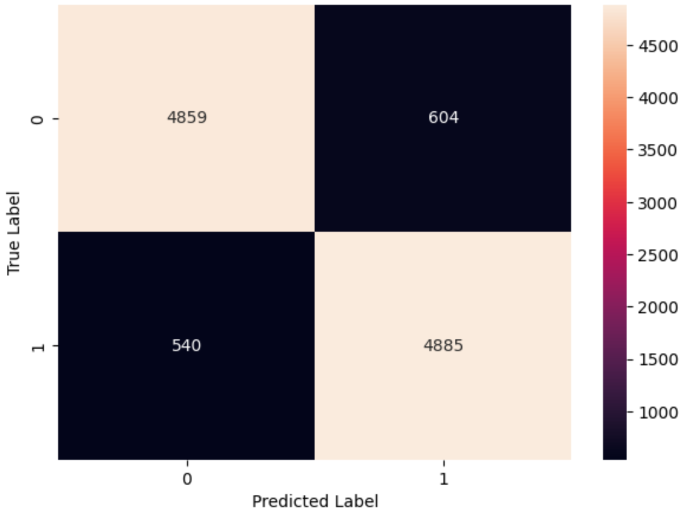

Figure 5. Confusion matrix from the best testing

Figure 5 shows the Confusion Matrix, indicating 4885 True Positives (TP) and 604 False Positives (FP). The True Negatives (TN) are 4859, with 540 False Negatives (FN). These calculations resulted in optimal accuracy due to the alignment of TP and TN values, signifying accurate predictions, with TP representing credible and TN representing non-credible.

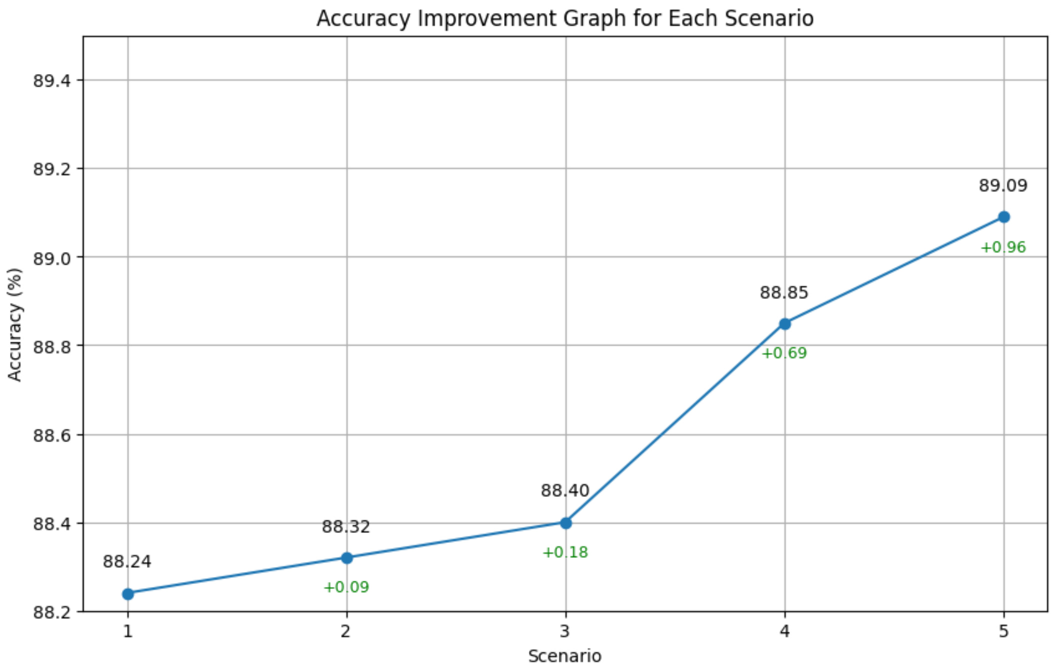

Figure 6. Accuracy improvement graph for each scenario

Figure 6 shows an increase in accuracy for each scenario. The accuracy improvement is calculated using the formula: (current scenario accuracy - baseline accuracy in scenario 1) / baseline accuracy * 100. In the first scenario, using the baseline (Unigram) and LSTM model with the best data split ratio and parameters resulted in the highest accuracy of 88.24%. The results of the first scenario were then utilized in the second scenario, comparing different baseline combinations, resulting in the best accuracy of 88.32%, with an improvement of 0.09%. Compared to the previous baseline, the third scenario added semantic features from RoBERTa to the baseline. The addition of RoBERTa semantic features yielded a higher accuracy of 88.40%, with an improvement of 0.18%. In the fourth scenario, feature expansion was conducted with GloVe from the previous scenario's baseline, constructing a corpus from tweets, IndoNews, and a combination of tweets+IndoNews. The best result was an accuracy of 88.85% on the IndoNews corpus with the top one similar word, showing an improvement of 0.69%. Lastly, in the fifth scenario, PSO was performed to optimize the hyperparameters for the LSTM model. As a result, the accuracy increased to 89.09%, with an improvement of 0.96%.

The study assessed the significance of accuracy changes in each scenario using the Z-Value and P-Value for statistical significance testing. The aim was to establish the statistical significance of the accuracy changes in each scenario. A 95% confidence level was used for the Z-Value. If Z-Value > 1.96 and P-Value < 0.01, the change in accuracy was considered highly significant. The change was considered significant if Z-Value > 1.96 and P-Value < 0.05. Otherwise, the change in accuracy was not significant. Table 15 presents the results of the statistical significance testing for all scenarios. According to Table 15, the accuracy changes from scenarios 3 to 4 and from scenarios 4 to 5 showed statistically significant improvements. This demonstrates that feature expansion and optimization can significantly enhance the accuracy and performance of the model. Additionally, this study's progression from the baseline scenario 1 to the best model in scenario 5 showed a highly significant statistical improvement, with a Z-Value of 10.932 and a P-Value of 0.000.

Table 15. Results of statistical significance tests in all scenarios

	S1→S2	S2→S3	S3→S4	S4→S5	S1→S5
Z-Value	0.585	0.331	2.442	2.880	10.932
P-Value	0.559	0.740	0.015	0.004	0.000
Significant?	False	False	True	True	True

CONCLUSION

This research detects the credibility of information on social media X using a dataset of 54,766 Indonesian tweets discussing the 2024 General Election. The applied methods include LSTM as a classification model, TF-IDF for feature extraction and a baseline, RoBERTa for semantic feature augmentation, GloVe for feature expansion, and PSO as an optimization method. The combination of these methods yielded the highest accuracy rate of 89.09%, representing a 0.96% improvement from the baseline model. The result and statistically significant tests support the conclusion that combining several methods, such as feature extraction, semantic features, feature expansion, and optimization, can significantly enhance the accuracy and efficiency of the model. Additionally, with TF-IDF, the more features used, the higher the accuracy. However, with GloVe feature expansion, using more corpuses does not always result in optimal accuracy. The application of PSO to find the optimal hyperparameters for LSTM provided the highest accuracy improvement compared to other methods, indicating that optimization methods play a significant role in achieving optimal accuracy. The optimal hyperparameters found were a learning rate of 0.0001, a dropout rate of 0.143, 8 units, and five layers consisting of an input layer, two hidden layers, a dropout layer, and a dense layer as the output. As a recommendation, future research could expand the dataset with more general topics and use various languages. Further exploration of method combinations and parameter testing can also be conducted to achieve more optimal results.

REFERENCES

[1] D. Wang and Y. Chen, “A neural computing approach to the construction of information credibility assessments for online social networks,” *Neural Comput Appl*, vol. 31, no. S1, pp. 259–275, Jan. 2019, doi: 10.1007/s00521-018-3734-4.

[2] M. Azer, M. Taha, H. H. Zayed, and M. Gadallah, “Credibility Detection on Twitter News Using Machine Learning Approach,” *International Journal of Intelligent Systems and Applications*, vol. 13, no. 3, pp. 1–10, Jun. 2021, doi: 10.5815/ijisa.2021.03.01.

[3] S. A. Floria, F. Leon, and D. Logofătu, “A credibility-based analysis of information diffusion in social networks,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 2018, pp. 828–838. doi: 10.1007/978-3-030-01424-7_80.

[4] S. F. Sabbeh and S. Y. Baatwah, “Arabic News Credibility on Twitter: An Enhanced Model Using Hybrid Features,” *J Theor Appl Inf Technol*, vol. 96, no. 8, 2018, [Online]. Available: www.jatit.org

[5] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier, “TweetCred: Real-Time Credibility Assessment of Content on Twitter,” *Proceedings of Social Informatics (SocInfo 2014)*, May 2014.

[6] A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi, “Faking Sandy: Characterizing and Identifying Fake Images on Twitter during Hurricane Sandy,” in *Proceedings of the 22nd International Conference on World Wide Web*, New York, NY, USA: ACM, May 2013, pp. 729–736. doi: 10.1145/2487788.2488033.

[7] C. Castillo, M. Mendoza, and B. Poblete, “Information credibility on twitter,” in *Proceedings of the 20th international conference on World wide web*, New York, NY, USA: ACM, Mar. 2011, pp. 675–684. doi: 10.1145/1963405.1963500.

[8] M. R. Morris, S. Counts, A. Roseway, A. Hoff, and J. Schwarz, “Tweeting is believing?: Understanding microblog credibility perceptions,” in *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, New York, NY, USA: ACM, Feb. 2012, pp. 441–450. doi: 10.1145/2145204.2145274.

[9] E. B. Setiawan, D. H. Widyantoro, and K. Surendro, “Measuring information credibility in social media using combination of user profile and message content dimensions,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 4, p. 3537, Aug. 2020, doi: 10.11591/ijece.v10i4.pp3537-3549.

[10] P. Vyas and O. F. El-Gayar, “Credibility Analysis of News on Twitter using LSTM: An exploratory study ,” *AMCIS 2020 Proceedings*, 2020.

[11] I. Fadhli, L. Hlaoua, and M. N. Omri, “Deep learning-based credibility conversation detection approaches from social network,” *Soc Netw Anal Min*, vol. 13, no. 1, Dec. 2023, doi: 10.1007/s13278-023-01066-z.

[12] Diaz Tiyasya Putra and Erwin Budi Setiawan, “Sentiment Analysis on Social Media with Glove Using Combination CNN and RoBERTa,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 3, pp. 457–563, Jun. 2023, doi: 10.29207/resti.v7i3.4892.

[13] A. R. I. Fauzy and Erwin Budi Setiawan, “Detecting Fake News on Social Media Combined with the CNN Methods,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 2, pp. 271–277, Mar. 2023, doi: 10.29207/resti.v7i2.4889.

[14] M. Roondiwala, H. Patel, and S. Varma, “Predicting Stock Prices Using LSTM,” *Article in International Journal of Science and Research*, vol. 6, 2017, doi: 10.21275/ART20172755.

[15] A. U. Azmi, R. Hidayat, M. Z. Arif, and J. Matematika, “Perbandingan Algoritma Particle Swarm Optimization (PSO) dan Algoritma Glowworm Swarm Optimization (GSO) Dalam Penyelesaian Sistem Persamaan Non Linier (Comparison of Particle Swarm Optimization (PSO) and Glowworm Swarm Optimization (GSO) Algorithms in Solving Non Linear Equation System)”, [Online]. Available: https://jurnal.unej.ac.id/index.php/MIMS/index

[16] K. F. Irnanda, A. P. Windarto, and I. S. Damanik, “Optimasi Particle Swarm Optimization Pada Peningkatan Prediksi dengan Metode Backpropagation Menggunakan Software RapidMiner,” *JURIKOM (Jurnal Riset Komputer)*, vol. 9, no. 1, p. 122, Mar. 2022, doi: 10.30865/jurikom.v9i1.3836.

[17] R. A. Rudiyanto and E. B. Setiawan, “Sentiment Analysis Using Convolutional Neural Network (CNN) and Particle Swarm Optimization on Twitter,” *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 9, no. 2, pp. 188–195, Feb. 2024, doi: 10.33480/jitk.v9i2.5201.

[18] J. L. Fleiss, B. Levin, and M. C. Paik, *Statistical methods for rates and proportions*. john wiley & sons, 2013.

[19] R. Siringoringo, “Klasifikasi data tidak Seimbang menggunakan algoritma SMOTE dan k-nearest neighbor,” *Journal Information System Development (ISD)*, vol. 3, no. 1, 2018.

[20] K. U. Wijaya and E. B. Setiawan, “Hate Speech Detection Using Convolutional Neural Network and Gated Recurrent Unit with FastText Feature Expansion on Twitter,” *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 3, pp. 619–631, 2023, doi: 10.26555/jiteki.v9i3.26532.

[21] M. N. Saadah, R. W. Atmagi, D. S. Rahayu, and A. Z. Arifin, “Information Retrieval of Text Document with Weighting TF-IDF and LCS,” *Jurnal Ilmu Komputer dan Informasi*, vol. 6, no. 1, p. 34, Oct. 2013, doi: 10.21609/jiki.v6i1.216.

[22] N. Hassan, W. Gomaa, G. Khoriba, and M. Haggag, “Credibility Detection in Twitter Using Word N-gram Analysis and Supervised Machine Learning Techniques,” *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 1, pp. 291–300, Feb. 2020, doi: 10.22266/ijies2020.0229.27.

[23] A. Barua, S. Thara, B. Premjith, and K. P. Soman, “Analysis of Contextual and Non-contextual Word Embedding Models for Hindi NER with Web Application for Data Collection,” in *Communications in Computer and Information Science*, Springer Science and Business Media Deutschland GmbH, 2021, pp. 183–202. doi: 10.1007/978-981-16-0401-0_14.

[24] Y. Liu *et al.*, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” Jul. 2019, [Online]. Available: http://arxiv.org/abs/1907.11692

[25] E. B. Setiawan, D. H. Widyantoro, and K. Surendro, “Feature expansion using word embedding for tweet topic classification,” in *2016 10th International Conference on Telecommunication Systems Services and Applications (TSSA)*, IEEE, Oct. 2016, pp. 1–5. doi: 10.1109/TSSA.2016.7871085.

[26] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global Vectors for Word Representation.” [Online]. Available: http://nlp.

[27] D. P. Putra and E. B. Setiawan, “Hoax Detection Using Long Short-Term Memory (LSTM) and Gate Recurrent Unit (GRU) on Social Media,” *Building of Informatics, Technology and Science (BITS)*, vol. 4, no. 4, Mar. 2023, doi: 10.47065/bits.v4i4.3084.

[28] W. K. Sari, D. P. Rini, and R. F. Malik, “Text Classification Using Long Short-Term Memory With GloVe Features,” *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 5, no. 2, p. 85, Feb. 2020, doi: 10.26555/jiteki.v5i2.15021.

[29] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau, “A C-LSTM Neural Network for Text Classification,” Nov. 2015, [Online]. Available: http://arxiv.org/abs/1511.08630

[30] M. N. Ibnu Sina and E. B. Setiawan, "Stock Price Correlation Analysis with Twitter Sentiment Analysis Using The CNN-LSTM Method," *sinkron*, vol. 8, no. 4, pp. 2190–2202, Oct. 2023, doi: 10.33395/sinkron.v8i4.12855.

[31] B. P. Salmon, W. Kleynhans, C. P. Schwegmann, and J. C. Olivier, "Proper comparison among methods using a confusion matrix," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, IEEE, Jul. 2015, pp. 3057–3060. doi: 10.1109/IGARSS.2015.7326461.