# Sentence embedding to improve rumour detection performance model

**Rini Anggrainingsih[1], Endar Suprih Wihidayat[2], Bambang Widoyono[3]**
[1]Informatics Department, Sebelas Maret University, Surakarta, Indonesia
[2]Informatics Education Department, Sebelas Maret University, Surakarta, Indonesia
[3]Master of Information Technology, Faculty of Computer Science, University of Indonesia, Jakarta, Indonesia

**Article Info**

***Article history:***

Received Nov 24, 2022
Revised Jan 18, 2023
Accepted Mar 10, 2023

***Keywords:***

Bidirectional encoder representation of transformer
Feature extraction
Rumour detection
Sentence embedding
Text classification

**ABSTRACT**

Recently, most individuals have preferred accessing the most recent news via social media platforms like Twitter as their primary source of information. Moreover, Twitter enables users to post and distribute tweets quickly and unsupervised. As a result, Twitter has become a popular platform for disseminating false information, such as rumours. These rumours were then propagated as accurate and influenced public opinion and decision-making. The issue will arise when a decision or policy with substantial consequences is made based on rumours. To avoid the negative impacts of rumours, several researchers have attempted to detect them automatically as early as feasible. Previous studies employed supervised learning methods to identify Twitter rumours and relied on feature extraction algorithms to extract tweet content and context elements. However, manually extracting features is time-consuming and labour-intensive. To encode each tweet's sentence as a vector based on its contextual meaning, we proposed utilising Bidirectional Encoder Representation of Transformer (BERT) as a sentence embedding. We then used these vectors to train some classifier models to detect rumours. Finally, we compared the performance of BERT-based models to feature engineering-based models. We discovered that the suggested BERT-based model improved all parameters by around 10% compared to the feature engineering-based classification model.



***Corresponding Author:***

Rini Anggrainingsih
Informatics Department, Sebelas Maret University
Ir Sutami St. 36 A, Kentingan, Jebres, Surakarta, Indonesia
Email: rini.anggrainingsih@staff.uns.ac.id

## 1. INTRODUCTION

Much false information spread worldwide swiftly due to the difficulty of proper control on social media platforms like Twitter. People often post and distribute breaking news without verifying its accuracy, leading to the widespread sharing of captivating but deceptive content. Consequently, such content may be shared thousands of times, despite containing misleading information. The most prevalent phrase for false information on the Internet is a rumour. A rumour appears to be a credible story, yet it is not easy to confirm. The rumours are of dubious veracity and provoke concern or skepticism among the audience [1], [2]. A characteristic of a rumour is difficult to confirm because it may be accurate, partially true, false, or unsubstantiated [3]. This study focuses on rumours transmitted through the Twitter network.

After reviewing the existing literature, we found that most methods for detecting rumours on Twitter employ supervised learning algorithms that rely on extracting features. They extracted features from both content and context of tweets [1]–[5]. The context-based feature components include information about tweets' surroundings, such as user and network information [2]–[7]. The content-based feature involves extracting features from the text of tweets, especially those related to language, like lexical, syntactic, and semantic features [6]–[9]. Unfortunately, manual feature extraction is ineffective and time-consuming. Moreover, Twitter does not always provide the supplementary data necessary for feature extraction beyond the tweet's text [10].

In recent years, transfer learning with pre-trained language models, such as Bidirectional Encoder Representation from Transformer (BERT), has become a powerful technique in natural language processing (NLP) [11], [12]. This method employs an encoder to encode a sentence into an embedding vector using an attention mechanism [13] to derive a numeric representation of a text that enables a computer to comprehend the context and meaning of the text [11]. This study aims to enhance the performance of classifier models in identifying rumours on Twitter by proposing a novel model that utilises BERT and neural networks as sentence embedding and classifiers in detecting rumours on Twitter and comparing the model's performance between feature engineering-based vectors and sentence embedding-based vectors to detect rumour on Twitter.

The structure of the study is: Section 2 investigates previous attempts at detecting rumour, and section 3 details our suggested approach for utilising BERT to identify misleading tweet information. Then, section 4 presents our experimental results and compares them to recent studies. Lastly, in section 5, the study concludes with a summary of our findings.

## 2. LITERATURE REVIEW

The majority of previous research on fake information detection employed supervised computer models to classify tweets as rumour or non-rumour based on extracted content and contextual features [6], [7], [10]–[12]. Context-based techniques extract features by considering information about tweets, such as user and network data. Table 1 illustrates the context-based elements derived from tweets and the studies that employed them. The content-based techniques extract features from tweets, particularly language characteristics such as lexical, syntactic, and semantic characteristics that indicate how words were employed in a tweet. For example, previous research suggested that terms of ambiguity, denial, conciseness, and brevity may disclose the legitimacy of a tweet [1]. Table 2 depicts the content-based features and their application in the research.

Table 1. Contextual characteristics retrieved from tweets

| Contextual-based features | |
|---|---|
| 1. Verified account or not [4], [5] | 6. Having over 500 followers [14] |
| 2. Has a description or not [5] | 7. Post on a day or weekday [4], [5] |
| 3. Has a URL or not [4], [6] | 8. Number of tweets [4], [5], [9] |
| 4. Followers [4], [6], [9], [14] | 9. Is it retweeted or not [4]–[6] |
| 5. Number of friends [4], [6], [9] | |

Table 2. List of tweet features based on their content

| Text content-based features | |
|---|---|
| 1. Hashtags [4]–[6], [9] | 16. The number of smile emote [5], [6] |
| 2.Words length [5], [6], [9] | 17. The number of frown emote [5], [6] |
| 3.Characters length [5], [6] | 18. Number of sentiment (+) words [5] |
| 4. Contains 100 top domain [2] | 19. number of sentiment (-) words [4], [5] |
| 5. Is it contains URL [4], [6] | 20. Sentiment score [4], [5] |
| 6. The number of URLs [4]–[6] | 21. The number of 1$^{st}$ pronouns [5], [15] |
| 7. Mention news agency [14] | 22. The number of 2$^{nd}$ pronouns [5], [15] |
| 8. The number of mention users [5], [6] | 23. The number of 3$^{rd}$ pronouns [5], [15] |
| 9. Contains stock symbol [2] | 24.The number of temporal reference [15] |
| 10. Contains numbers [14] | 25.The number of lexical density [15] |
| 11. Contains selected users [2] | 26. Slang Terminology [14] |
| 12. Uppercase [2], [6] | 27. The number of intensifiers [14] |
| 13. Question mark [5], [6] | 28. Contains repeated characters [14] |
| 14. Exclamation mark [2], [3], [5], [6] | 29. Contains all uppercase word [14] |
| 15. Contains multi '?' or '!' [2], [3], [6] | 30. Title capitalisation [14] |

Content-based or context-based manual extraction tasks to classify rumour tweets take a lot of time and are hard to do. For this reason, recent studies have used neural networks (NN) techniques to sort tweets about rumours. In the finding false information context, recurrent neural network-based (RNN) frameworks are used a lot [10], [16], [17] and convolutional neural networks (CNN) [18], [19]. Alkhodair *et al.* reported the recent performance of the RNN model for rumour detection, which got 71.6% and 83.9% F1 scores for the rumour and non-rumour classes, respectively [17]. The most recent CNN model for classifying rumours, presented by Bharti and Jindal *et al.* did the best job and got a weighted average F1-score of 0.84. [19].

Other researchers, like Ajao *et al.* [20], employed a hybrid framework using a combination of CNN and long short-term memory (LSTM) to automatically extract features from a Twitter post without any prior knowledge of the subject area or topic of discussion to identify fake news on Twitter. Their model achieved an accuracy of 82.29% for all classes but only a precision score of 44.35%. Other researchers, Kotteti *et al.* sought to improve the performance of supervised learning models in detecting rumours by reducing the time required for detection. To achieve this, they proposed a strategy that analyses multiple time-series data to utilise temporal aspects of tweets instead of relying on the content, which requires feature selection and text mining [9]. They used the Gaussian Naive Bayes classifier to implement their proposed approach, which made computations easier and achieved a high precision score of 94%. However, their method only scored 35.6% for recall and 51.8% for F1-score.

Xu *et al*. proposed a new algorithm for detecting fake news on Twitter called the topic-driven novel detection (TDRD) algorithm [21]. They were inspired by a communication theory that suggests the topic of a post can indicate whether it is likely to be spread as a rumour. The TDRD algorithm classified tweet topics and incorporated them into a deep-learning framework for rumour detection. The authors employed the CNN model, which achieved an accuracy of 82.66%, the highest among their experimental results.

BERT is a unique language model created by Google AI that uses a deep bidirectional transformer to extract information from unlabeled text. It combines both left and right context representations of a token from all layers to capture relationships between words and create a vector representation for each word based on its relationship with other words in the phrase [11]. This allows BERT to infer the meaning of a word from its surrounding context. For example, the vector for the word "apple" in the sentences "I got a new apple tablet" and "I have a fresh apple" would differ. BERT comes in two versions: BERT-Base, which has twelve transformer blocks, and BERT-Large, which has twenty-four transformer blocks. This study used BERT-Base, resulting in 768 vector arrays for each sentence.

## 3. MATERIAL AND METHOD

The model proposed in this study involves several steps for detecting rumours using BERT. Firstly, BERT is used to generate sentence embeddings that represent each tweet as a vector based on its contextual meaning and linguistic patterns. Next, these vectors are utilised for training different classifier models for rumour detection. Finally, the results obtained from the proposed BERT-based method are compared with those obtained using traditional feature engineering techniques. Figure 1 illustrates the overall process of the proposed rumour classification model that uses BERT's sentence embeddings.

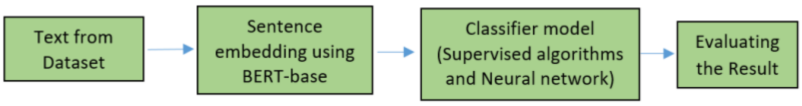


Figure 1 The steps of the proposed model for rumour detection using BERT

### 3.1. Dataset

Due to the complexities of data collection procedures, there are few publicly available datasets on rumour classification [1]. Therefore, to validate our models, we obtained datasets from the PHEME project [5], which is considered a benchmark and publicly accessible over the Internet. This dataset contains rumour-tagged (1,969 tweets) and non-rumour-tagged (3,822 tweets). We allocate 70% of each dataset class for training and 30% for testing.

### 3.2. Classifier model

We trained different supervised-classifier models and a simple neural network model (MLP) using BERT-embedded and feature-based vectors from tweet text and then compared their results. An MLP is made up of a layer for receiving signals, a layer for making predictions, and any number of hidden layers that work as the MLP's computing engine [22]. We used some supervised learning approaches that are widely known as eminent methods in text classification [23]. Those supervised models included support vector machines (SVM), logistic regression (LR), Naive Bayes classifier (NBC), AdaBoost, and k-nearest neighbors are some of the supervised classifier models (KNN).

### 3.3. Evaluation model

We evaluated our model using a confusion matrix and the following formulas to calculate its Accuracy, Precision, Recall, and F1 scores. The confusion matrix measures the performance of a model by comparing its predictions against the actual outcomes. The four key metrics derived from the confusion matrix are,

- True-positive (TP) : Tweets that are correctly predicted as non- rumour tweets.
- False-negative (FN) : Rumour tweets that are wrongly identified as non-rumour tweets.
- False-positive (FP) : Non-rumour tweets that are wrongly identified as rumour tweets.
- True-negative (TN) : Non-rumour tweets that are correctly predicted as non-rumour tweets.

$$\text{Accuracy (A)} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{1}$$

$$\text{Precision (P)} = \frac{\text{TP}}{\text{TP+FP}} \tag{2}$$

$$\text{Recall (R)} = \frac{\text{TP}}{\text{TP+FN}} \tag{3}$$

$$\text{F1} = \frac{2 \text{ x (Precision x Recall)}}{\text{Precision+Recall}} \tag{4}$$

### 3.4. Experiment steps

By utilising SKlearn [24] and PyTorch [25], a well-known library for machine learning and deep learning tasks, we experimented with feature engineering-based techniques and sentence embedding using BERT to recognise a rumour tweet and compare these approaches' performance. Figure 2 shows our procedures in our experiment to discriminate between rumour and non-rumour tweets. First, we preprocessed and tokenised the tweets using BERT to provide the tokenised form of the tweets for the proposed approach. The tokenised sentences were then transformed into vectors using BERT-base and Sentence Transformer. Finally, the vectors mentioned at the second step were employed for model training, which encompassed algorithms such as AdaBoost, k-nearest neighbors, support vector machines (SVM), logistic regression (LR), Naive Bayes classification (NBC), and a four layers perceptron (4L-MLP).

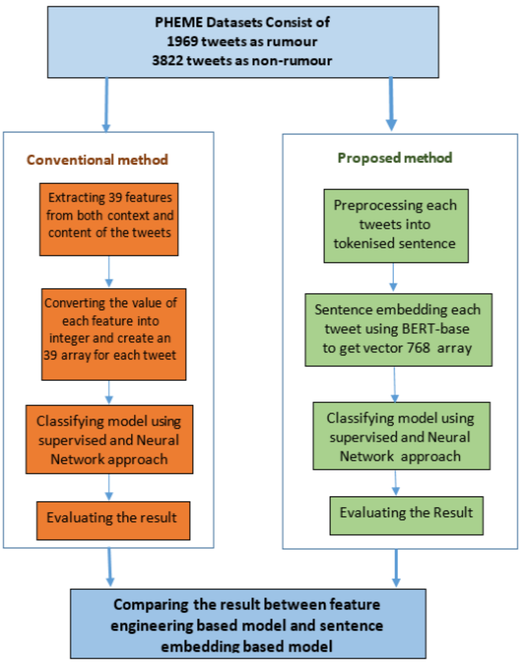


Figure 2. The experiment steps to detect rumour

In addition, we employ the feature engineering technique by extracting 39 characteristics from the context and content of tweets, as shown in Tables 1 and 2. Then, we transform the values of those features to integer data types and generate a 39-element array of features for each tweet. Finally, we used these vectors to train and compare all the similar models we trained with BERT-embedding vectors and evaluate the performance result using (1) to (4).

## 4. RESULT

We analysed and compared the performance of these classifier models by examining and comparing their confusion matrices. The confusion matrix for each model's prediction outcomes is depicted in Table 3. We evaluated the performance of each classifier model based on the predictions in Table 3. Table 4 compares the performance of classifier models based on BERT and classifier models based on feature engineering, revealing that BERT-based classifier models perform better than feature engineering-based classifier models for all parameters. Each model's accuracy and precision improved by approximately 10% on average by employing BERT vectors. In addition, a basic neural network utilising 4-MLP earned the best performance across all classes. These findings provide a positive outlook on the use of BERT sentence embedding as a viable approach for identifying rumour tweets, as it has shown the ability to minimise the effort needed for rumour detection by eliminating the need for text feature extraction. In simpler terms, the suggested technique has demonstrated its potential in streamlining the process of identifying rumours in tweets.

Table 3. Confusion matrix result for each classifier model

| Classifier | Approach | Prediction | Non-Rumours | Rumours |
|---|---|---|---|---|
| Support Vector Machine | BERT | Non-Rumours | 1000 | 155 |
| | | Rumours | 160 | 422 |
| | 39 Features | Non-Rumours | 1035 | 332 |
| | | Rumours | 125 | 245 |
| Logistic Regression | BERT | Non-Rumours | 1020 | 154 |
| | | Rumours | 140 | 423 |
| | 39 Features | Non-Rumours | 1037 | 332 |
| | | Rumours | 123 | 245 |
| Naive Bayes | BERT | Non-Rumours | 835 | 131 |
| | | Rumours | 325 | 446 |
| | 39 Features | Non-Rumours | 645 | 129 |
| | | Rumours | 515 | 448 |
| ADA Boost | BERT | Non-Rumours | 983 | 198 |
| | | Rumours | 177 | 379 |
| | 39 Features | Non-Rumours | 1001 | 296 |
| | | Rumours | 159 | 281 |
| K-Nearest Neighbor | BERT | Non-Rumours | 989 | 108 |
| | | Rumours | 171 | 469 |
| | 39 Features | Non-Rumours | 914 | 260 |
| | | Rumours | 246 | 317 |
| 4- layers MLP | BERT | Non-Rumours | 1016 | 125 |
| | | Rumours | 144 | 452 |
| | 39 Features | Non-Rumours | 972 | 237 |
| | | Rumours | 188 | 340 |

Table 4. Comparison results in rumour detection using BERT and feature engineering

| Model | Dataset | All Classes | | | | Non-Rumours | | | Rumours | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Acc | Prec | Rec | F1 | Prec | Rec | F1 | Prec | Rec | F1 |
| Support Vector | BERT | 81.9% | 79.50% | 79.7% | 79.6% | 86.6% | 86.2% | 86.4% | 72.5% | 73.0% | 72.8% |
| Machine | 39 Features | 73.7% | 71.00% | 65.8% | 68.3% | 75.7% | 89.2% | 81.9% | 66.2% | 42.5% | 51.7% |
| | Improved | 8.2% | 8.6% | 13.8% | 11.3% | 10.9% | -3.0% | 4.5% | 6.3% | 30.7% | 21.1% |
| Logistic Regression | BERT | 83.1% | 81.0% | 80.6% | 80.8% | 86.9% | 87.9% | 87.4% | 75.1% | 73.3% | 74.2% |
| | 39 Features | 73.8% | 71.2% | 65.9% | 68.4% | 75.7% | 89.4% | 82.0% | 66.6% | 42.5% | 51.9% |
| | Improved | 9.3% | 9.8% | 14.7% | 12.4% | 11.1% | -1.5% | 5.4% | 8.6% | 30.8% | 22.4% |
| Naive Bayes | BERT | 73.7% | 72.1% | 74.6% | 73.4% | 86.4% | 72.0% | 78.6% | 57.8% | 77.3% | 66.2% |
| | 39 Features | 62.9% | 64.9% | 66.6% | 65.8% | 83.3% | 55.6% | 66.7% | 46.5% | 77.6% | 58.2% |
| | Improved | 10.8% | 7.2% | 8.0% | 7.6% | 3.1% | 16.4% | 11.9% | 11.3% | -0.3% | 8.0% |
| ADA Boost | BERT | 78.4% | 75.7% | 75.2% | 75.5% | 83.2% | 84.7% | 84.0% | 68.2% | 65.7% | 66.9% |
| | 39 Features | 73.8% | 70.5% | 67.5% | 69.0% | 77.2% | 86.3% | 81.5% | 63.9% | 48.7% | 55.3% |
| | Improved | 4.6% | 5.2% | 7.7% | 6.5% | 6.1% | -1.6% | 2.5% | 4.3% | 17.0% | 11.6% |
| K-Nearest Neighbor | BERT | 83.9% | 81.7% | 83.3% | 82.5% | 90.2% | 85.3% | 87.6% | 73.3% | 81.3% | 77.1% |
| | 39 Features | 70.9% | 67.1% | 66.9% | 67.0% | 77.9% | 78.8% | 78.3% | 56.3% | 54.9% | 55.6% |
| | Improved | 13.1% | 14.6% | 16.4% | 15.5% | 12.3% | 6.5% | 9.3% | 17.0% | 26.3% | 21.5% |
| 4-Layers oMLP | BERT | 84.5% | 82.4% | 83.0% | 82.7% | 89.0% | 87.6% | 88.3% | 75.8% | 78.3% | 77.1% |
| | 39 Features | 75.5% | 72.4% | 71.4% | 71.9% | 80.4% | 83.8% | 82.1% | 64.4% | 58.9% | 61.5% |
| | Improved | 9.0% | 10.0% | 11.6% | 10.8% | 8.6% | 3.8% | 6.2% | 11.4% | 19.4% | 15.5% |

### 4.1. Comparison models

Using the PHEME dataset, previous researchers have employed several techniques to identify rumours on Twitter. These earlier works served as benchmarks against which we compared the results of our experiment. Table 5 compares our best model to the models from previous studies using the PHEME dataset. It demonstrates that our presented model outperforms existing classifier models and surpasses the current state of the art in regard to performance parameters.

Table 5. Comparison of our model to earlier studies on the PHEME dataset

| Previous works on PHEME dataset | Method | Best Result | | | |
|---|---|---|---|---|---|
| | | Accuracy (%) | Precision (%) | Recall (%) | F1 (%) |
| Zubiaga *et al*. [4] | Conditional random field (CRF) based on content and social features | NA | 66.7 | 55.6 | 60.7 |
| Hassan *et al*. [5] | Various supervised learning algorithms | 78.4 | 79.6 | 91.9 | 85.2 |
| Ajao *et al*. [20] | Combining CNN and LSTM models | 82.29 | 44.35 | NA | NA |
| Kotteti et al. [9] | using time series data to reduce time and supervised learning algorithms | NA | 94.9 | 35.6 | 51.8 |
| Alkhodair *et al*. [17] | Using word embedding and CNN | NA | 72.8-R, 83.3-NR | 70.6-R, 84.7-NR | 79.5-all class 71.6-R, 83.9-NR, |
| Bharti and Jindal [19] | CNN | NA | 79-R 87-NR | 76-R 89-NR | 77-R 88-NR |
| Xu *et al*. [21] | Topic-driven rumour detection (TDRD), by combining topic model and CNN | 82.66 | 81.33-R, 83.14-NR | 63.55-R, 92.49-NR | 71.20-R, 87.55-NR |
| Our model | By using BERT as a sentence embedding and 4-layers MLP as a classifier | 84.5 | 82.4-all 75.8-R 89.0-NR | 83.0– all 78.3– R 87.6- NR | 82.7– all 77.1– R 88.3- NR |

*all: all class, R: rumour, NR: non-rumour

## 5. CONCLUSION

According to the findings of our experiment, it was discovered that sentence embedding vector utilisation significantly enhances the performance of all classifier models by 10% compared to feature extraction vectors. Moreover, by employing BERT's embedding vectors and four layers of MLP, we achieve the most optimal model performance, surpassing baseline models with accuracy, precision, recall, and F1 scores of 84.5%, 82.4%, 83.0%, and 82.7%, respectively. Therefore, we confidently suggest that sentence embedding using BERT is a promising technique for identifying rumours, eliminating the need for traditional feature extraction steps.

## ACKNOWLEDGEMENTS

The authors would like to express their gratitude to Sebelas Maret University for providing significant financial support for this study under research scheme funding number 254/UN27.22/PT.01.03/2022.

## REFERENCES

[1] A. Bondielli and F. Marcelloni, "A survey on fake news and rumour detection techniques," *Information Sciences*, vol. 497, pp. 38–55, 2019, doi: 10.1016/j.ins.2019.05.035.

[2] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on Twitter," *Proceedings of the 20th International Conference Companion on World Wide Web, WWW 2011*, pp. 675–684, 2011, doi: 10.1145/1963405.1963500.

[3] J. Ito, H. Toda, Y. Koike, J. Song, and S. Oyama, "Assessment of tweet credibility with LDA features," *WWW 2015 Companion - Proceedings of the 24th International Conference on World Wide Web*, pp. 953–958, 2015, doi: 10.1145/2740908.2742569.

[4] A. Zubiaga, M. Liakata, and R. Procter, "Learning reporting dynamics during breaking news for rumour detection in social media," 2016.

[5] N. Y. Hassan, W. H. Gomaa, G. A. Khoriba, and M. H. Haggag, "Supervised learning approach for twitter credibility detection," Dec. 2018, doi: 10.1109/icces.2018.8639315.

[6] A. Ghenai and Y. Mejova, "Catching zika fever: Application of crowdsourcing and machine learning for tracking health misinformation on Twitter," *Proceedings - 2017 IEEE International Conference on Healthcare Informatics, ICHI 2017*, p. 518, 2017, doi: 10.1109/ICHI.2017.58.

[7] W. Herzallah, H. Faris, and O. Adwan, "Feature engineering for detecting spammers on Twitter: Modelling and analysis," *Journal of Information Science*, vol. 44, no. 2, pp. 230–247, Jan. 2017, doi: 10.1177/0165551516684296.

[8] K. Sato, J. Wang, and Z. Cheng, "Credibility evaluation of Twitter-based event detection by a mixing analysis of heterogeneous data," *IEEE Access*, vol. 7, pp. 1095–1106, 2019, doi: 10.1109/ACCESS.2018.2886312.

[9] C. M. M. Kotteti, X. Dong, and L. Qian, "Multiple time-series data analysis for rumor detection on social media," Dec. 2018, doi: 10.1109/bigdata.2018.8622631.

[10] J. Ma *et al.*, "Detecting rumors from microblogs with recurrent neural networks," *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2016-January, pp. 3818–3824, 2016.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," Oct. 2018.
[12] S. Sharma, M. Saraswat, and A. K. Dubey, "Fake news detection using deep learning," in *Knowledge Graphs and Semantic Web*, Springer International Publishing, 2021, pp. 249–259.
[13] A. Vaswani *et al.*, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 2017-December, pp. 5999–6009, 2017.
[14] S. Chatterjee, S. Deng, J. Liu, R. Shan, and W. Jiao, "Classifying facts and opinions in Twitter messages: a deep learning-based approach," *Journal of Business Analytics*, vol. 1, no. 1, pp. 29–39, 2018, doi: 10.1080/2573234X.2018.1506687.
[15] Y. Hu, K. Talamadupula, and S. Kambhampati, "Dude, srsly?: The surprisingly formal nature of Twitter's language," *Proceedings of the International {AAAI} Conference on Web and Social Media*, vol. 7, no. 1, pp. 244–253, Aug. 2021, doi: 10.1609/icwsm.v7i1.14443.
[16] N. Ruchansky, S. Seo, and Y. Liu, "CSI: A hybrid deep model for fake news detection," Nov. 2017, doi: 10.1145/3132847.3132877.
[17] S. A. Alkhodair, S. H. H. Ding, B. C. M. Fung, and J. Liu, "Detecting breaking news rumors of emerging topics in social media," *Information Processing and Management*, vol. 57, no. 2, 2020, doi: 10.1016/j.ipm.2019.02.016.
[18] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A convolutional approach for misinformation identification," *IJCAI International Joint Conference on Artificial Intelligence*, vol. 0, pp. 3901–3907, 2017, doi: 10.24963/ijcai.2017/545.
[19] M. Bharti and H. Jindal, "Automatic rumour detection model on social media," *PDGC 2020 - 2020 6th International Conference on Parallel, Distributed and Grid Computing*, pp. 367–371, 2020, doi: 10.1109/PDGC50313.2020.9315738.
[20] O. Ajao, D. Bhowmik, and S. Zargari, "Fake news identification on Twitter with hybrid CNN and RNN models," *ACM International Conference Proceeding Series*, pp. 226–230, 2018, doi: 10.1145/3217804.3217917.
[21] F. Xu, V. S. Sheng, and M. Wang, "Near real-time topic-driven rumor detection in source microblogs," *Knowledge-Based Systems*, vol. 207, 2020, doi: 10.1016/j.knosys.2020.106391.
[22] H. Taud and J. F. Mas, "Multilayer perceptron (MLP)," in *Geomatic Approaches for Modeling Land Change Scenarios*, Springer International Publishing, 2017, pp. 451–455.
[23] A. I. Kadhim, "Survey on supervised machine learning techniques for automatic text classification," *Artificial Intelligence Review*, vol. 52, no. 1, pp. 273–292, 2019, doi: 10.1007/s10462-018-09677-1.
[24] J. Hao and T. K. Ho, "Machine learning made easy: A review of scikit-learn package in python programming language," *Journal of Educational and Behavioral Statistics*, vol. 44, no. 3, pp. 348–361, Feb. 2019, doi: 10.3102/1076998619832248.
[25] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, 2019.


## BIOGRAPHIES OF AUTHORS

**Rini Anggrainingsih** She received her bachelor's degree from Diponegoro University and received her master's degree from Gadjahmada University in Indonesia. She is currently working as an academic staff at the Informatics Department of Sebelas Maret University and has been pursuing her Ph.D. at The University of Western Australia. Her current research interest areas are Business Process Analysis and NLP, particularly about twitter data credibility analysis to improve info-surveillance on social media. She can be contacted at email: rini.anggrainingsih@staff.uns.ac.id.

**Endar Suprih Wihidayat** He received his bachelor's degree from Diponegoro University and received his master's degree from Gadjahmada University in Indonesia. He is currently working as an academic staff at the Informatics Education Department of Sebelas Maret University and has been pursuing his Ph.D. at Curtin University Australia. His current research areas are Machine learning, Game-based learning, and the Internet of things. He can be contacted at email: Endars@staff.uns.ac.id.

**Bambang Widoyono** He received his bachelor of informatics engineering from Telkom Institute Technology and received his master's degree from the University of Indonesia. He is currently working as a senior consultant in a Japanese IT Company and as a lecturer assistant at the University of Indonesia. His research areas of interest include data analytics, social media analytics, data management, software development, and project management. He can be contacted at email: bambang.widoyono@ui.ac.id or bambangwidoyono@gmail.com.