

News classification using light gradient boosted machine algorithm

Muhammad Hatta Rahmatul Kholiq, Wiranto, Sari Widya Sihwi

Department of Informatics, Faculty of Mathematics and Natural Sciences, Universitas Sebelas Maret, Surakarta, Indonesia

Article Info

Article history:

Received Jul 31, 2021

Revised May 11, 2022

Accepted May 30, 2022

Keywords:

Count vectorizer

Fake news detection

LightGBM

Machine learning

Tfidf vectorizer

ABSTRACT

News classification is a complex issue as people are easily convinced of misleading information and lack control over the spread of fake news. However, we can break the problem of spreading fake news with artificial intelligence (AI), which has developed rapidly. This study proposes a news classification model using a light gradient boosted machine (LightGBM) algorithm. The model is analyzed using two feature extraction techniques, count vectorizer and Tfidf vectorizer and compared with a deep learning model using long-short term memory (LSTM). The experimental evaluation showed that all LightGBM models outperform LSTM. The best model is the count vectorizer LightGBM, which achieves an accuracy value of 0.9933 and an area under curve (AUC) score of 0.9999.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Wiranto

Department of Informatics, Faculty of Mathematics and Natural Sciences, Universitas Sebelas Maret

Street of Ir. Sutami No. 36A, Kentingan, Jebres, Surakarta, Central Java 57126, Indonesia

Email: wiranto@staff.uns.ac.id

1. INTRODUCTION

Humans today live in a society where everyone is connected via the internet. Through news portals that spread on the internet, humans can effortlessly obtain, process, and share information. Humans are bombarded with information all the time without knowing the truth. News classification is a difficult task as humans tend to think of misleading information and lack control over the spread of fake news [1]. Fake news is also seen as a significant threat to democracy, journalism, and the economy [2]. The extent of the reach of fake news is quite challenging to deal with today because of the large number of information-sharing platforms that can spread fake news or misinformation. The spread of fake news is becoming a bigger problem due to advances in artificial intelligence (AI), causing bots to create and spread fake news quickly [3].

The issue of spreading fake news can have a severe negative impact on people and civilization. First, fake news can upset the balance of authenticity of the news ecosystem. Second, fake news deliberately entices buyers to accept biased or false assumptions. Third, fake news changes the way people understand and react to real news [4]. Fifth, it can bring negative consequences to the market in generating profit [5], [6].

However, fake news detection is a complicated task that ordinary people can hardly detect without any extra information instead of the news contents [7]. Therefore, fake news detection has become a quite active study in natural language processing (NLP), and several studies have explored algorithms for solving inaccurate fake detection [8]. In addition, there are also many various rumour datasets in English that have been provided, including information security and object technology (ISOT) dataset.

To get a better result in detecting fake news, some researchers propose some new models or compare several existing algorithms such as [9], [10]. Agarwal *et al.* [9], a comparative study of classification algorithms, shows the support vector machine (SVM) as the best algorithm, with an F1 score

value of 0.61. Other studies also use several classification algorithms and two feature extraction techniques, count vectorizer and Tfidf vectorizer, resulting in an accuracy value of 0.928 using the SVM algorithm and Tfidf extraction technique [11]. However, research [10] shows LSTM outperformed SVM, Naïve Bayes (NB) and neural network in terms of accuracy, precision, recall and F1-Score. LSTM achieved an average accuracy of 94.21%. Similarly, Rohera *et al.* [12] also shows LSTM outperform SVM, random forests (RF), and SVM in classifying fake news. On the other side, [13] found that XGBoost (XGB) and RF performed best in detecting fake news among k-nearest neighbors (KNN), NB, RF, SVM with RBF kernel (SVM), and XGB.

One of the widely used datasets in fake detection research is called information security and object technology (ISOT) dataset whose each instance is longer than 200 characters. Ahmed *et al.* [14], [15] collected it from real-world sources consisting of news articles from Reuters.com and Kaggle.com. Ahmed *et al.* [14] detect fake news using several classification algorithms and give the n-gram effect to their study with the best algorithm is support vector machine, with an accuracy value of 92%. Baarir and Djeflal [16] also use the ISOT dataset in their research on fake news detection. The evaluation metric used in this research is the F1 score. This study concludes that the fake news detection model using the linear regression algorithm is the best, with an F1 score of 96.51%. Nasir *et al.* [17] also conducted the same study with the ISOT dataset. He conducted experiments using CNN, RNN, and Hybrid CNN-RNN. The accuracy obtained using RNN is 0.98, while for CNN and Hybrid CNN-RNN, it is 0.99. A different study that also uses the ISOT dataset is the Kula study [18]. His study discusses the detection of fake news using deep learning with the LSTM algorithm resulting in an accuracy of 99.86%.

Since many researchers have achieved high accuracy by using LSTM and ensemble models [7], this study will use light gradient boosting machine (LightGBM), a state of the art classifier from the decision tree boosting algorithms family, and also compare it with LSTM. The ensemble method has proven to be very effective and versatile over a wide range of problems as it was initially developed to reduce variance and thus increase accuracy [19]. LightGBM is the new version of gradient boosting decision tree (GBDT), a widely-used ensemble machine learning algorithm, due to its efficiency, accuracy, and interpretability [20]. In most scenarios, its prediction accuracy is better than other machine learning algorithms [21]. LightGBM aims to enhance the model's efficiency and decrease memory usage [20]. The dataset used is ISOT dataset. We compare LightGBM's performance with LSTM's performance since LSTM effectively improve performance by memorizing and finding the pattern of crucial information [22]. Working with ISOT dataset, [23] shows LSTM has a better result compared to NB, SVM and feed forward neural network (FFNN). The experiment will use two feature extraction techniques, Count Vectorizer and Tfidf Vectorizer, to know which one gives the best result to the model.

2. LIGHTGBM

Proposed in 2017, LightGBM is an adaptive gradient boosting model, an efficient implementation form of gradient boosting trees, that improves the algorithm's computing power and prediction accuracy by using the histogram algorithm and other algorithms [21]. LightGBM has two sampling techniques, and we use the gradient-based one side sampling (GOSS) in this study to build fake news model detection. GOSS stores all instances with large gradients and performs random sampling of instances with slight gradients. To compensate for the effect on data distribution, when calculating information gain, GOSS introduces a constant multiplier for small gradient instances. GOSS sorts the data instances according to the absolute value of their gradient. It selects the top $a \times 100\%$ instance, where a is the large gradient instance. Then randomly takes a sample of $b \times 100\%$ from the other data, where b is a slight gradient instance. Furthermore, GOSS strengthens the sample data with a slight gradient using the constant $(1-a)/b$, so that the calculation of information gain can be written by the equation [20].

$$S_A = \frac{(\sum_{X_i \in A_l} g_i + \frac{1-a}{b} \sum_{X_i \in B_l} g_i)^2}{n_l^j(d)} \quad (1)$$

$$S_B = \frac{(\sum_{X_i \in A_r} g_i + \frac{1-a}{b} \sum_{X_i \in B_r} g_i)^2}{n_r^j(d)} \quad (2)$$

$$V_j(d) = \frac{1}{n} (S_A + S_B) \quad (3)$$

where,

$$A_l = \{x_i \in A: x_{ij} \leq d\}$$

$$A_r = \{x_i \in A: x_{ij} > d\}$$

$$B_l = \{x_i \in B: x_{ij} \leq d\}$$

$$B_r = \{x_i \in B: x_{ij} > d\}$$

3. RESEARCH METHOD

3.1. Datasets

This study uses a fake news dataset from a research lab of the University of Victoria called information security and object technology (ISOT). The real news is collected from Reuters, while the fake news is a collection from various websites marked by PolitiFact. The dataset by Ahmad *et al.* [14], [15] used contains 23,481 fake news and 21,417 real news. By adopting research done by Ahmad *et al.* [14], [15] divided the dataset into 1,000; 5,000; 10,000; and 50,000. We also experiment with fractions of the dataset. We do the fractions from 2,000 and their multiples up to 26,000. Therefore, there are 13 fraction sub-datasets.

3.2. Research design

The design of the research can be seen in Figure 1. This research uses two classification algorithms, which are LightGBM and LSTM. Since we aim to know the best feature extraction method combined with LGBM, then the research will conduct count vectorizer and also TfIdf Vectorizer. Classifying with LSTM will be done after feature extraction using Integer Encoding. Since there are 13 fraction sub-datasets, then there are 13 trials for one model, which are from 2000 and its multiples up to 26000.

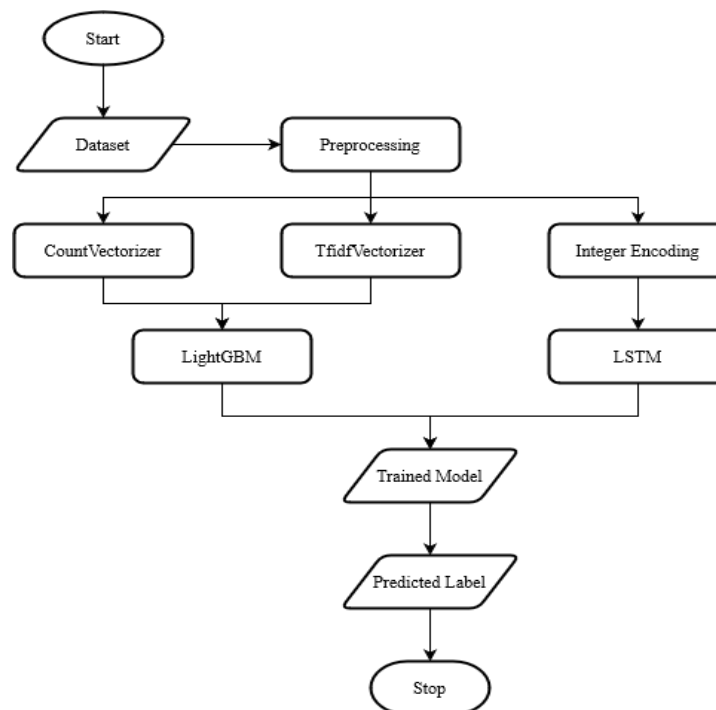


Figure 1. Proposed model

3.3. Preprocessing

As seen in Figure 1, this research begins with preprocessing. The preprocessing stage contains data cleansing, case folding, stopwords removal, tokenization, and lemmatization. This stage uses Pandas to manage datasets and natural language toolkit (NLTK) to solve NLP problems such as stopwords removal, tokenization, and lemmatization.

3.4. Feature extraction

The next step after preprocessing is feature extraction from the data. LightGBM classifier uses two feature extraction techniques, namely the count vectorizer and TfIdf vectorizer, and LSTM uses the Integer Encoding technique. Count vectorizer is an implementation of bag-of-words content analysis that takes words or terms as a set (or bag) [24]. Term frequency-inverse document frequency (TF-IDF) is a numerical statistic intended to represent the importance of a word in a document in one corpus. The TF-IDF formulation used in this study comes from sklearn. Integer encoding is a feature extraction technique that represents each word in

a document as an integer value and builds a representation matrix. The integer encoding stage in the feature extraction process is given the same treatment as the feature extraction technique used in LightGBM.

3.5. Classification

The classification process consists of training and validation. This study uses two classification processes as written with the goal of finding the best results between the two models. The first classification uses the LightGBM algorithm, which begins with finding the best parameters for the LightGBM algorithm using the GridSearchCV library as well as doing cross-validation. The cross-validation technique used is k-fold cross-validation with a value of $k=10$. There are no formal rules for selecting the value, but when it gets bigger, the size difference between the training set and the resampling subset gets smaller. When this difference decreases, the bias value becomes smaller [25].

The second classification uses the LSTM algorithm, which begins with determining the layer for the deep learning model, which is then followed by the training and validation processes. The classification step uses the sci-kit-learn library and LightGBM. The two libraries are used to carry out the training and testing process for the LightGBM model. Meanwhile, deep learning models use the Keras library.

3.6. Evaluation

The evaluation step of each model begins with predicting the testing data using the trained model. The prediction results are then processed to be used as values for several metrics to measure model performance. Some of these metrics are accuracy, f1-score, ROC/AUC, and logistic loss. The best model is the model with the highest accuracy value, f1-score, and AUC scores, while the loss value is the lowest.

4. RESULTS AND DISCUSSION

This work uses an ISOT dataset [14], [15], which is divided evenly for the REAL and FAKE classes. The number of datasets is divided into multiples of one thousand per class (or 2,000 as the sum of both classes) for each number of datasets so that a new dataset is generated with a total of 2,000 datasets up to 26,000 datasets. All of the datasets are processed and discussed in this chapter as follows: preprocessing, feature extraction, classification and evaluation.

4.1. Preprocessing

The first step in preprocessing is data cleansing, which starts by removing null value data. The dataset was randomly selected as many as 13,000 articles for each real and fake class. After the dataset was selected, labelling FAKE and REAL was carried out, followed by attribute selection. This study uses the title, and text attributes combined into one where the title is the news title while the text is the news content, and the label has been added. The original text can be seen in Figure 2.

Original text

Campaigning began on Tuesday in an election that pits Prime Minister Shinzo Abe's Liberal Democratic Party against the fledgling Party of Hope led by Tokyo Governor Yuriko Koike and other smaller parties contesting seats in Japan's more powerful lower house of parliament.

Figure 2. Original text

The next step is case folding, as can be seen in Figure 3. Case folding is done to equate all text data in one lowercase form so that there are no longer the exact words with different letter cases. This will speed up computation by eliminating terms that occur due to the same word in a different letter case. As can be seen in Figure 4, the next step of this research is stopwords removal, which aims to eliminate words that have no significant impact, such as conjunctions, pronouns, question words, and punctuation. The following step, as can be seen in Figure 5, is tokenization accompanied by lemmatization. Tokenization serves to break a sentence into a list of words. At the same time, lemmatization changes the word that has been given an affix into its original form.

Case-folded text

campaigning began on tuesday in an election that pits prime minister shinzo abe's liberal democratic party against the fledgling party of hope led by tokyo governor yuriko koike and other smaller parties contesting seats in japan's more powerful lower house of parliament.

Figure 3. Case folded text

Stopwords removed from text

campaigning began tuesday election pits prime minister shinzo abe liberal democratic party fledgling party hope led tokyo governor yuriko koike smaller parties contesting seats japan powerful lower house parliament

Figure 4. Stopwords removal

Tokenized and lemmatized text

['campaigning', 'began', 'tuesday', 'election', 'pits', 'prime', 'minister', 'shinzo', 'abe', 'liberal', 'democratic', 'party', 'fledgling', 'party', 'hope', 'led', 'tokyo', 'governor', 'yuriko', 'koike', 'smaller', 'parties', 'contesting', 'seats', 'japan', 'powerful', 'lower', 'house', 'parliament']

Figure 5. Tokenization and lemmatization

4.2. Feature extraction

At this step, the data is divided into training and testing data with a ratio of 80/20. Text data is then represented in a matrix using the feature extraction technique—first, a feature extraction experiment using count vectorizer with the LightGBM algorithm. A total of 1,600 training data were obtained from the distribution previously described. There are 29,823 features that have been successfully extracted or called vocabulary, so the training data representation matrix's size is (1,600; 29,823). Each term in the vocabulary is represented as an integer number. The following feature extraction technique is Tfidf vectorizer. The features successfully extracted using the Tfidf vectorizer were 29,980 out of 1,600 training data. The feature representation matrix's size with this technique is (1,600; 29,980). As in the count vectorizer, the Tfidf term is also represented as an integer value. Term frequency-inverse document frequency (TF-IDF) is a numerical statistic intended to represent the importance of a word in a document in one corpus (Rajaraman & Ullman, 2011). The TF-IDF formulation used in this study comes from sklearn.

4.3. Classification

The classification is divided into three, LightGBM with count vectorizer, LightGBM with Tfidf Vectorizer and LSTM integer encoding. The classification step begins with hyperparameter tuning using GridSearchCV. The hyperparameter values are obtained from adjustments to the hardware used because the memory used cannot accommodate all datasets. After the tuning is done, each parameter's value is obtained from the best results. In the research, we set some hyperparameters with values 800 for *the n_iterations*, 0.1 for *the learning_rate*, and 2 for *the min_data_in_leaf*. The step after hyperparameter tuning is training. The trained model then predicts the news label fake or real.

4.4. Evaluation

The evaluation step begins after predicting the test data has been carried out. The model is evaluated using a confusion matrix that can later be used to calculate other metrics such as accuracy, precision, recall, F1 score, ROC/AUC. Besides that, it is also evaluated with loss values. In this study, there are two models where each one is trained three times to verify the models. The first model is a model that uses the LightGBM algorithm with the Count Vectorizer feature extraction technique. Its result can be seen in Table 1. The second model is a model with the LightGBM algorithm and the Tfidf Vectorizer extraction technique. Its result can be seen in Table 2. The third model in Table 3 is an LSTM algorithm with Integer. Through the three models above, it can be seen that the first model and the second model are optimal when the dataset is 20000 because the evaluation metric is at its highest value, while the third model is optimal in the dataset with 26000 data. In summary, the comparison of the three models can be seen in Table 4. As can be seen, LSTM has nearly perfect scores for precision, recall, F1 and AUC. However, LightGBM with both feature extraction techniques outperform its scores and also with lower loss score. LightGBM is more optimal when using CountVectorizer as its feature extraction technique.

Table 1. Result of LightGBM CountVectorizer

Dataset	Accuracy	Precision	Recall	F1	AUC	Loss
2.000	0,9825	0,9856	0,9810	0,9833	0,9991	0,0436
4.000	0,9825	0,9870	0,9768	0,9819	0,9988	0,0647
6.000	0,9900	0,9949	0,9848	0,9898	0,9992	0,0365
8.000	0,9869	0,9888	0,9851	0,9869	0,9989	0,0634
10.000	0,9915	0,9948	0,9876	0,9912	0,9997	0,0278
12.000	0,9900	0,9917	0,9884	0,9900	0,9995	0,0337
14.000	0,9932	0,9957	0,9908	0,9932	0,9998	0,0236
16.000	0,9919	0,9969	0,9870	0,9919	0,9996	0,0374
18.000	0,9911	0,9912	0,9912	0,9912	0,9994	0,0391
20.000	0,9933	0,9950	0,9916	0,9933	0,9999	0,0188
22.000	0,9955	0,9959	0,9950	0,9954	0,9996	0,0290
24.000	0,9942	0,9967	0,9917	0,9942	0,9999	0,0218
26.000	0,9938	0,9949	0,9926	0,9938	0,9998	0,0269

Table 2. Result of LightGBM TfidfVectorizer

Dataset	Accuracy	Precision	Recall	F1	AUC	Loss
2.000	0,9675	0,9660	0,9707	0,9684	0,9965	0,1120
4.000	0,9838	0,9869	0,9792	0,9831	0,9986	0,0664
6.000	0,9842	0,9851	0,9835	0,9843	0,9993	0,0484
8.000	0,9856	0,9925	0,9790	0,9857	0,9989	0,0620
10.000	0,9860	0,9918	0,9798	0,9858	0,9991	0,0580
12.000	0,9908	0,9924	0,9890	0,9907	0,9997	0,0290
14.000	0,9946	0,9949	0,9942	0,9946	0,9997	0,0221
16.000	0,9897	0,9963	0,9834	0,9898	0,9992	0,0530
18.000	0,9929	0,9960	0,9915	0,9938	0,9998	0,0227
20.000	0,9903	0,9950	0,9857	0,9903	0,9996	0,0408
22.000	0,9952	0,9959	0,9946	0,9952	0,9997	0,0250
24.000	0,9935	0,9942	0,9929	0,9935	0,9998	0,0267
26.000	0,9938	0,9946	0,9930	0,9938	0,9997	0,0282

Table 3. Result of LSTM IntegerEncoding

Dataset	Accuracy	Precision	Recall	F1	AUC	Loss
2.000	0,9346	0,9200	0,9484	0,9338	0,9822	0,1849
4.000	0,9629	0,9649	0,9603	0,9625	0,9901	0,1197
6.000	0,9669	0,9610	0,9733	0,9671	0,9939	0,1047
8.000	0,9694	0,9647	0,9740	0,9692	0,9951	0,0982
10.000	0,9767	0,9851	0,9680	0,9765	0,9960	0,0890
12.000	0,9747	0,9784	0,9706	0,9744	0,9951	0,0831
14.000	0,9775	0,9855	0,9693	0,9773	0,9971	0,0729
16.000	0,9756	0,9810	0,9700	0,9755	0,9956	0,0816
18.000	0,9799	0,9789	0,9814	0,9801	0,9969	0,0711
20.000	0,9788	0,9838	0,9735	0,9786	0,9967	0,0695
22.000	0,9745	0,9765	0,9720	0,9742	0,9962	0,0854
24.000	0,9809	0,9850	0,9771	0,9810	0,9965	0,0664
26.000	0,9847	0,9875	0,9822	0,9848	0,9977	0,0542

Table 4. Model comparison

	Accuracy	Precision	Recall	F1	AUC	Loss
LightGBM CountVectorizer	0,9933	0,9950	0,9916	0,9933	0,9999	0,0188
LightGBM TfidfVectorizer	0,9903	0,9950	0,9857	0,9903	0,9996	0,0408
LSTM IntegerEncoding	0,9788	0,9838	0,9735	0,9786	0,9967	0,0695

5. CONCLUSION

This study succeeded in building a news classification model using the LightGBM algorithm. LightGBM is a mighty classification algorithm. This can be seen by applying two feature extraction techniques, count vectorizer and Tfidf vectorizer, which give relatively the same results. LightGBM algorithm and n-gram analysis alone return almost perfect results compared to other studies that use more features or use deep learning with fairly complex architecture.

This research has shortcomings when it comes to data collection and feature extraction. The author suggests solving the problem of data imbalance to process wild data. As much as possible, the classification model should use balanced data so that the classification results are not biased. This research has proven that the LightGBM algorithm works well to solve news classification problems with large datasets.

ACKNOWLEDGEMENTS

Funding was supported by Group Research Grant from non-APBN fund Universitas Sebelas Maret 2021 No: 260/UN27.22/HK.07.00/2021 for intelligent systems and humanized computing (ISHC) Research Group.




REFERENCES

- [1] N. Lemann, "Solving the problem of fake news," *The New Yorkers*, New York, NY, USA, Nov. 30, 2016.
- [2] X. Zhou and R. Zafarani, "Fake news: A survey of research, detection methods, and opportunities," *arXiv arXiv:1812.00315*, 2018.
- [3] C. Shao, G. L. Ciampaglia, O. Varol, K.-C. Yang, A. Flammini, and F. Menczer, "The spread of low-credibility content by social bots," *Nat. Commun.*, vol. 9, no. 1, 2018, doi: 10.1038/s41467-018-06930-7.
- [4] M. P. Lynch, "Fake news," *The New York Times*, Nov. 2016.
- [5] K. Rapoza, "Can 'fake news' impact the stock market?," *Forbes*, 2017.
- [6] I. Ahmad, M. Yousaf, S. Yousaf, and M. O. Ahmad, "Fake News Detection Using Machine Learning Ensemble Methods," *Complexity*, vol. 2020, p. 8885861, 2020, doi: 10.1155/2020/8885861.
- [7] M. Samadi, M. Mousavian, and S. Momtazi, "Deep contextualized text representation and learning for fake news detection," *Inf. Process. Manag.*, vol. 58, no. 6, p. 102723, 2021, doi: 10.1016/j.ipm.2021.102723.
- [8] A. Pardamean and H. F. Pardede, "Tuned bidirectional encoder representations from transformers for fake news detection," *Indonesian J. Electr. Comput. Sci.*, vol. 22, no. 3, p. 1667, 2021, doi: 10.11591/ijeecs.v22.i3.pp1667-1671.
- [9] V. Agarwal, H. P. Sultana, S. Malhotra, and A. Sarkar, "Analysis of Classifiers for Fake News Detection," *Procedia Comput. Sci.*, vol. 165, pp. 377–383, 2019, doi: 10.1016/j.procs.2020.01.035.
- [10] S. A. Alameri and M. Mohd, "Comparison of Fake News Detection using Machine Learning and Deep Learning Techniques," in *2021 3rd International Cyber Resilience Conference (CRC)*, 2021, pp. 1–6, doi: 10.1109/CRC50527.2021.9392458.
- [11] K. Poddar, G. B. Amali D., and K. S. Umadevi, "Comparison of Various Machine Learning Models for Accurate Detection of Fake News," in *2019 Innovations in Power and Advanced Computing Technologies (i-PACT)*, 2019, vol. 1, pp. 1–5, doi: 10.1109/i-PACT44901.2019.8960044.
- [12] D. Rohera et al., "A Taxonomy of Fake News Classification Techniques: Survey and Implementation Aspects," *IEEE Access*, vol. 10, pp. 30367–30394, 2022, doi: 10.1109/ACCESS.2022.3159651.
- [13] J. C. S. Reis, A. Correia, F. Murai, A. Veloso, F. Benevenuto, and E. Cambria, "Supervised Learning for Fake News Detection," *IEEE Intell. Syst.*, vol. 34, no. 2, pp. 76–81, 2019, doi: 10.1109/mis.2019.2899143.
- [14] H. Ahmed, I. Traore, and S. Saad, "Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques," in *Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, 2017, doi: 10.1007/978-3-319-69155-8_9.
- [15] H. Ahmed, I. Traore, and S. Saad, "Detecting opinion spams and fake news using text classification," *Secur. Priv.*, vol. 1, no. 1, p. e9, 2017, doi: 10.1002/spy2.9.
- [16] N. F. Baarir and A. Djeflal, "Fake News detection Using Machine Learning," in *2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH)*, 2021, pp. 125–130, doi: 10.1109/IHSH51661.2021.9378748.
- [17] J. A. Nasir, O. S. Khan, and I. Varlamis, "Fake news detection: A hybrid CNN-RNN based deep learning approach," *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 1, p. 100007, 2021, doi: 10.1016/j.jjime.2020.100007.
- [18] S. Kula and R. K. P. W. M. Choraś Michaland Kozik, "Sentiment analysis for fake news detection by means of neural networks," in *International Conference on Computational Science*, 2020, pp. 653–666, doi: 10.1007/978-3-030-50423-6_49.
- [19] R. Polikar, "Ensemble Learning," in *Ensemble Machine Learning: Methods and Applications*, 2012th ed., Springer, 2012, pp. 1–34, doi: 10.1007/978-1-4419-9326-7_1.
- [20] G. Ke et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 3149–3157.
- [21] D. Wang, L. Li, and D. Zhao, "Corporate finance risk prediction based on LightGBM," *Inf. Sci. (Ny)*, vol. 602, pp. 259–268, 2022, doi: 10.1016/j.ins.2022.04.058.
- [22] N. Rai, D. Kumar, N. Kaushik, C. Raj, and A. Ali, "Fake News Classification using transformer based enhanced LSTM and BERT," *Int. J. Cogn. Comput. Eng.*, vol. 3, no. October 2021, pp. 98–105, 2022, doi: 10.1016/j.ijce.2022.03.003.
- [23] M. M. M. Hlaing and N. S. M. Kham, "Comparative study of fake news detection using machine learning and neural network approaches," *2021 11th Int. Work. Comput. Sci. Eng. WCSE 2021*, no. Wcse, pp. 455–460, 2021, doi: 10.18178/wcse.2021.02.010.




- [24] M. McTear, Z. Callejas, and D. Griol, *The Conversational Interface: Talking to Smart Devices*: Springer International Publishing, 2016, doi: 10.1007/978-3-319-32967-3.
- [25] M. Kuhn, K. Johnson, and others, *Applied predictive modeling*, vol. 26. Springer, 2013, doi: 10.1007/978-1-4614-6849-3.

BIOGRAPHIES OF AUTHORS






Muhammad Hatta Rahmatul Kholiq    received the B.Sc. degree in computer science from the Universitas Sebelas Maret Surakarta, Indonesia. He is currently a Software Engineer at PT. Insera Sena. His research interests include soft computing, machine learning, and intelligent systems. His interest came because of his fondness for the python programming language. He can be contacted at email: developerkecil@gmail.com.



Wiranto    currently works as an associate professor at the Department of Informatics, Universitas Sebelas Maret. He actively works as a researcher in Intelligent Systems and Humanized Computing research group. He holds a bachelor's degree from Universitas Gajah Mada. He took his master's degree in Computer Science from Universitas Indonesia and Universitas Gajah Mada. He finished his doctoral degree from Universitas Gajah Mada. He can be contacted at email: wiranto@staff.uns.ac.id.



Sari Widya Sihwi    holds bachelor's and master's degrees from the Faculty of Computer Science, Universitas Indonesia. Currently, she works as a lecturer at the Department of Informatics, Universitas Sebelas Maret, a public university in Indonesia. She has been interested in the topic of Artificial Intelligence (AI) since she was an undergraduate student. Currently, she has published several papers on the topic of AI, including natural language processing (NLP) and machine learning. She can be contacted at email: sariwidya@staff.uns.ac.id.