Jurnal Teknik Informatika Vol. 19 No. 1, April 2026 (211-220)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.50385

JURNAL TEKNIK INFORMATIKA

Homepage : http://journal.uinjkt.ac.id/index.php/ti

Detecting Hoax News in Indonesian Language Using Web-Based Multinomial Naïve Bayes

Fitri Mintarsih[1], Ivan Ananda Putra[2], Arini[3], Victor Amrizal[4], Hendra Bayu Suseno[5*]

[1,2,3,4,5] Department of Informatics Faculty of Science and Technology, Syarif Hidayatullah State Islamic University Jakarta
[1,2,3,4,5]Jl. Ir. H. Juanda No. 95, Ciputat, 15412

Article:
Accepted: January 29, 2026
Revised: November 28, 2025
Issued: April 30, 2026

***Correspondence Address:**
hendra.bayu@uinjkt.ac.id

ABSTRACT

This study addresses the growing problem of hoax news in Indonesia, which has contributed to social conflicts. It aims to develop an effective detection method using the Multinomial Naive Bayes algorithm. The study integrates Indonesian specific text preprocessing and feature engineering within the CRISP-DM framework to enhance classification performance. A dataset of 5,226 news articles (2,612 non-hoax and 2,614 hoax) was collected from kompas.com and turnbackhoax.id. Preprocessing steps included case folding, tokenization, stopword removal, and stemming tailored to the Indonesian language. Feature extraction was performed using the TF-IDF weighting scheme to convert text into numerical representations. The Multinomial Naive Bayes algorithm achieved an average accuracy of 86%, precision of 86%, recall of 86%, and F1 score of 86%, indicating stable and balanced performance. Furthermore, the trained model was successfully deployed using the Flask framework and stored in (pickle/joblib) format, demonstrating its practical applicability in real world systems. The results indicate that the integration of Indonesian specific preprocessing and TF-IDF feature representation significantly supports the effectiveness of the Multinomial Naive Bayes algorithm in detecting hoax news. This study provides a scalable and implementable approach to combating the spread of false information in Indonesian digital media.

Keywords : *text classification; hoax news detection; multinomial naïve bayes; flask framework.*

Jurnal Teknik Informatika Vol. 19 No. 1, April 2026 (211-220)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.50385

1. INTRODUCTION

Fake news or hoaxes can have a serious impact on society. This is because incorrect information about health, safety, political issues and others can cause distrust of the media, cause conflict, and even threaten public safety. The digital era and the development of social media allow fake news to spread quickly and widely. However, the rapid spread of news does not guarantee that the information and news we receive comes from valid sources. The ease of disseminating information through digital platforms increases the potential for misinformation to spread throughout the world in a short time. Fake news or hoaxes are information that is deliberately created or distributed with the aim of misleading, deceiving, or manipulating public perception [1].

Various studies have shown that the Multinomial Naïve Bayes (MNB) algorithm is effective in detecting hoax news and other classifications with a high level of accuracy. [2] achieved 94% accuracy in detecting hoax news with 250 articles. [3] also achieved 94.29% accuracy in classifying news from five leading online media using MNB and TF-IDF. Sandifer et al. (2018) found that MNB had 85.9% accuracy in analyzing hoax hotel reviews, while [4] reported 96% accuracy in bacteriophage virion protein classification, higher than the usual Naïve Bayes which only reached 75%.

From the results of this study, it is proven that MNB consistently provides high accuracy results in various domains, so it was chosen for this study. MNB is considered appropriate because of its assumption that each feature used is multinomially distributed, which is in accordance with the characteristics of the data studied. This study overcomes the shortcomings of previous studies by increasing the amount of data and adopting a new approach. Using the Multinomial Naive Bayes method, the model is saved in pickle format (.pkl) and deployed using the Flask framework from Python. These steps are expected to improve the accuracy of fake news detection and help mitigate the spread of false information in the digital environment.

In this study, the dataset for detecting hoax news in Indonesia was taken from two sources: Kompas.com for valid news and TurnBackHoax.id for hoax news. Kompas.com was chosen because of its credibility as one of the largest and most trusted news media in Indonesia, which always goes through a strict verification process. According to [5]. Meanwhile, TurnBackHoax.id, which is managed by the Indonesian Anti-Slander Society (MAFINDO), focuses on verifying and clarifying hoax news.

This study also divides the dataset into two parts, namely training data and test data with a ratio of 70:30. This ratio was chosen because it provides a good balance between training and testing the model. According to previous research, a 70:30 ratio ensures that the model has enough data to learn, while providing enough data to accurately evaluate model performance on previously unseen data [6].

In addition, this study uses TF-IDF for feature extraction due to its ability to capture the essence of significant words and reduce the influence of less informative common words. By using TF-IDF, the text classification model can focus more on relevant and unique words in the document, thereby improving the accuracy and performance of the model [7]. After the model is saved in formal pickle (.pkl), the model is then deployed using the flask framework, this framework is used because Flask is for a very simple and flexible micro web.

Jurnal Teknik Informatika Vol. 19 No. 1, April 2026 (211-220)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.50385

2. METHODS

This study uses CRISP-DM (Cross-Industry Standard Process for Data Mining) which consists of Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, Deployment. For more details, it is illustrated in Fig 1.

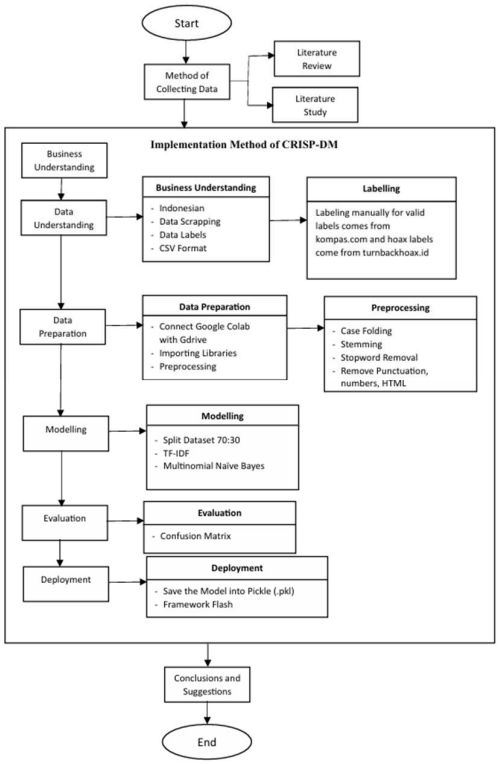

Figure 1. *Research framework*

2.1. Business Understanding

This research is based on the increasing spread of hoax news and the increasing ease of people getting the news. This phenomenon requires serious attention because it can have a negative impact on society. Therefore, the author emphasizes the importance of developing a classification model that is able to distinguish between hoax news and valid news. This model is expected to be an effective tool in reducing the negative impact of the spread of hoax news.

2.2. Data Understanding

At this stage, the researcher determines the data needs used as test data and also testing data from the model explained above. The needs of this dataset can be described as follows:

a. Indonesian
The dataset used to test and evaluate the hoax classification model must be in the form of Indonesian language text.

b. News data scraping
Data retrieval through scraping techniques is used to collect news information available in Indonesia. In this context, Kompas.com, as one of the trusted news sources, To identify hoax news, researchers collect data from turnbackhoax.id.

c. Hoax and Non-Hoax Labels
The dataset must have a clear label for each news item, indicating whether the news item is a hoax (1) or non-hoax (0). This label is important for training the model to identify patterns and characteristics that distinguish hoax and non-hoax news items.

d. CSV format
The dataset should be stored in CSV (Comma Separated Values) format to facilitate reading and processing of data by the model. CSV format is a standard format for tabular data and is easily accessible by various programs and applications. By using this format, data can be imported into various data analysis platforms such as Python, making it easier to analyze and interpret data. CSV format also allows for high interoperability and ensures that data remains well-structured during the transfer and processing process.

2.3. Data Preparation

Before being used to train and evaluate hoax classification models, Indonesian text data must go through a preprocessing process which includes:

a. Connecting google colab with google drive:
This process involves connecting Google Colab and Google Drive to access datasets that have been stored in Google Drive in CSV format.

b. Importing required libraries in python:
This process involves several Python libraries that are required to prepare and run the model.

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.50385

c. Preprocessing

During the preprocessing stage, various methods are used, including case folding, stemming, stopword removal, and tokenizing.

1. Case folding

This process converts all letters in the text to lowercase. This is important to eliminate variations in uppercase and lowercase that are irrelevant to the meaning of the text.

2. Stemming

This process reduces words to their basic form. For example, "berjalan", "berjalan-jalan", and "berjalanlah" will be reduced to "jalan".

3. Stopword removal

This process removes words that have no significant meaning in the text, such as "this", "that", "and", etc.

4. Text cleaning

The data cleaning steps include removing numbers, punctuation, repeated characters, triple dots, words containing numbers, HTML markup, URLs, and extra spaces. By implementing this cleaning process, the resulting data will be cleaner and ready for further analysis. This process is very important to ensure that the classification model to be developed can work with higher accuracy.

5. Tokenizing

This process breaks down text into smaller individual units, such as words, punctuation, and symbols called tokens. An example of tokenizing the sentence "I like learning Python." would produce the following tokens: ["I", "like", "learn", "Python", "."].

2.4. Modelling

At this modeling stage, the data is divided into two parts, namely training data and testing data. The word weighting technique used is TF-IDF (Term Frequency-Inverse Document Frequency). This data division aims to train the model on one subset of data (training data) and test its performance on another subset (test data).

a. Split Dataset

The dataset in this study was divided into two parts: training data and testing data with a ratio of 70:30 between training data and testing data, with 70% of the data used to train the model and 30% to test model performance.

b. Feature Extraction

Converting news text into a numeric representation that can be used by a classification algorithm is an important step in text processing. TF-IDF (Term Frequency-Inverse Document Frequency), which calculates the frequency of occurrence of words in each document. This method helps in giving weight to significant words, thus producing a more meaningful representation and supporting the performance of the classification algorithm in identifying patterns and information from the text. This gives additional weight to words that are not very common and can provide more relevant information to distinguish documents [8]. Here is the TF-IDF formula:

1. Term Frequency (TF) :

$$TF(t,d) = \frac{number\ of\ words\ t\ in\ document\ d}{total\ words\ in\ document\ d}$$

2. Inverse Document Frequency (IDF) :

$$IDF(t,D) = \log\left(\frac{total\ documents\ in\ corpus\ D}{number\ of\ documents\ the\ word\ t + 1}\right)$$

3. TF-IDF :

$$TF - IDF(t,d,D) = TF(t,d) \times (t,D)$$

In the manual calculation example, the researcher explains the text vectorization process using a single test data sample. The results of the TF-IDF calculation using the TfidfVectorizer are as follows:

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.50385

An example of text taken from the training data is as follows:

"video bayi lahir selamat runtuh gempa maroko"

The results of the TF-IDF calculation using TfidfVectorizer are as follows:

a) Word: 'bayi', TF-IDF: 0.38052
b) Word: 'gempa', TF-IDF: 0.29837
c) Word: 'lahir', TF-IDF: 0.41734
d) Word: 'maroko', TF-IDF: 0.47478
e) Word: 'runtuh', TF-IDF: 0.43418
f) Word: 'selamat', TF-IDF: 0.35549
g) Word: 'video', TF-IDF: 0.22738

The TF-IDF calculation process for the word "bayi" is described as follows:

a. Term Frequency (TF) :
For the word "bayi", the TF is calculated using the following formula:

$$TF_{bayi} = \frac{1}{7} \approx 0.142857$$

where 1 represents the frequency of the word "bayi" in the text, and 7 represents the total number of words in the text.

b. Inverse Document Frequency (IDF) :
The IDF is calculated using the following formula:

$$Idf_{bayi} = \log\left(\frac{1+3661}{1+79}\right) + 1$$
$$Idf_{bayi} = \log\left(\frac{3662}{80}\right) + 1$$
$$IDF_{bayi} = \log(45.775) + 1$$
$$= 1.66 + 1$$
$$\approx 2.66$$

where 3661 represents the total number of documents in the corpus, and 79 represents the number of documents containing the word "bayi".

c. TF-IDF :
The TF-IDF value for the word "bayi" is calculated as follows:

$$TF - IDF_{bayi} = 0.142857 \times 2.66 \approx 0.38052$$

In this way, TF-IDF provides a measure of the importance of the word "bayi" within the context of the text and the overall corpus

4. Multinomial Naïve Bayes

The Multinomial Naive Bayes model classifies data with discrete features, such as text represented by TF-IDF. This model assumes that each feature (word in the text) is conditioned independently on the class it belongs to (Hoax or non-Hoax), and uses a multinomial distribution to model the possible combinations of words in each class.

$$P(c|d) \propto P(c) \prod_{i=1}^{n} P(t_i \mid c)^{f_i}$$

Where:

a) $P(c|d)$: is the probability of class c given document d
b) $\propto$: This symbol indicates "proportional to." We use proportional because what is important in classification is comparing probabilities between classes, and the actual values do not need to be known.
c) $P(c)$: is the prior probability of class (c), which is obtained from the class distribution on the training dataset.
d) $\prod_{i=1}^{n} P(x_i|C)^{count(x_i)}$: This is the likelihood probability, which is calculated as the product of the conditional probabilities of each feature (x_i) in class (C) raised to the power of the number of occurrences of that feature in the observation $(count(x_i))$.

In this manual calculation example, the author explains the process of applying the Multinomial Naive Bayes algorithm for text classification.

Known Information:

a. Total words in Valid document:

$$\text{Total words} = 4826.743422719326$$

b. The TF-IDF values for the words in the document:
   1. $TF - IDF_{bayi} = 0.38052$
   2. $TF - IDF_{gempa} = 0.29837$
   3. $TF - IDF_{lahir} = 0.41734$
   4. $TF - IDF_{maroko} = 0.47478$
   5. $TF - IDF_{runtuh} = 0.43418$
   6. $TF - IDF_{selamat} = 0.35549$
   7. $TF - IDF_{video} = 0.22738$

c. The prior probability for the Valid class:
   1. $P(Valid) = \frac{Jumla \quad dokumen\ Valid}{Total\ Jumlah\ Dokumen}$
   2. $P(Valid) = \frac{1831}{3661}$
   3. $P(Valid) = 0.50014$

d. The logarithm of the probability of the word given the Valid class:
   1. $\log(P(bayi|Valid)) = -7.82511$
   2. $\log(P(gempa|Valid)) = -8.03853$
   3. $\log(P(lahir|Valid)) = -8.66086$
   4. $\log(P(maroko|Valid)) = -9.28133$
   5. $\log(P(runtuh|Valid)) = -8.66979$
   6. $\log(P(selamat|Valid)) = -8.05999$
   7. $\log(P(video|Valid)) = -7.69407$

Steps of the Calculation:

a) the prior probability for the Valid class:

$$\log P(Valid) = \log 0.50014 \approx -0.693$$

b) Likelihood for Each Word:

$$\text{TF} - \text{IDF}(\omega_i) \times \log(P(\omega_i \mid Valid))$$

The calculation for each word is as follows:

a) For `bayi`: $0.38052 \times -7.82511 = -2.97806$
b) For `gempa`: $0.29837 \times -8.03853 = -2.39853$
c) For `lahir`: $0.41734 \times -8.66086 = -3.61556$
d) For `maroko`: $0.47478 \times -9.28133 = -4.40805$
e) For `runtuh`: $0.43418 \times -8.66979 = -3.76445$
f) For `selamat`: $0.35549 \times -8.05999 = -2.86548$
g) For `video`: $0.22738 \times -7.69407 = -1.74996$

c) Calculating the Total Log-Probability for the Valid Class:

$$\log P(Valid \mid dokumen) = \log 0.50014 + \sum_{i=1}^{n} \text{TF} - \text{IDF}(\omega_i) \times \log P(\omega_i \mid Valid)$$

$$\log P(Valid \mid dokumen) = -0.693 + (-2.97806) + (-2.39853) + (-3.61556) + (-4.40805) + (-3.76445) + (-2.86548) + (-1.74996)$$

$$\log P(Valid \mid dokumen) = -0.693 - 2.97806 - 2.39853 - 3.61556 - 4.40805 - 3.76445 - 2.86548 - 1.74996$$

$$\log P(Valid \mid dokumen) = -22.46909$$

In conclusion, the final results show that the total log-probability for the Valid class is -22.46909. When the log-probability for the Hoax class is calculated using the same method, the result is -20.71404. After computing the exponential values of these log-probabilities, the probability of the Hoax class is found to be higher than that of the Valid class. Therefore, this text is more likely to be classified as "Hoax."

Jurnal Teknik Informatika Vol. 19 No. 1, April 2026 (211-220)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.50385

3. RESULTS AND DISCUSSION

3.1. Evaluation

Evaluating the performance of the classification model, the authors used metrics such as Accuracy, Precision, Recall, and F1-Score. Each of these metrics provides a different perspective on the model's performance in classifying data. Confusion matrix visualization helps in understanding the model's performance by showing the distribution of true and false predictions in more detail. These results provide a comprehensive picture of the model's performance in classifying text data as hoax or non-hoax. Here is an image of the confusion matrix results:

a. Confusion Matrix for training data

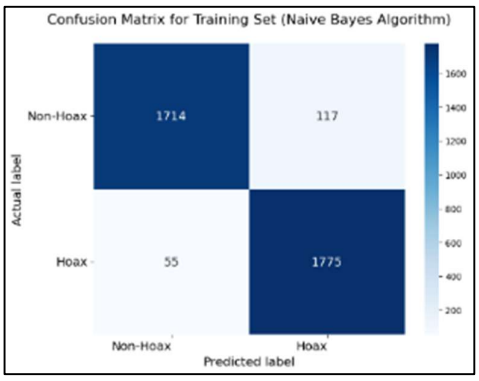

Figure 2. *Confusion matrix for training data*

The Multinomial Naive Bayes model shows very good performance on the training set. This is indicated by the high number of correct predictions for both classes:

1. The model successfully identified 1714 Non-Hoax samples correctly, and only made 117 errors by classifying Non-Hoax as Hoax.
2. The model also successfully identified 1775 Hoax samples correctly, with only 55 errors classifying Hoax as Non-Hoax.

From the results of the confusion matrix visualization, we can calculate the precision, recall, f1-score, and accuracy of each class, namely 0 (non-hoax) and 1 (hoax).

1. Class 0 (non-hoax)

a) Precision:

$$Precision = \frac{TP0}{TP0 + FP0} = \frac{1714}{1714 + 55} = \frac{1714}{1769} \approx 0.9689$$

b) Recall:

$$Recall = \frac{TP0}{TP0 + FN0} = \frac{1714}{1714 + 117} = \frac{1714}{1831} \approx 0.9361$$

c) F1-Score:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} = 2 \times \frac{0.9069}{1.905} \approx 0.9521$$

2. Class 1 (hoax)

a) Precision:

$$Precision = \frac{TP1}{TP1 + FP1} = \frac{1775}{1775 + 117} = \frac{1775}{1892} \approx 0.9381$$

b) Recall:

$$Recall = \frac{TP1}{TP1 + FN1} = \frac{1775}{1775 + 55} = \frac{1775}{1830} \approx 0.9699$$

c) F1-Score:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} = 2 \times \frac{0.9098}{1.908} \approx 0.9536$$

Jurnal Teknik Informatika Vol. 19 No. 1, April 2026 (211-220)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.50385

After obtaining the precision, recall, and f1-score values, the next step in evaluating model performance is to calculate accuracy. Mathematically, accuracy can be expressed by the formula:

$$Accuracy = \frac{number\ of\ correct\ predictions}{total\ number\ of\ predictions} =$$

$$\frac{1714 + 1775}{3661} = \frac{3489}{3661} \approx 0.9530$$

By compiling the evaluation results into a table, researchers can easily make comparisons between these metrics.

Table 1. *Training data comparison*

	Precision	recall	F1-score	support
0	0.97	0.94	0.95	1831
1	0.94	0.97	0.95	1830
Accuracy			0.95	3661
Macro avg	0.95	0.95	0.95	3661
Weighted avg	0.95	0.95	0.95	3661

b. Confusion Matrix for test data

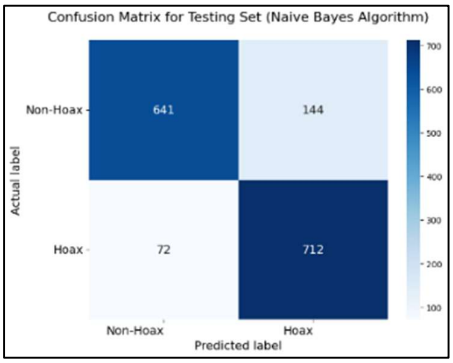

Figure 3. *Confusion matrix for testing data*

The Multinomial Naive Bayes model shows very good performance on the training set. This is indicated by the high number of correct predictions for both classes:

a. The model successfully identified 641 Non-Hoax samples correctly, with only 144 errors classifying Non-Hoax as Hoax.
b. The model successfully identified 712 Hoax samples correctly, with only 72 errors classifying Hoax as Non-Hoax.

From the results of the confusion matrix visualization, we can calculate the precision, recall, f1-score, and accuracy of each class, namely 0 (non-hoax) and 1 (hoax).

1. Class 0 (non-hoaks)

a) Precision:

$$Precision = \frac{TP0}{TP0 + FP0} = \frac{641}{641 + 72} = \frac{641}{713} \approx 0.8990$$

b) Recall:

$$Recall = \frac{TP0}{TP0 + FN0} = \frac{641}{641 + 144} = \frac{641}{785} \approx 0.8165$$

c) F1-Score:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} = 2 \times \frac{0.7340}{1.7155} \approx 0.8557$$

2. Class 1 (hoaks)

a) Precision:

$$Precision = \frac{TP1}{TP1 + FP1} = \frac{712}{712 + 144} = \frac{712}{856} \approx 0.8317$$

b) Recall:

$$Recall = \frac{TP1}{TP1 + FN1} = \frac{712}{712 + 72} = \frac{712}{784} \approx 0.9081$$

c) F1-Score:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} = 2 \times \frac{0.7552}{1.7398} \approx 0.8685$$

After obtaining the precision, recall, and f1-score values, the next step in evaluating model performance is to calculate accuracy. Mathematically, accuracy can be expressed by the formula:

$$Accuracy = \frac{number\ of\ correct\ predictions}{total\ number\ of\ predictions} =$$

$$\frac{641 + 712}{1569} = \frac{1353}{1569} \approx 0.8623$$

By compiling the evaluation results into a table, researchers can easily make comparisons between these metrics.

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.50385

Table 2. *Testing data comparison*

	Precision	recall	F1-score	support
0	0.90	0.82	0.86	785
1	0.83	0.91	0.87	784
Accuracy			0.86	1569
Macro avg	0.87	0.86	0.86	1569
Weighted avg	0.87	0.86	0.86	1569

3.2. Deployment

Based on the results of the confusion matrix and the comparison table obtained, the model performance shows satisfactory results. As a next step, the model will be deployed using the Flask framework from Python. The model is saved in pickle format.

The Flask framework was chosen because it is very useful in reducing development time and facilitating model testing in a production environment [9]. And the models are stored in pickle format because this is very important for the reproducibility of scientific experiments, where the same model can be reused without the need to retrain from scratch [10].

CONCLUSION

This study demonstrates that the Multinomial Naive Bayes algorithm can effectively classify valid and hoax news in the Indonesian context, achieving an average accuracy, precision, recall, and F1-score of 86% on the test dataset. With a relatively balanced dataset consisting of 2,612 non-hoax and 2,614 hoax news articles, the model shows consistent performance across evaluation metrics. Furthermore, the implementation of the model using the Flask framework enables practical deployment, providing a functional tool for detecting hoax news based on data from trusted sources.

The main contribution of this study is the successful deployment of a model using the Flask framework from python that has been tested and trained using datasets from trusted sources related to hoax and non-hoax labels. Thus, this study not only proves the effectiveness of the Multinomial Naive Bayes algorithm in news classification, but also provides a ready-to-use model that can be used to identify hoax news in Indonesia effectively.

Future research should focus on improving data diversity, enhancing model generalization, and extending the approach to multilingual settings. Additionally, the adoption of more advanced Natural Language Processing techniques, such as transformer-based models, is recommended to further improve classification performance and contextual understanding.

REFERENCES

[1] C. Juditha, "Interaksi Komunikasi Hoax di Media Sosial serta Antisipasinya," Jurnal Pekommas, vol. 3, no. 1, 2018.

[2] B. Zaman, A. Justitia, K. N. Sani, and E. Purwanti, "An Indonesian Hoax News Detection System Using Reader Feedback and Naïve Bayes Algorithm," Cybernetics and Information Technologies, vol. 20, pp. 82–94, 2020, doi: 10.2478/cait-2020-0006.

[3] A. Rahman and A. Doewes, "Online News Classification Using Multinomial Naive Bayes," Jurnal Ilmiah Teknologi dan Informasi, vol. 6, no. 1, pp. 32–38, 2017, [Online]. Available: www.kompas.com

[4] Y. Pan, H. Gao, H. Lin, Z. Liu, L. Tang, and S. Li, "Identification of Bacteriophage Virion Proteins Using Multinomial Naïve Bayes with g-Gap Feature Tree," Int J Mol Sci, vol. 19, no. 6, p. 1779, Jun. 2018, doi: 10.3390/ijms19061779.

[5] N. Walter, J. Cohen, R. L. Holbert, and Y. Morag, "Fact-Checking: A Meta-Analysis of What Works and for Whom," Polit Commun, vol. 37, no. 3, pp. 350–375, May 2020, doi: 10.1080/10584609.2019.1668894.

[6] S. R. S. Gowda, B. R. Archana, P. Shettigar, and K. K. Satyarthi, "Sentiment Analysis of Twitter Data Using Naïve Bayes Classifier," 2022, pp. 1227–1234. doi: 10.1007/978-981-16-3690-5_117.

[7] S. Robertson, "Understanding inverse document frequency: On theoretical arguments for IDF," Journal of Documentation, vol. 60, no. 5, pp. 503–520, 2004, doi: 10.1108/00220410410560582.

[8] A. S. Amirul Haj, V. Amrizal, and A. Arini, "Analisis Sentimen Kinerja KPU Pemilu 2019 Menggunakan Algoritma K-Means Dengan Algoritma Confix

Stripping Stemmer," Journal of Innovation Information Technology and Application (JINITA), vol. 2, no. 01, pp. 9–18, Jun. 2020, doi: 10.35970/jinita.v2i01.119.

[9] P. Singh, Deploy Machine Learning Models to Production. Berkeley, CA: Apress, 2021. doi: 10.1007/978-1-4842-6546-8.

[10] H. Sayyed, "How to create & run pickle file for machine learning model." Accessed: Jul. 27, 2024. [Online]. Available: https://studygyaan.com/data-science/how-to-create-run-pickle-file-for-machine-learning-model

[11] R. Anggrainingsih, G. M. Hassan, and A. Datta, "Evaluating BERT-based language models for detecting misinformation," Neural Computing and Applications, 2025.