# COMPARISON BETWEEN CNN AND RANDOM FOREST PERFORMANCE IN DETECTING HOAX INDONESIAN NEWS ARTICLES

**[1]Franciska Nugrahaeni Siwi Pramudya, [2]Yonathan Purbo Santosa,**
[1,2]Program Studi Teknik Informatika Fakultas Ilmu Komputer,
Universitas Katolik Soegijapranata
[2]yonathansantosa@unika.ac.id

## ABSTRACT

*Hoax news is a serious problem in this era. Many people are easily led by opinions made by certain people without seeing the truth or looking for existing facts. To overcome this, many researchers have conducted hoax news detection using various algorithms. In some studies, it is said that Random Forest has better performance to overcome this hoax news problem. In other studies, it is also said that CNN has the same level of performance as the Random Forest algorithm. In addition, the problem that is often found is the error in prediction due to improper preprocessing methods. Therefore, in this research, the appropriate preprocessing method is searched by using several preprocessing scenarios for the Convolutional Neural Network (CNN) and Random Forest algorithms. Therefore, in addition to finding the right preprocessing method for each algorithm, a performance comparison is also carried out on the CNN and Random Forest algorithms using a dataset of 4000 news facts from Kompas.com and 4000 hoax news from the turnback.hoax site. the results obtained in this study are random forest has an average model accuracy value of 90% and the CNN algorithm has an average model accuracy value of 60% using the same extraction method, namely TFIDF combined with Ngrams worth one or unigram*

**Keywords:** Hoax news detection, Convolutional Neural Network(CNN), Random Forest.

## INTRODUCTION

Information and news have always been very important parts of life. Especially in this era, information and news can be obtained and accessed anywhere, whether through newspapers, TV, social media, and many more. Because of the many accesses to information and news that exist, there are also many people who take advantage of this. Many people take advantage of this moment to spread untrue and false news and information, commonly called hoaxes. As a result of this, it is easy for people to be swayed by opinions, which has a negative impact on their thoughts and perceptions of something that might be good.

To minimize this, the government has created a system to detect hoax news and information through the Ministry of Communication and Information Technology to establish the Anti-Hoax Society movement, which aims to make the public aware of fake news. In addition, many scientists and researchers have conducted studies comparing the accuracy of hoax news and information detection. In this case, many researchers applied various algorithms to solve the problem. In research that discusses the detection of hoax news in Indonesian using deep learning, the methods

used are CNN and LSTM. The CNN algorithm in this study has a dataset accuracy of 88%, while LSTM has an accuracy of 84%.[1] Whereas in research that discusses the detection of Indonesian hoax content using Expansion with Word2Vec, the Random Forest algorithm excels with an accuracy rate of 89% compared to other comparison algorithms, namely Logistic Regression and Support Vector Machine (SVM)[2].

Seeing the comparison of the accuracy level above, which is only a few percent adrift, it can be concluded that the Random Forest algorithm is better for solving this problem. Therefore, in this study, the author wants to compare the two algorithms using the same dataset. The author uses several preprocessing scenarios that aim to compare which algorithm has better performance and what type of preprocessing is suitable for the dataset.

## RESEARCH METHODOLOGY

### *2.1 Data Collection*

The dataset is taken through the Kaggle.com site with the name Indonesian Fact and Hoax Political News. In the dataset there are several sources of articles from various media such as from cnn, kompas, tempo, and turnbackhoax. But the dataset that will be used in this study is a dataset from the kompas.com site for fact news and from the turnbackhoax site for hoax news. The dataset is in the form of an XLSX file consisting of 4,750 fact news data and 10,383 for hoax news. In this study, the dataset used is 4,000 data from each news site. Datasets can be downloaded via the link https://www.kaggle.com/datasets/linkgish/indonesian-fact-and-hoax-political-news[1]. Datasets will be used for training and test data.

### *2.2 Data Preprocessing*

The first step is to perform the sentiment analysis process. The dataset will be labeled, namely whether the news article is classified as negative or positive sentiment. Then preprocessing will be done with 3 scenarios, namely:

1. Scenario 1 : Stop words are not removed, slang words are not converted to standard words, and objects/subjects are not removed. Used to determine the sentiment analysis model's accuracy without the three preprocessing steps.
2. Scenario 2 : Removal of stop words and slang words are not converted to standard words and subjects/objects are not removed. Used to determine the accuracy if removing words as well as not converting slang and not removing subjects/objects.
3. Scenario 3 : Stop words were removed and slang words were converted to standard words and objects/subjects were not removed. It is used to determine the effect of removing stop words and converting slang words. However, in this study, the

[1] kaggle.com/datasets/linkgish/indonesian-fact-and-hoax-political-news. (accessed on May 17, 2023)

conversion of slang words to standard words was not carried out because the news site for its sentences certainly used standard sentences.

After the three scenarios have been carried out, the next step is Word Feature extraction. In this research, the extraction methods that will be used are N-Gram, Count Vectorizer, and TF-IDF. After the three methods above are carried out, data with the results of the execution of each scenario will be tried on each algorithm, namely CNN and Random Forest. After that, accuracy, precision, and recall tests will be carried out on each algorithm for each scenario tested. The highest value of each algorithm and scenario will be seen, and the best value of the four scenarios from each algorithm will be used in completing hoax news detection in the next step.

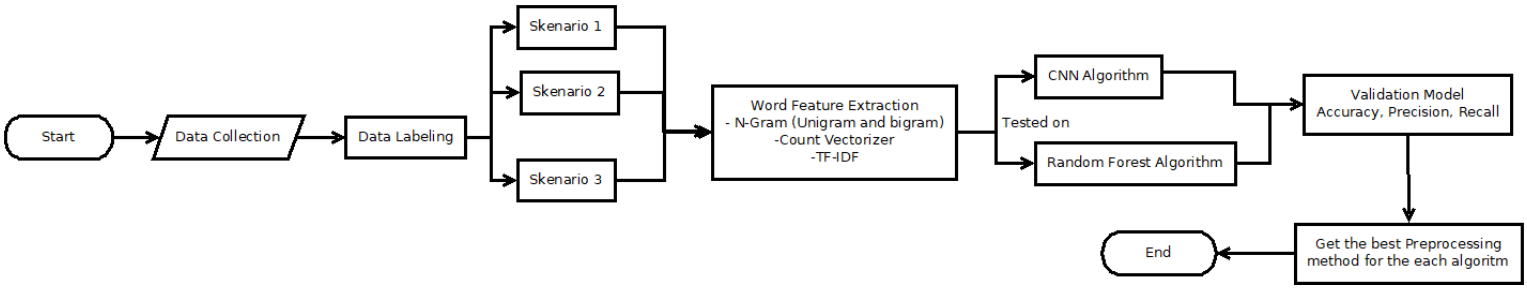


**Figure 1. Flowchart Preprocessing**

### *2.3 Compare Both Algoritms*

After each algorithm gets a good preprocessing method for the dataset used, a comparison will be made between the two algorithms to determine which algorithm has the best performance for detecting hoax articles with the preprocessing method that has been chosen. The first step is to preprocess the dataset with the processing method that has been obtained in the previous step for each algorithm. Then the second step is to test the prediction results of each algorithm method with existing data, and then it will be seen whether the prediction is correct or not.

The next step is to validate the model by looking at the values of accuracy, precision, and recall. After getting the results of the accuracy, precision, and recall values of each algorithm, a temporary comparison between the two algorithms will be made. Then the last step taken is that the model will be tested again using a new dataset to ensure and see if, if there is a new dataset, the accuracy, precision, and recall values will change, such as increasing or decreasing. In addition, if each algorithm is tested with new data, it will also be seen whether prediction errors will occur or not. After that, the next step is to compare the two algorithms to determine which one has the best performance to overcome the hoax news article problem based on the prediction results and the value of the model validation based on the results of several tests on each algorithm with new data.

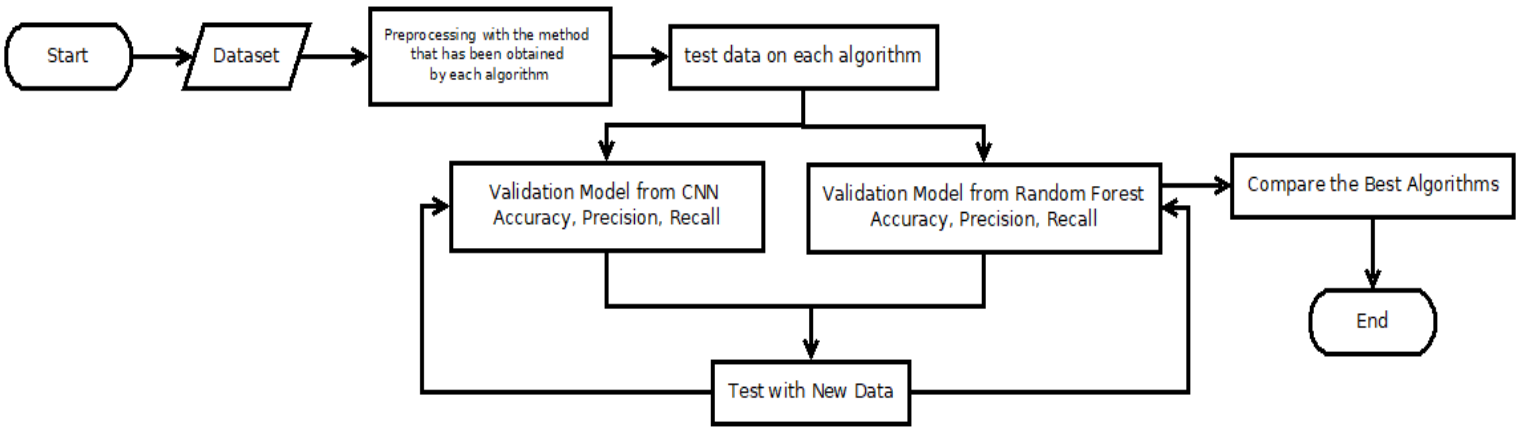


**Figure 2. Flowchart Compare Algorithm**

### *2.4 Conclusion Retrieval*

After model validation is generated using trials with new data, the next step is to draw conclusions from these trials. After the data is inputted anew, the comparison will be seen from the accuracy, precision, and recall values to see whether the two algorithms increase or decrease. After being observed, it will be determined which of the two algorithms has the best performance for detecting hoax news in Indonesian.

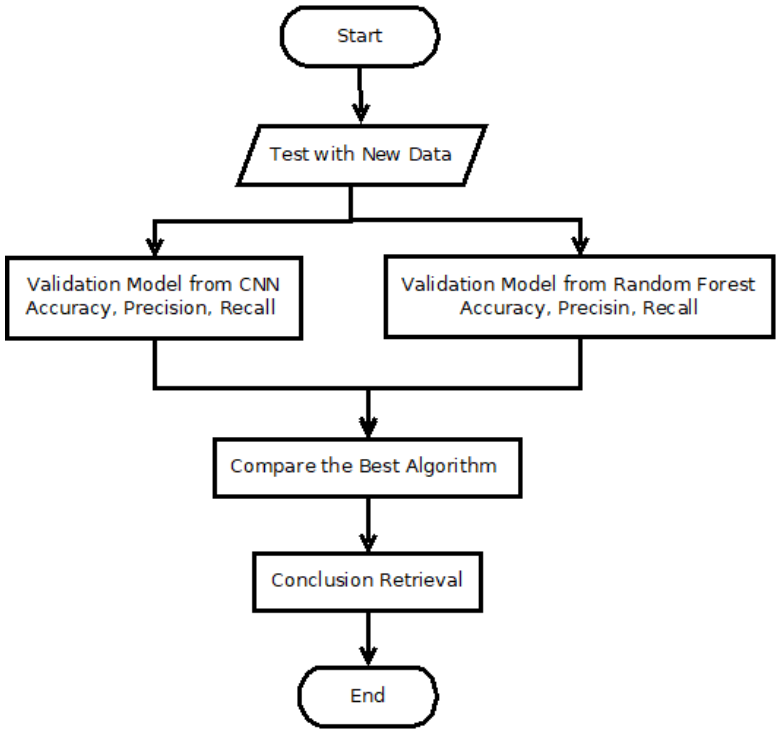


**Figure 3. Flowchart Conclusion Retrieval**

## RESULT

In this study, researchers have tried the three preprocessing methods to both algorithms, namely CNN and Random Forest. After researchers found the value of accuracy, precision, recall, and f1 score in each scenario and each algorithm, researchers compared the parameter values of each scenario to be taken and made a news prediction model. In this study, researchers took the highest precision value from each scenario.

### *3.1 Result For Preprocessing Scenario 1*

**Table 1. CNN Result For Preprocessing Scenario 1**

| PARAMETER | Count Vectorizer | | TF-IDF | |
|---|---|---|---|---|
| | Unigram | Bigram | Unigram | Bigram |
| Accuracy | 0.548125 | 0.541875 | 0.537500 | 0.506250 |
| Precision | 0.548384 | 0.542557 | 0.537520 | 0.525370 |
| Recall | 0.548125 | 0.541875 | 0.537500 | 0.506250 |
| F1 Score | 0.543173 | 0.541665 | 0.537509 | 0.436157 |

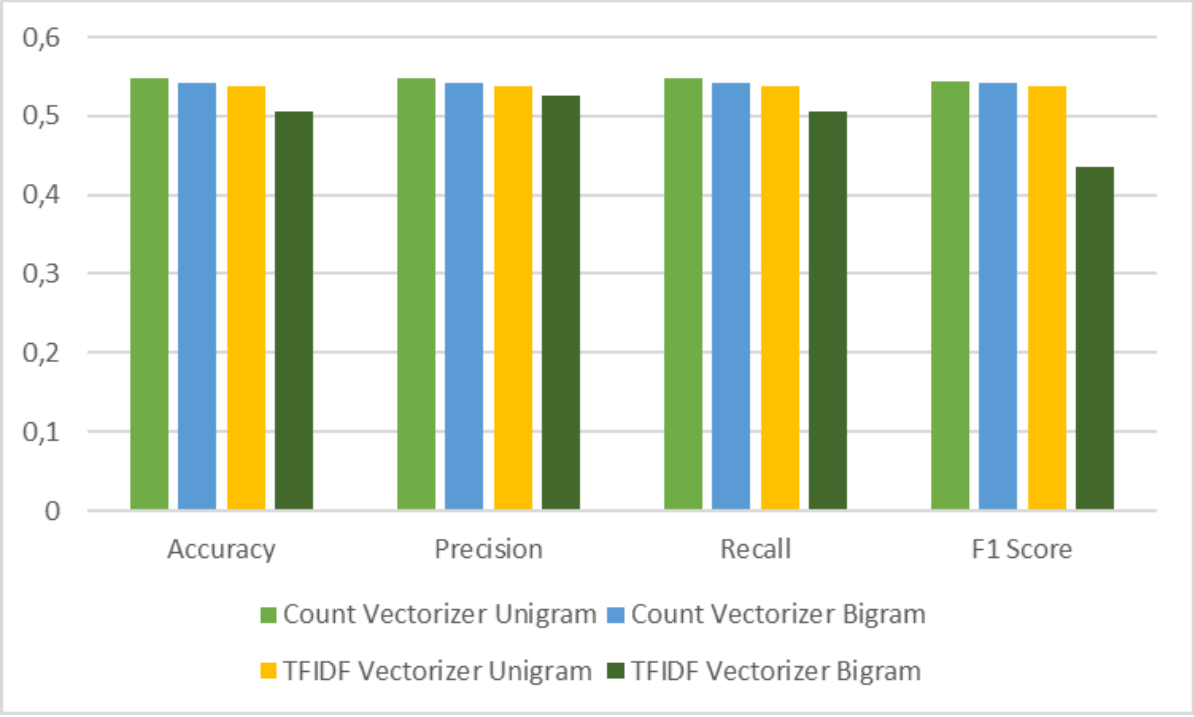


**Figure 4. CNN Scenario 1 Preprocessing Result Graph**

In table 1 researchers used various parameters and one layer. The activation convolution functions used in this study are relu and softmax. In addition, there are also several other parameters used, namely a filter of size 64 and a kernel of size 5. Researchers also use 10 iterations and use the earlystopping function to stop the model training process when its performance on validation data no longer increases or even decreases. The results of this scenario 1 on the CNN model have a low value even in the 55% range with the highest precision value being in the unigram count vectorizer extraction method. in addition, the value of F1 Score in the bigram extraction method also has a very low value, which is only 43%.

**Table 2. Random Forest Result For Preprocessing Scenario 1**

| PARAMETER | Count Vectorizer | | TF-IDF | |
|---|---|---|---|---|
| | Unigram | Bigram | Unigram | Bigram |
| Accuracy | 0.801875 | 0.860625 | 0.803750 | 0.867500 |
| Precision | 0.986641 | 0.811738 | 0.986717 | 0.823198 |
| Recall | 0.625151 | 0.932569 | 0.628778 | 0.930025 |
| F1 Score | 0.765358 | 0.867969 | 0.768094 | 0.873357 |

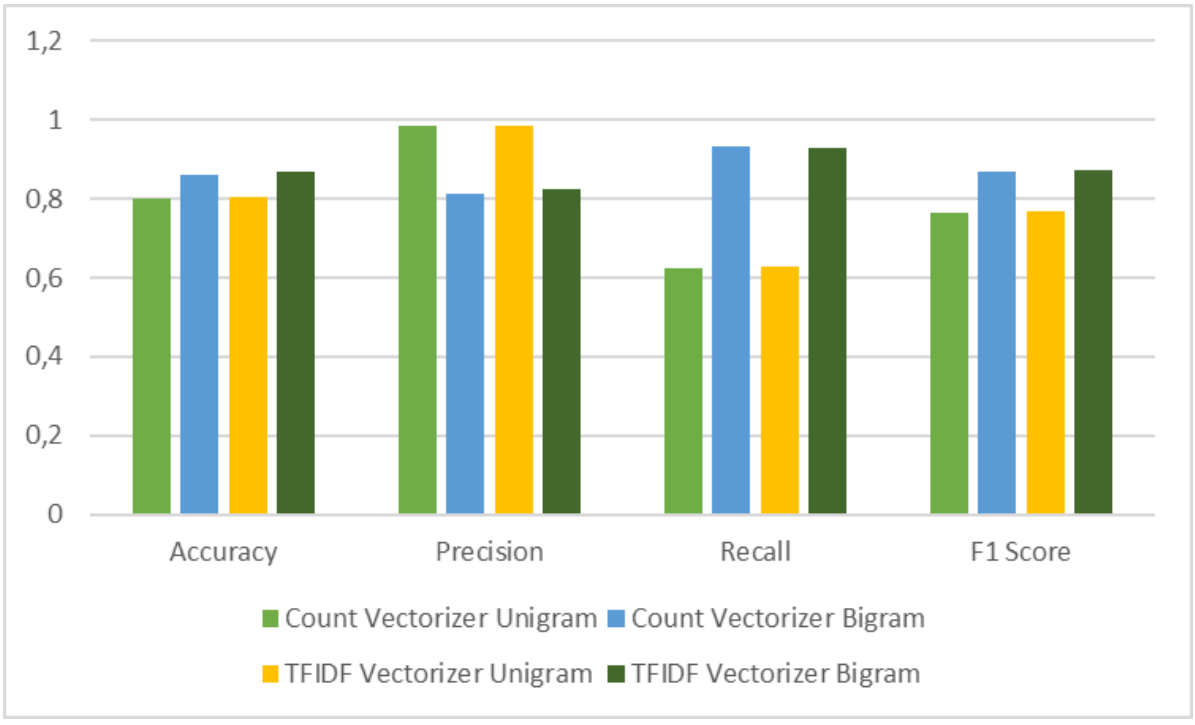


**Figure 5. Random Forest Scenario 1 Preprocessing Result Graph**

In table 2 researchers used the library from sklearn to use the random forest model. The parameters used in this preprocessing are n_estimator of 100, seed value or random state of 42,

and the number of tree depths of 5. In this scenario, as in table 2, the results of the accuracy and precision values of all extraction methods are quite high, which are in the range of values of 80% and above. As for the recall and F1 Score values in the extraction method that uses the n_gram 1 value, it has a low value below 80%. in contrast to the recall and F1 Score values of the n_gram 2 extraction method, it has a high value, which is above 86%.

### *3.2 Result For Preprocessing Scenario 2*

**Table 3. CNN Result For Preprocessing Scenario 2**

| PARAMETER | Count Vectorizer | | TF-IDF | |
|---|---|---|---|---|
| | Unigram | Bigram | Unigram | Bigram |
| Accuracy | 0.570625 | 0.560625 | 0.578750 | 0.558749 |
| Precision | 0.577519 | 0.560425 | 0.578580 | 0.566506 |
| Recall | 0.570625 | 0.560625 | 0.578750 | 0.558750 |
| F1 Score | 0.555709 | 0.559354 | 0.578433 | 0.550457 |

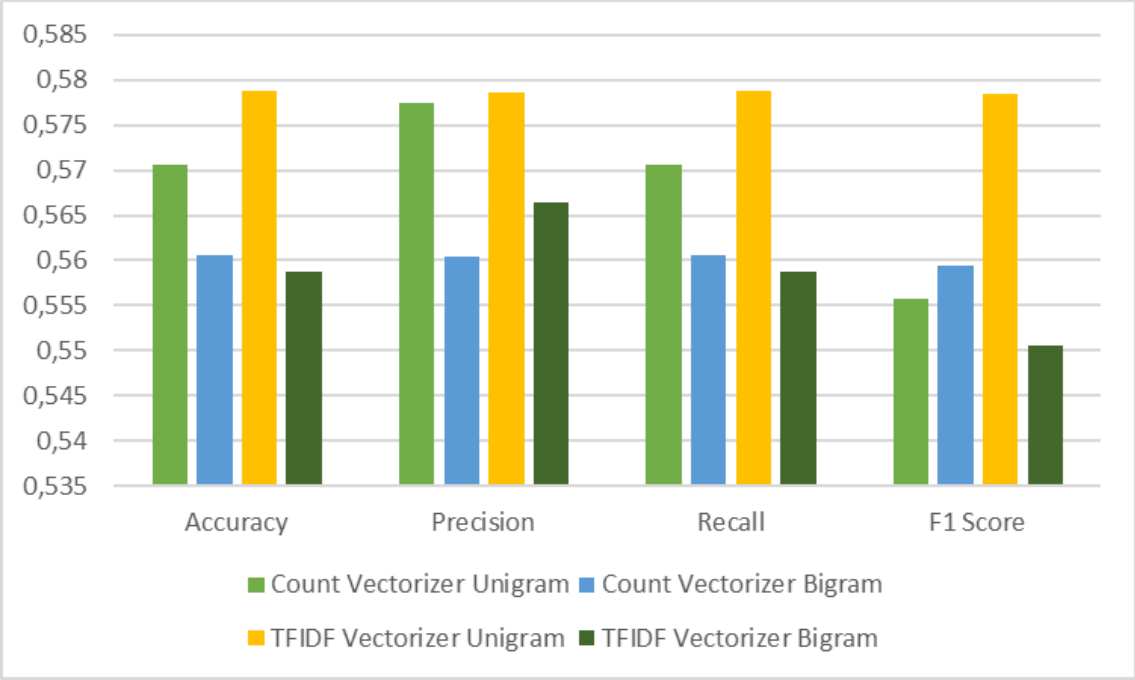


**Figure 6. CNN Scenario 2 Preprocessing Result Graph**

In table 3 researchers used various parameters and one layer. The activation convolution functions used in this study are relu and softmax. In addition, there are also several other parameters used, namely a filter of size 64 and a kernel of size 5. Researchers also use 10 iterations

and use the earlystopping function to stop the model training process when its performance on validation data no longer increases or even decreases. In this scenario 2, the value of this CNN algorithm is more improved than scenario 1 in table 4.1, besides that the value of each extraction is also significant, which is in the range of 55-57%.

**Table 4. Random Forest Result For Preprocessing Scenario 2**

| PARAMETER | Count Vectorizer | | TF-IDF | |
|---|---|---|---|---|
| | Unigram | Bigram | Unigram | Bigram |
| Accuracy | 0.834375 | 0.776875 | 0.828125 | 0.778125 |
| Precision | 0.787909 | 0.698060 | 0.782786 | 0.700092 |
| Recall | 0.929866 | 0.961832 | 0.923821 | 0.959287 |
| F1 Score | 0.853022 | 0.808988 | 0.847476 | 0.809447 |

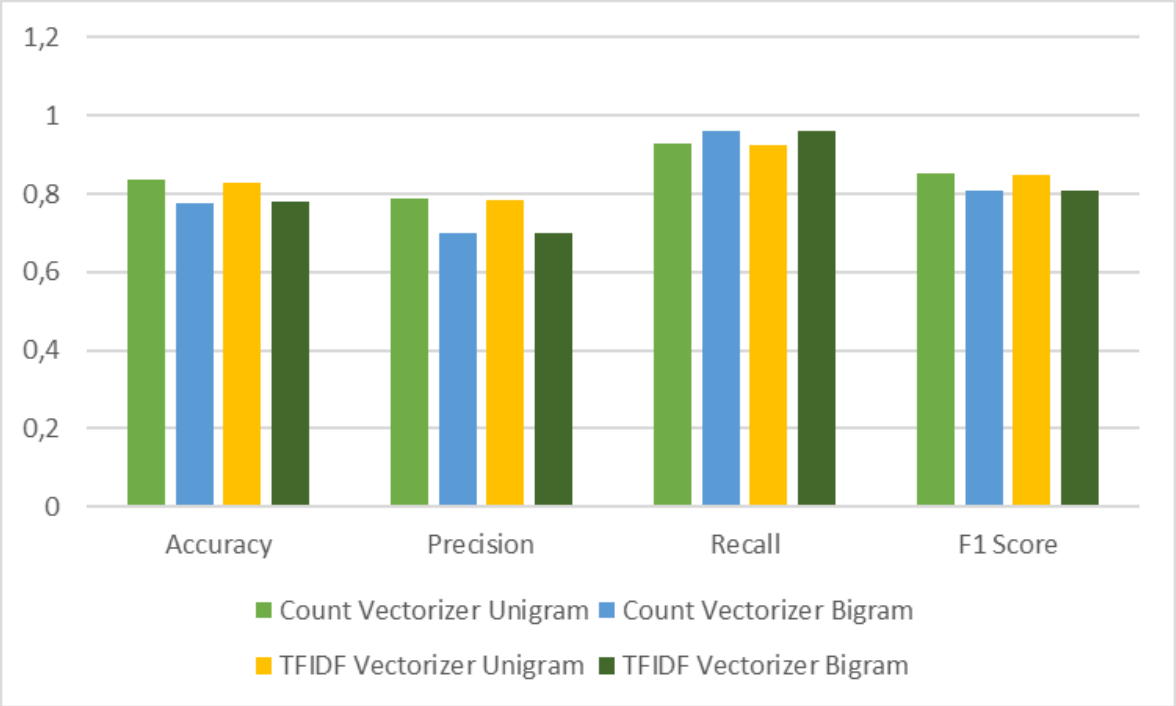


**Figure 7. Random Forest Scenario 2 Preprocessing Result Graph**

In table 4 researchers used the library from sklearn to use the random forest model. The parameters used in this preprocessing are n_estimator of 100, seed value or random state of 42, and the number of tree depths of 5. In this second scenario, the random forest algorithm actually decreased from the first scenario as in table 1. Contrary to the previous scenario, in this scenario

the lowest value is actually obtained from the extraction method using ngram 2 or bigram, while in scenario 1 the extraction method using ngram 2 or bigram actually has the highest value.

### *3.3 Result For Preprocessing Scenario 3*

**Table 5. CNN Result For Preprocessing Scenario 3**

| PARAMETER | Count Vectorizer | | TF-IDF | |
|---|---|---|---|---|
| | Unigram | Bigram | Unigram | Bigram |
| Accuracy | 0.550000 | 0.555000 | 0.521250 | 0.540624 |
| Precision | 0.549695 | 0.564931 | 0.521168 | 0.551097 |
| Recall | 0.550000 | 0.555000 | 0.521250 | 0.540625 |
| F1 Score | 0.548667 | 0.528908 | 0.521196 | 0.524773 |

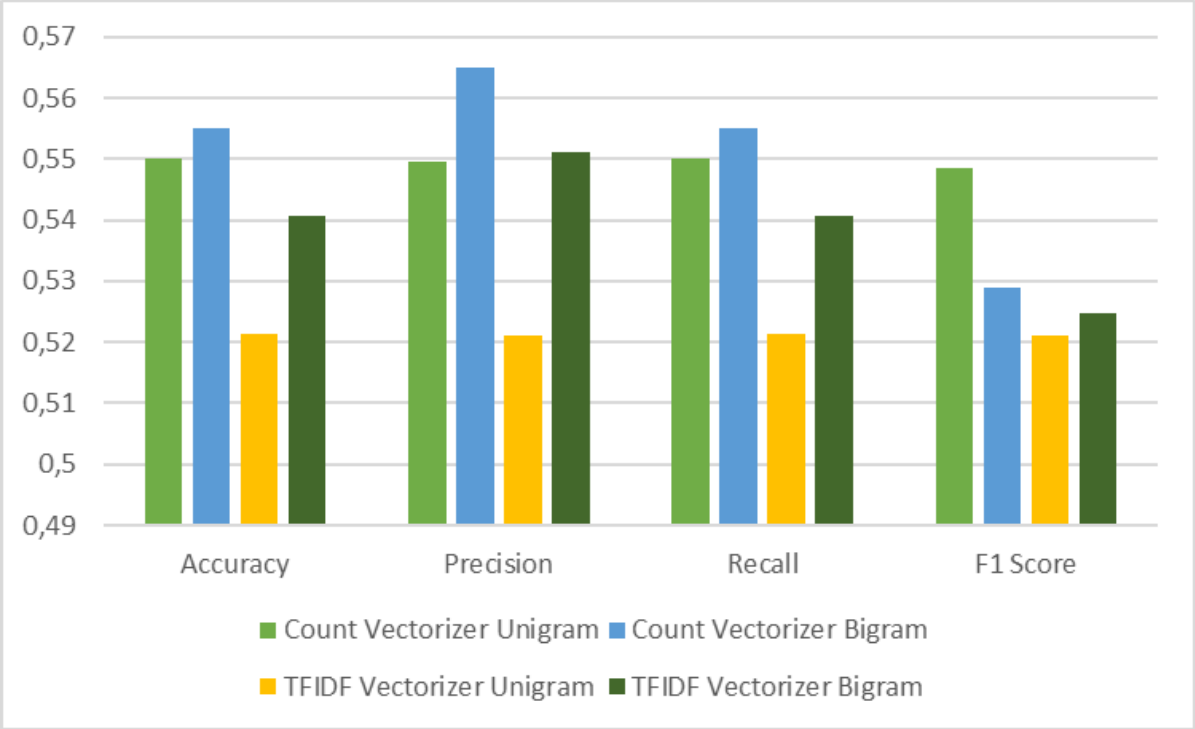


**Figure 8. CNN Scenario 3 Preprocessing Result Graph**

In table 5 researchers used various parameters and one layer. The activation convolution functions used in this study are relu and softmax. In addition, there are also several other parameters used, namely a filter of size 64 and a kernel of size 5. Researchers also use 10 iterations and use the earlystopping function to stop the model training process when its performance on validation data no longer increases or even decreases. In this scenario, the value of accuracy and precision in the count vectorizer extraction method both bigram and unigram has the same value

of 55%. In addition, in the TFIDF Vectorizer extraction method, the value of accuracy and recall on unigram has the same value of 52%.

**Table 6. Random Forest Result For Preprocessing Scenario 3**

| PARAMETER | Count Vectorizer | | TF-IDF | |
|---|---|---|---|---|
| | Unigram | Bigram | Unigram | Bigram |
| Accuracy | 0.785000 | 0.845625 | 0.797500 | 0.846875 |
| Precision | 0.983967 | 0.787620 | 0.980879 | 0.789304 |
| Recall | 0.593712 | 0.938931 | 0.620314 | 0.938931 |
| F1 Score | 0.740573 | 0.856645 | 0.760000 | 0.857640 |

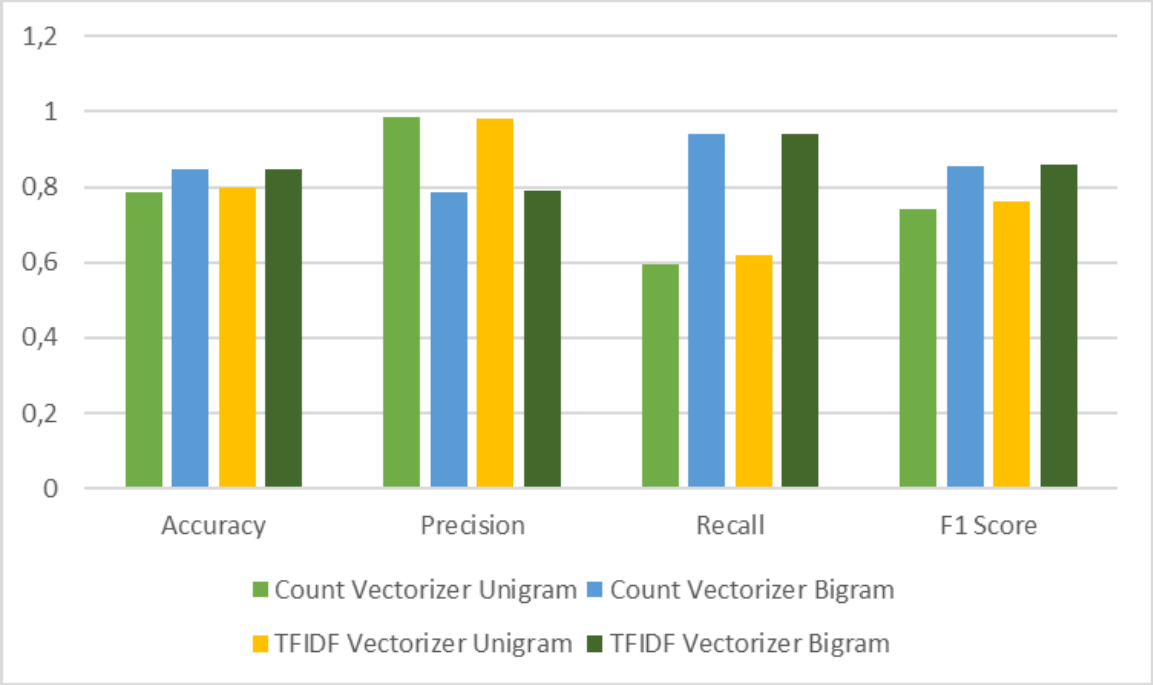


**Figure 9. Random Forest Scenario 3 Preprocessing Result Graph**

In table 6 researchers used the library from sklearn to use the random forest model. The parameters used in this preprocessing are n_estimator of 100, seed value or random state of 42, and the number of tree depths of 5. In this third scenario, the resulting value of each evaluation metric is the same as the first scenario, where the value of using ngram 2 or bigram is higher than if using ngram 1 or unigram.

### *3.4 Result For Model Prediction CNN*

**Tabel 7 Recapitulation Of Precision Value Of CNN Algorithm**

| Metode Ekstraksi | SKENARIO 1 | SKENARIO 2 | SKENARIO 3 |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Count Vectorizer Unigram | 0,548384 | 0,577519 | 0,549695 |
| Count Vectorizer Bigram | 0,542557 | 0,560425 | 0,564931 |
| TFIDF Vectorizer Unigram | 0,53752 | 0,57858 | 0,521168 |
| TFIDF Vectorizer Bigram | 0,52537 | 0,566506 | 0,551097 |

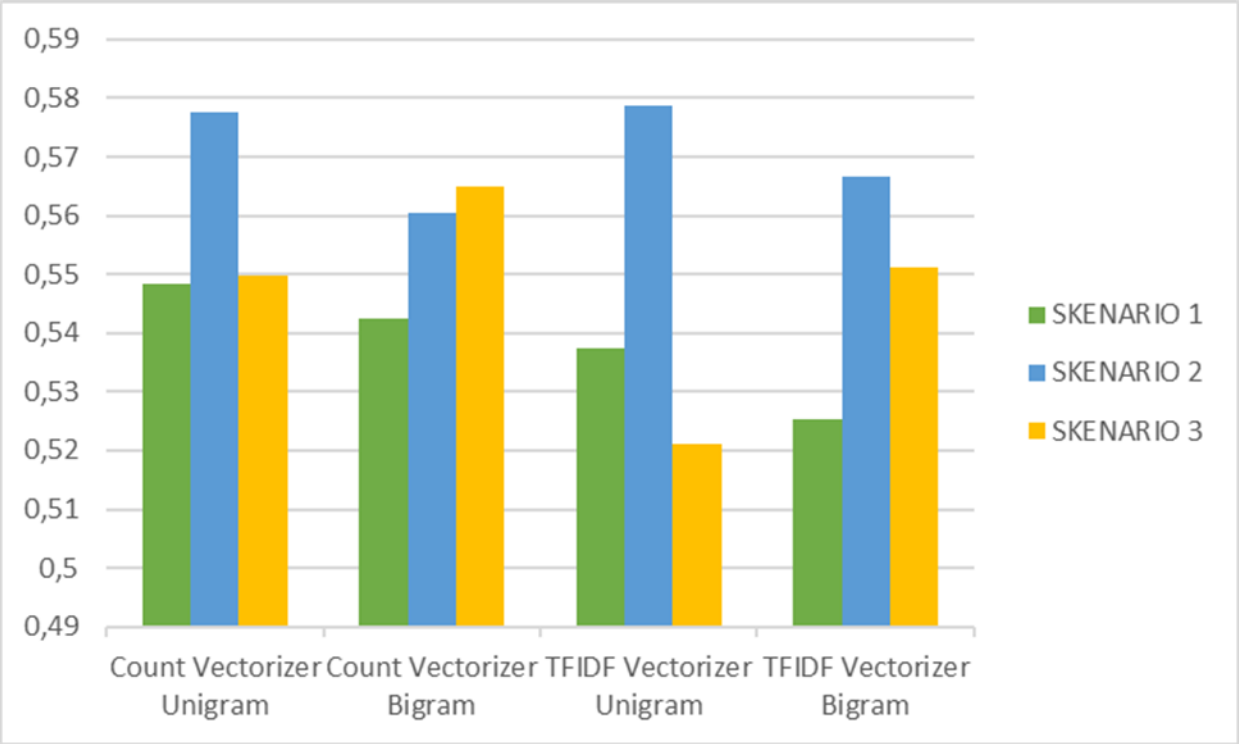


**Figure 10. CNN Algorithm Precision Value Graph**

Based on the graphic image in figure 4.7 above and from the three preprocessing methods that have been carried out on the CNN algorithm, the values of accuracy, precision, recall, and F1 Score are higher in scenario 2, this proves that a more complete preprocessing method is better for this CNN algorithm. Therefore, the suitable preprocessing on this dataset for the CNN algorithm is to use scenario 2 preprocessing with the TFIDF Unigram extraction method. After the prediction model is created, the resulting accuracy, recall, F1 Score value is 60% and precision is 61%. And after the new dataset is tested on the CNN algorithm model, there are still frequent prediction errors, this is because the values of accuracy, precison, recall, and F1 Score are low and only in the range of 60%.

### *3.5 Result For Model Prediction Random Forest*

**Tabel 8 Recapitulation Of Precision Value Of Random Forest Algorithm**

| Extraction Methods | SKENARIO 1 | SKENARIO 2 | SKENARIO 3 |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Count Vectorizer Unigram | 0,986641 | 0,787909 | 0,983967 |
| Count Vectorizer Bigram | 0,811738 | 0,69806 | 0,78762 |
| TFIDF Vectorizer Unigram | 0,986717 | 0,782786 | 0,980879 |
| TFIDF Vectorizer Bigram | 0,823198 | 0,700092 | 0,789304 |

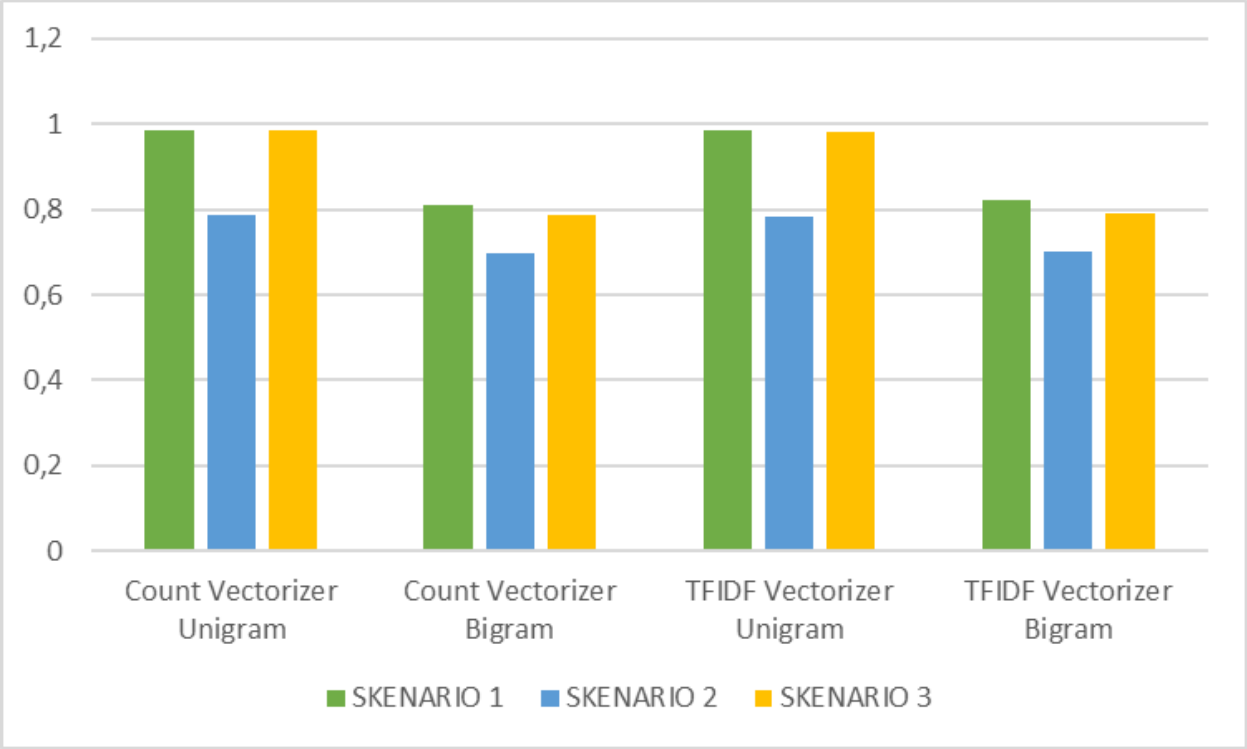


**Figure 11. Random Forest Algorithm Precision Value Graph**

Based on the graphic image in figure 4.8 above and from the three preprocessing methods that have been carried out on the Random Forest algorithm, the values of accuracy, precision, recall, and F1 Score are higher in scenario 1, this proves that for this dataset, the Random Forest algorithm only requires several preprocessing methods. So in the Random Forest algorithm, suitable preprocessing on this dataset for the Random Forest algorithm is to use scenario 1 preprocessing with the TFIDF Unigram extraction method. After the prediction model is created, the resulting accuracy value is 94%, precision 95%, recall 92% and F1 score of 94%. These results are certainly very far compared to the CNN algorithm, and then after the data is tested with a new dataset, the Random Forest algorithm does not yet have a prediction error, this is because the value obtained is quite high, which is around the 90% range.

## CONCLUSION

Based on the results that have been obtained from the research conducted, for the appropriate preprocessing method for the CNN algorithm on the dataset used is to use preprocessing scenario 2, namely by using the dataset converted to lowercase, removing numbers, removing symbols and removing stop words then slang words are not converted to standard words and subjects / objects are not removed then with the TFIDF Vectorizer extraction method with ngram 1 or unigram, this is seen based on the highest precision value and the stability of the results of the evaluation metrics of each scenario. Furthermore, for the random forest algorithm, the preprocessing method that is suitable for this hoax news detection problem in Indonesian is to use preprocessing method 1 where the dataset is only converted to lowercase, removal of numbers, removal of symbols then with the same extraction method as CNN, namely TFIDF with ngram 1 or unigram.

From the detection results of the two algorithms that have been tested with the latest dataset, random forest has better performance in detecting hoax news in Indonesian based on the prediction results obtained and the final value results in terms of accuracy, precision, recall and F1 Score values. This is proved by a random forest algorithm that is almost always better at predicting news than a CNN algorithm that has too many predictive errors. In addition, the final evaluation metric value of the two algorithms is also quite far adrift where CNN has an accuracy value of 60% while the random forest algorithm has an accuracy value of 94%.

Suggestions for further research are that other variations of preprocessing methods can be added so that they can be more complete and varied, as well as finding the right extraction method especially for the CNN algorithm so that there is no need to convert numpy matrix to numpy sparsetensor.

## REFERENCES


[1] A. A. Kurniawan and M. Mustikasari, "Implementasi Deep Learning Menggunakan Metode CNN dan LSTM untuk Menentukan Berita Palsu dalam Bahasa Indonesia," *JIUP*, vol. 5, no. 4, p. 544, Dec. 2021, doi: 10.32493/informatika.v5i4.6760.

[2] Friskadini Ismayanti and Erwin Budi Setiawan, "Deteksi Konten Hoax Berbahasa Indonesia di Twitter Menggunakan Fitur Ekspansi dengan Word2Vec," *openlibrarypublications.telkomuniversity.ac.id*, 2020, [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/15697

[3] F. Rahutomo, I. Y. R. Pratiwi, and D. M. Ramadhani, "Eksperimen Naïve Bayes Pada Deteksi Berita Hoax Berbahasa Indonesia," *Jurnal PKOP*, vol. 23, no. 1, Jul. 2019, doi: 10.33299/jpkop.23.1.1805.

[4] S. Khomsah and Agus Sasmito Aribowo, "Text-Preprocessing Model Youtube Comments in Indonesian," *RESTI*, vol. 4, no. 4, pp. 648–654, Aug. 2020, doi: 10.29207/resti.v4i4.2035.

[5] T. T. A. Putri, "ANALYSIS AND DETECTION OF HOAX CONTENTS IN INDONESIAN NEWS BASED ON MACHINE LEARNING," vol. 4, no. 1, 2019.

[6] D. S. Wahyuni and Y. Sibaroni, "Comparison of Ensemble Methods for Detecting Hoax News," *bits*, vol. 4, no. 2, Sep. 2022, doi: 10.47065/bits.v4i2.1957.
[7] H. K. Farid, E. B. Setiawan, and I. Kurniawan, "Implementation Information Gain Feature Selection for Hoax News Detection on Twitter using Convolutional Neural Network (CNN)," vol. 5, no. 3, 2021.
[8] D. T. Hermanto, A. Setyanto, and E. T. Luthfi, "Algoritma LSTM-CNN untuk Sentimen Klasifikasi dengan Word2vec pada Media Online," vol. 8, no. 1, 2021.
[9] D. Alita and A. R. Isnain, "Pendeteksian Sarkasme pada Proses Analisis Sentimen Menggunakan Random Forest Classifier," *komputasi*, vol. 8, no. 2, Oct. 2020, doi: 10.23960/komputasi.v8i2.2615.
[10] T. Pranckevičius and V. Marcinkevičius, "Comparison of Naive Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression Classifiers for Text Reviews Classification," *BJMC*, vol. 5, no. 2, 2017, doi: 10.22364/bjmc.2017.5.2.05