

---

## Hadith Text Classification Based on Topic Using Convolutional Neural Network (CNN) and TF-IDF

M. Rafi Athallah<sup>1</sup>, Kemas Muslim Lhaksana<sup>2</sup>

<sup>1</sup>Faculty of Informatics, Telkom University, Jl. Terusan Buah Batu, Bandung, 40257, Indonesia, [athallahrafi@student.telkomuniversity.ac.id](mailto:athallahrafi@student.telkomuniversity.ac.id)

<sup>2</sup>Faculty of Informatics, Telkom University, Jl. Terusan Buah Batu, Bandung, 40257, Indonesia, [kemasmuslim@telkomuniversity.ac.id](mailto:kemasmuslim@telkomuniversity.ac.id)

✉Corresponding Author: [kemasmuslim@telkomuniversity.ac.id](mailto:kemasmuslim@telkomuniversity.ac.id)

Received: December 30, 2024

Revision: January 15, 2025

Accepted: March 10, 2025

---

### Abstract

Convolutional Neural Networks (CNN) will develop a hadith classification system to categorize texts based on specific topics or categories. This study compares two text representation techniques, namely Term Frequency-Inverse Document Frequency (TF-IDF) and Word2Vec, concerning the application of stemming and without stemming in the process. This study utilizes Category ID 0-5. About 2,845 data have been processed as required for testing. The data was divided into two parts, with a proportion of 80:20 for training and testing. Next, several models were evaluated, namely Word2Vec with stemming, TFIDFCNN without stemming, and TFIDFCNN with stemming. Accuracy, precision, recall, and F1 score metrics were used to assess the performance. The results show that the TFIDFCNN model without stemming performs best with 85% accuracy in topic-based text classification. This is due to the stability and efficiency of the model in processing data.

**Keywords:** Convolutional Neural Network (CNN), Hadith, TFIDF, Word2Vec

---

### Introduction

The Arabic term “hadith” refers to the act of conveying or reporting an event. Hadith is an oral tradition in Islam that includes the sayings, actions, and approvals of the Prophet Nabi Muhammad SAW that were transmitted in secret. Each hadith has a chain of transmitters, which is a lineage of individuals who heard and transmitted the hadith. Hadith are seen to provide guidance in various aspects of life, even though the Qur'an has only a few verses directly related to law (Irfan Renaldy, 2020), (Umi Hani, 2022). However, Muslims face the challenge of classifying traditions into categories such as recommendation, prohibition, information, or a combination of these categories. This complicates the process of learning hadith as well as applying their content and meaning in daily life. Therefore, there is a need for a classification system that can categorize hadith based on topics (Hanafi et al., 2020), to facilitate Muslims in practicing the teachings contained in the hadith (Harish et al., 2021)

Research (Santoso, 2019) categorized authentic hadith from Sahih Bukhari into six main categories based on their underlying themes. The process involved various preprocessing techniques, such as tokenization, case normalization, punctuation removal, and common words elimination. The study results showed that the application of TF-IDF normalization improved the classification accuracy to 89.29%, compared to 87.97% when normalization was not applied. Another study (Hanafi et al., 2020) utilized the translation of Bukhari's hadith into Bahasa Indonesia to develop a multi-label classification system. In this study, the k-Nearest Neighbor (KNN) algorithm was applied with Mutual Information (MI) as the feature selection method. The preprocessing process includes tokenization, stemming, and noise removal, followed by feature extraction using TF-IDF. The best performance was achieved with a parameter value of  $k = 7$  and a Hamming Loss of 0.0886. Research (Harish et al., 2021) exhaustively examines the application of feature selection and classification methods using Random Forest to categorize data into advice, prohibition, and information categories based on Relief scores. The study highlighted the importance of the feature selection process in improving classification accuracy, where optimal results were achieved without using stemming and with a threshold value of 0.1. A comparison between stemmed and non-stemmed data using TF-IDF and Random Forest shows that non-stemmed data yields higher accuracy. Of the feature selection methods tested, namely Chi-Square, Mutual Information, and ReliefF, Chi-Square produced the highest accuracy of 91.7%. Research (Rasenda, Rini, Nova, 2024) explored the use of artificial neural networks for categorize hadith from the books of Muslim and Bukhari into command and prohibition categories. The study emphasized the importance of effective text preprocessing and feature extraction using word embeddings. The neural network architecture with two layers and 256 neurons proved to be the most accurate, achieving an accuracy rate of 97.72%, underscoring the potential of neural networks in improving the understanding of religious texts. TF-IDF is also discussed in other studies. Research (Abubakar & Umar, 2022) evaluated the effectiveness of feature extraction methods, such as TF-IDF and Word2Vec, in classifying emotions from tweets related to commuter lines. This study used Support Vector

Machine (SVM) and Multinomial Naïve Bayes (MNB) algorithms. The results show that the combination of SVM with TF-IDF provides the best performance in emotion classification on tweets about Commuterline and Transjakarta, with the highest achievement in accuracy, precision, recall, and F1-measure metrics. In contrast, SVM with Word2Vec showed the lowest performance, especially in detecting emotions on smaller datasets. This study concluded that TF-IDF is more effective than Word2Vec for this kind of task. Research (Xiao et al., 2024) discusses various text vectorization techniques, such as Bag of Words, TF-IDF, Word2Vec, and Doc2Vec, used in sentiment classification. The article emphasized the importance of converting text data into numerical vectors in order to be processed by machine learning algorithms. The results show that TF-IDF is generally superior to Word2Vec and Doc2Vec in sentiment classification of book reviews. In addition, this study recommends combining TF-IDF with Word2Vec to improve classification accuracy based on various evaluation metrics. The results of the studies discussed in this document show that TF-IDF feature vector representation generally outperforms other vectorization methods such as word2vec and doc2vec, especially in sentiment classification of book reviews. The combination of TF-IDF with word2vec results in more informative feature vectors, which improves the performance of sentiment classification in terms of accuracy, precision, recall, and F1 score.

The contribution of this research utilizes a Convolutional Neural Network (CNN) to support the development of a hadith classification system. The main focus of this research is to integrate the CNN model with the text representation using Term Frequency- Inverse Document Frequency (TF-IDF) and Word2Vec without involving additional feature selection. Preprocessing was applied to the text, including the option of using stemming or no stemming, to evaluate the impact of preprocessing on the performance of the model. The results were used to compare the effectiveness of the two text representation techniques and assess the CNN model's ability to categorize hadith texts into specific topics.

This research consists of several sections. The second section includes a literature review that discusses related research, either in terms of techniques, methods, or classifications, and includes sub-sections relevant to the research topic. The third section will specifically review the methods, techniques and other relevant things that will be applied, along with an overview of the system architecture. While the fourth section will discuss the accuracy results obtained.

## Literature Review

### Preprocessing

Data preprocessing is the first stage in data processing, involving various text-related processes. These processes aim to prepare and process the text before it is analyzed, with a focus on removing irrelevant information from the text. Most of the processing involves breaking the text into words or tokens, converting letters to lowercase, as well as other related steps that are considered an essential part of processing (Nilla & Setiawan, 2024), (Sihombing et al., 2024). Text preprocessing generally involves several stages, including (Suryaningrum, 2023):

1. Case Folding, used to equalize text and avoid differences due to differences in upper and lower case in analysis, by converting all capital letters to lower case. This step is very important to make it easier to find information.
2. Data Cleaning, a step that removes elements in the data, such as punctuation marks, symbols, excess spaces, duplicate characters, and numbers, that may affect the analysis results.
3. Normalization, a stage where word forms are adjusted, for example by correcting spelling mistakes or changing abbreviations to their full form.
4. Tokenization, a method that divides sentences in a dataset into separate words by using spaces or punctuation, while ignoring irrelevant characters and numbers.
5. Stopword removal, which involves removing words such as question words, conjunctions, and words that do not have a significant impact on data processing.
6. Stemming is the process of returning words such as nouns, verbs, and adjectives to their basic or root form (Rizky Amalia Putri & Terza Damaliana, 2024).

### Term Frequency- Inverse Document Frequency (TF-IDF)

Term Frequency- Inverse Document Frequency (TF-IDF) method is used to measure the importance or weight of a word (token) in a document relative to a collection of other documents. This technique is often applied in text processing and information retrieval systems to analyze the relationship between words in a particular document (Abubakar & Umar, 2022), (Suryaningrum, 2023).

$$C(t, d) \times \log \frac{N_d}{N_{dt}} \quad (1)$$

The raw frequency of occurrence of a word or term in document  $d$  is represented by the pair  $(t, d)$ , while the total number of documents in the corpus is denoted by  $N_d$ , and the number of documents containing term  $t$  is shown in (Abubakar & Umar, 2022).

### Word2Vec

Word2Vec is a subset of embedding techniques that utilize simple neural networks to map words into a low-dimensional vector space based on their linguistic context. This method generates distributed vector representations for words under the assumption that words with similar meanings in a given context will have similar vector representations. However, similar vectors do not necessarily reflect identical meanings. The Word2Vec network consists of two main layers. The training process of the first projection layer uses backpropagation and stochastic gradient descent algorithms. This layer is in charge of generating a continuous vector representation for words in the context of  $n$ -grams. When words in  $n$ -grams that co-occur frequently have similar activation weights, correlative relationships are formed. The  $V \times N$  matrix  $W$

in this network represents the weights connecting the input layer to the projection layer and the projection layer to the output layer, while the W matrix connecting the projection layer to the output layer is  $N \times D$ , with D being the dimension of the output layer and N being the dimension of the projection layer (Abubakar & Umar, 2022), (Xiao et al., 2024), (Dharma et al., 2022).

**Convolutional Neural Network (CNN)**

The working principle of the human brain forms deep learning models such as Convolutional Neural Network (CNN) (Alsaleh & Larabi-Marie-Sainte, 2021), (Kurniawan & Mustikasari, 2020). CNNs can handle data with grid-like network structures well. CNN has three dimensions-one for processing text and signals, two for images or audio, and three for analyzing video. CNN can also classify text well, although it is often used in computer vision to classify images. For text classification, word vectors created through word concatenation techniques are used. This technique is similar to the way for image classification (Nilla & Setiawan, 2024).

Google developed TensorFlow, a machine learning framework that utilizes tensors to extend the concept of vectors and matrices to multiple dimensions. The Keras TensorFlow library is used to design CNN models with multiple layers, including input layers, 1D conv, 1D flatten max pooling, dense layers, as well as dropouts to reduce the risk of overfitting. The 1D CNN was used to recognize patterns from one-dimensional text data, with dropouts applied at various stages of training to maintain model performance. The model is equipped with a dense layer that uses the ReLU activation function, as well as an output layer with a softmax activation function to generate the probability of each class. CNNs were chosen for their ability to provide accurate predictions (Nilla & Setiawan, 2024), (Nisa & Kuan, 2021), (Ramadhanti & Setiawan, 2023). The model architecture design can be seen in Figure 1.

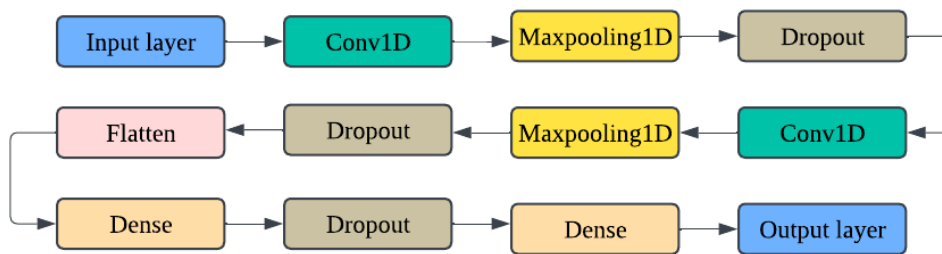


Figure 1. Architecture CNN Model

**Confusion Matrix**

One of the main challenges in computational statistics and machine learning is evaluating classification results. When an algorithm is used to distinguish between two states of a dataset, such as positive and negative, the results are usually presented in the form of a two-class confusion matrix that shows the number of correct and incorrect predictions. Data that is mistakenly categorized as negative is called false negative (FN), while data that is correctly categorized as positive is called true positive (TP). Conversely, data that is correctly categorized as negative is called true negative (TN), and data that is negative but incorrectly categorized as positive is called false positive (FP). The full details can be seen in Table 1 for easier understanding (Heydarian et al., 2022), (Chicco et al., 2021).

Table 1. Cconfusion Matrix

Class	Predicted	
	Positif	Negatif
Positif	TP (True Positive)	FN (False Negative)
Negatif	FP (False Positive)	TN (True Negative)

In order to produce a more objective classification, metrics such as accuracy, precision, recall, and F1 score are required to produce a more objective classification. Here are the definitions and formulas of each metric (Grandini et al., 2020):

1. Accuracy is a performance metric calculated by dividing the amount of data correctly classified by the system by the total amount of data available. The formula to calculate it is as follows.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

2. Precision measures the proportion of correct positive results out of all positive results generated by the classification system. The calculation formula is as follows.

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

3. Recall describes the proportion of correct positive results that are successfully identified by the system. The formula used to calculate it is as follows.

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

4. F1 Score reflects the balance between precision and recall through the calculation of harmonic mean. The calculation formula is as follows.

$$F1\ Score = 2 \times \frac{precision \times recall}{precision + recall} \tag{5}$$

### Materials & Methods

The process of the system developed in this research is shown in Figure 2. The implemented stages include dataset, pre-processing, feature extraction, CNN model application, and evaluation using certain metrics.

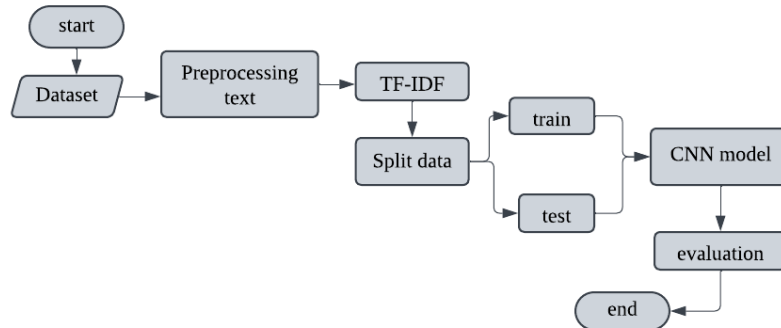


Figure 2. TF IDF & CNN Architecture

The dataset used in this research amounts to about 40,000 data and is obtained from sunnah.com. The dataset includes various features, such as collection, bookNumber, chapterID, EnglishBabNumber, hadithNumber, ourhadithNumber, arabicURN, and arabicBabName. Furthermore, this data is processed through a preprocessing stage to suit the needs of the research. After going through the preprocessing and labelling process, the primary dataset for this study presented in Labels in Table 2 is divided into categories from 0 to 5, where label 0 represents hajj, 1 represents doomsday, 2 represents marriage, 3 represents fasting, 4 represents prayer, and 5 represents zakat. Before the preprocessing stage, the raw dataset containing about 2,845 data was summarized and named Ready Dataset or Data preprocessing. This dataset includes four main features, namely collection, Englishgrade1, cleaned\_text, category, and category\_id. The hadith used in this study are already available in English translation. The category\_id feature is generated from the labelling process.

Table 2. Data Preprocessing

Collection	English Grade 1	Cleaned_Text	Category	Category_id
bukhari	sahih	Allah’s Apostle said: Islam is based on (the f..)	haji	0
bukhari	sahih	Allah’s Apostle was asked, “What is the best d..	haji	0
...	...	...	...	...
bukhari	sahih	While the Prophet was in funeral procession...	zakat	5
bukhari	sahih	Allah’s Apostle said, “Horses are kept for on...	zakat	5

The preprocessing stage in this study involved five main steps: text cleaning, case folding, tokenization, stopword removal, and stemming. Some column names were changed to simplify the analysis process. This research uses two types of preprocessing, namely without stemming and with stemming. The preprocessing result dataset without stemming is shown in Table 3 and named TFIDFCNN without stemming, while the preprocessing result dataset with stemming is shown in Table 4 under the name TFIDFCNN with stemming.

Table 3. TFIDFCNN without Stemming

Collection	Grade	Hadist_Text	Category	Category_id
bukhari	sahih	[allahs, apostle, said, islam, based, following...	haji	0
bukhari	sahih	[allahs, apostle, asked, best, deed, replied...	haji	0
...	...	...	...	...
bukhari	sahih	[prophet, funeral, procession, picked, somethi...	zakat	5
bukhari	sahih	[allahs, apostle, said, horses, kept, one, thr...	zakat	5

Table 4. TFIDFCNN with Stemming

Hadist_Text	Category_id	Tokenized_Text
[allahs, apostle, said, islam, based, following...	0	['allah', 'apostl', 'islam', 'base', '...
[allahs, apostle, asked, best, deed, replied...	0	['allah', 'apostl', 'ask', 'best', 'deed', 're...
...	...	...
[prophet, funeral, procession, picked, somethi...	5	['prophet', 'funer', 'process', 'took', 'small...
[allahs, apostle, said, horses, kept, one, thr...	5	['allah', 'aposit', 'said',

The TFIDFCNN models, both stemming and non-stemming, utilize scikit-learn's to convert the text in the "hadist\_text" column into a numeric vector. In this study, the TfidfVectorizer is limited to consider a maximum of 5000 features, with the aim of reducing the data dimension that is too large and retaining only significant features based on the TF-IDF value. The category labels contained in the category\_id column, which include values 0-5, are then transformed using the one-hot encoding method to match the data division for training and testing.

In both TFIDFCNN models, the CNN architecture is built using the Keras TensorFlow library with multiple layers, including input layer, 1D convolution, flatten layer, 1D maximal pooling, dense layer, and dropout to reduce potential overfitting. The 1D CNN is used to recognize patterns in one-dimensional text data, with the application of dropout techniques at multiple training stages to maintain model performance. The model is equipped with a dense layer that uses the ReLU activation function and an output layer that uses the softmax activation function to generate class probabilities. Model evaluation is done using confusion matrix, precision, recall, f1 score, and accuracy.

This research not only compares TFIDFCNN models with and without stemming, but also compares the results with feature extraction using Word2Vec. The Word2Vec embedding method utilizes a simple neural network to map words into a low-dimensional vector space based on their linguistic context. This technique generates a distributed vector representation for words, assuming that words that have similar meanings in a given context will have similar vector representations (Abubakar & Umar, 2022). The Word2Vec architecture used with CNN is shown in Figure 3.

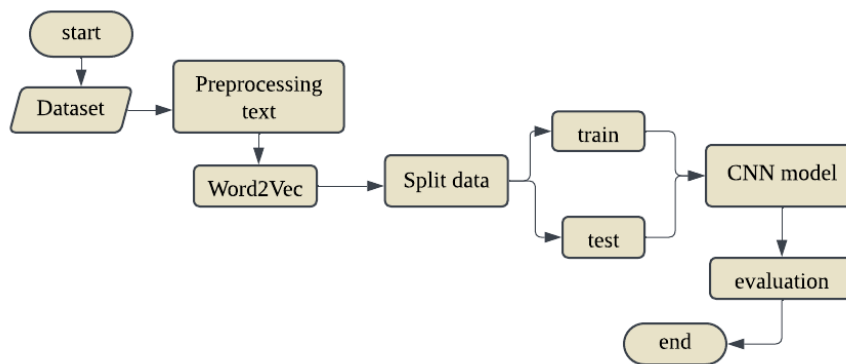


Figure 3. Word2Vec & CNN Design

As seen in Table 4, the processed dataset is used to develop Word2vec. The hadist\_text and category\_id features are applied in this process. Furthermore, the features were incorporated and prepared to divide the data into training and testing sets. This research is named the Word2vecCNN model. It was built using a similar approach to the previous two TFIDFCNN models and then evaluated.

### Results and Discussion

This research develops three models, namely TFIDFCNN with stemming, TFIDFCNN without stemming, and Word2Vec as a comparison in feature extraction. The data sharing ratio for both TFIDFCNN and Word2VecCNN models is 80:20. Table 5 shows the test results using CNN.

Table 5. Test Results

	Category_id	TFIDFCNN without stemming (80:20)	TFIDFCNN with stemming (80:20)	Word2VecCNN with stemming (80:20)
Precision (%)	0	0.90	0.87	0.84
	1	0.92	0.85	1.00
	2	0.80	0.79	0.84
	3	0.90	0.88	0.92
	4	0.84	0.82	0.66
	5	0.96	0.82	0.75
Recall (%)	0	0.85	0.84	0.75
	1	0.76	0.70	0.36
	2	0.85	0.77	0.58
	3	0.87	0.74	0.55
	4	0.85	0.87	0.96
	5	0.90	0.90	0.60
F1-score (%)	0	0.87	0.85	0.79
	1	0.83	0.77	0.53
	2	0.82	0.78	0.68
	3	0.88	0.80	0.69
	4	0.85	0.84	0.79



	5	0.93	0.86	0.67
Accuracy (%)		0.85	0.83	0.75

The test results for each category\_id (0-5) show a comparison of the performance of three different models in classifying the data based on the metrics of precision, recall, F1 score, and accuracy. In terms of precision, the TFIDFCNN model without stemming shows the highest value in category 5 (0.96), but lower in category 4 (0.84). The TFIDFCNN model with stemming experienced a decrease in accuracy in most categories, with the largest decrease occurring in category 5 (0.82). Meanwhile, the Word2Vec model with stemming shows a more varied level of accuracy, with the highest values in category 1 (1.00) and category 0 (0.84).

On recall, the TFIDFCNN model without stemming has higher recall values in categories 0, 2, 3, 4, and 5, although it is lower in category 1 (0.76) compared to the other categories. The TFIDFCNN model with stemming showed increased recall in categories 4 (0.87) and 5 (0.90), but a significant decrease in category 1 (0.70). Word2VecCNN with stemming showed the lowest recall in most categories, with the lowest value in category 1 (0.36), indicating the difficulty of the model in identifying this category.

On F1 score, the TFIDFCNN model without stemming showed good results in most categories, especially in category 0 (0.87) and category 3 (0.88), reflecting a good balance between precision and recall. The TFIDFCNN model with stemming shows a slight decrease in F1 score in almost all categories, especially in category 1 (0.77) and category 3 (0.80), which indicates a decrease in performance due to the use of stemming. The Word2Vec model with stemming shows a lower F1 score compared to the other two models. The TFIDFCNN model without stemming has the highest accuracy of 0.85 when compared to the other two models.

## Conclusions

The results show that the objectives have been successfully achieved. The TFIDFCNN model without stemming is compared with Word2VecCNN with stemming and TFIDFCNN with stemming on various key evaluation metrics, such as accuracy, precision, and F1-score. The TFIDFCNN model without stemming recorded precision rates of 0.90, 0.92, 0.80, 0.90, 0.84, and 0.96, with recall scores of 0.85, 0.76, 0.85, 0.87, 0.85, and 0.90, respectively. In addition, the F1 scores for the six category\_ids with values of 0 to 5 were 0.87, 0.83, 0.82, 0.88, 0.85, and 0.93, with a total accuracy of 0.85. Overall, the model proved superior in recognizing patterns and classifying data effectively. In addition, the model proved to be more effective in recognizing patterns and classifying data with a high degree of accuracy. This finding indicates that text representation using TF-IDF without additional preprocessing, such as stemming, is more suitable for the dataset used. This is consistent with the objective of the study, which was to identify the optimal method for the development of a hadith classification system.

In this case, TFIDFCNN without stemming shows better performance in terms of stability and effectiveness of the model in data classification, which indicates that the use of TF-IDF features without additional preprocessing such as stemming may be more suitable for this dataset. On the other hand, Word2VecCNN with stemming showed worse performance, possibly due to the model's difficulty in identifying relevant patterns in the text data, despite using Word2Vec-based word representation. Thus, TFIDFCNN without stemming is considered superior to Word2Vec-based models for research and development of text-based emotion detection systems.

## References

- Abubakar, H. D., & Umar, M. (2022). Sentiment Classification: Review of Text Vectorization Methods: Bag of Words, Tf-Idf, Word2vec and Doc2vec. *SLU Journal of Science and Technology*, 4(1 & 2), 27-33. <https://doi.org/10.56471/slujst.v4i.266>
- Alsaleh, D., & Larabi-Marie-Sainte, S. (2021). Arabic Text Classification Using Convolutional Neural Network and Genetic Algorithms. *IEEE Access*, 9, 91670-91685. <https://doi.org/10.1109/ACCESS.2021.3091376>
- Chicco, D., Tötsch, N., & Jurman, G. (2021). The matthews correlation coefficient (Mcc) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. *BioData Mining*, 14, 1-22. <https://doi.org/10.1186/s13040-021-00244-z>
- Dharma, E. M., Lumban Gaol, F., Leslie, H., Warnars, H. S., & Soewito, B. (2022). The Accuracy Comparison Among Word2Vec, Glove, and Fasttext Towards Convolution Neural Network (CNN) Text Classification. *Journal of Theoretical and Applied Information Technology*, 31(2). [www.jatit.org](http://www.jatit.org)
- Grandini, M., Bagli, E., & Visani, G. (2020). *Metrics for Multi-Class Classification: an Overview*. <http://arxiv.org/abs/2008.05756>
- Hanafi, A., Adiwijaya, A., & Astuti, W. (2020). Klasifikasi Multi Label pada Hadis Bukhari Terjemahan Bahasa Indonesia Menggunakan Mutual Information dan k-Nearest Neighbor. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, 9(3), 357-364. <https://doi.org/10.32736/sisfokom.v9i3.980>
- Harish, H. N., Al Faraby, S., & Dwifabri, M. (2021). Klasifikasi Multi Label Pada Hadis Bukhari Terjemahan Bahasa Indonesia menggunakan Random Forest, Mutual Information, dan Chi-Square. 8(5), 10583.
- Heydarian, M., Doyle, T. E., & Samavi, R. (2022). MLCM: Multi-Label Confusion Matrix. *IEEE Access*, 10, 19083-19095. <https://doi.org/10.1109/ACCESS.2022.3151048>
- Irfan Renaldy. (2020). Analisis perbandingan algoritma k-nearest neighbor dengan algoritma support vector machine pada pengklasifikasian hadits shahih muslim studi kasus: hadits shahih muslim pada software ensiklopedi hadits "kitab 9 imam." *Fakultas Sains Dan Teknologi UIN Syarif Hidayatullah Jakarta*. <https://repository.uinjkt.ac.id/dspace/handle/123456789/55999>

- Kurniawan, A. A., & Mustikasari, M. (2020). Implementasi Deep Learning Menggunakan Metode CNN dan LSTM untuk Menentukan Berita Palsu dalam Bahasa Indonesia. 5(4), 2622–4615. <https://doi.org/10.32493/informatika.v5i4.7760>
- Nilla, A., & Setiawan, E. B. (2024). Film Recommendation System Using Content-Based Filtering and the Convolutional Neural Network (CNN) Classification Methods. *Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika*, 10(1), 17. <https://doi.org/10.26555/jiteki.v9i4.28113>
- Nisa, E. C., & Kuan, Y. Der. (2021). Comparative assessment to predict and forecast water-cooled chiller power consumption using machine learning and deep learning algorithms. *Sustainability (Switzerland)*, 13(2), 1–18. <https://doi.org/10.3390/su13020744>
- Ramadhanti, W., & Setiawan, E. B. (2023). Topic Detection on Twitter Using Deep Learning Method with Feature Expansion GloVe. *Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika (JITEKI)*, 9(3), 780–792. <https://doi.org/10.26555/jiteki.v9i3.26736>
- Rasenda, Rini, Nova, S. (2024). Sistem Informasi Manajemen Pengklasifikasian Hadist Shahih Bukhari dan Muslim Menggunakan Algoritma Neural Network. *Sustainability in Action: Transformative Strategies in Management and Accounting*.
- Rizky Amalia Putri, N., & Terza Damaliana, A. (2024). Sentiment Analysis on Digital Korlantas POLRI Application Reviews Using the Distilbert Model. *Journal of Renewable Energy, Electrical, and Computer Engineering*, 4(x2), 83–89. <https://doi.org/10.29103/jreece.v4i2.17197>
- Santoso, S. P. J. (2019). Pengelompokan pada Hadits Menggunakan Naive Bayes Classifier. *E-Proceeding of Engineering*, 6(2), 9894.
- Sihombing, J. J., Arnita, A., Al Idrus, S. I., & Niska, D. Y. (2024). Implementation of text summarization on indonesian scientific articles using textrank algorithm with TF-IDF web-based. *Journal of Soft Computing Exploration*, 5(3), 310–319. <https://doi.org/10.52465/josce.v5i3.475>
- Suryaningrum, K. M. (2023). Comparison of the TF-IDF Method with the Count Vectorizer to Classify Hate Speech. *Engineering, MAthematics and Computer Science (EMACS) Journal*, 5(2), 79–83. <https://doi.org/10.21512/emacsjournal.v5i2.9978>
- Umi Hani. (2022). Buku Ajar Pengantar Studi Islam. In *Universitas Islam Kalimantan Muhammad Arsyad Al Banjary*.
- Xiao, L., Li, Q., Ma, Q., Shen, J., Yang, Y., & Li, D. (2024). Text classification algorithm of tourist attractions subcategories with modified TF-IDF and Word2Vec. *PloS One*, 19(10), e0305095. <https://doi.org/10.1371/journal.pone.0305095>