JOURNAL OF SOFT COMPUTING AND DATA MINING
e-ISSN: 2716-621X

JSCDM

Vol. 5 No. 2 (2024) 208-222
https://publisher.uthm.edu.my/ojs/index.php/jscdm

Optimizing Sentiment Analysis of Indonesian Texts: Enhancing Deep Learning Models with Genetic Algorithm-Based Feature Selection

Siti Mujilahwati[1,2], Noor Zuraidin Mohd Safar[1]*, Ku Muhammad Naim Ku Khalif[3,4], Nasyitah Ghazalli[5]

[1] *Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Persiaran Tun Dr. Ismail, 86400 Parit Raja, Johor, MALAYSIA*
[2] *Informatics Engineering, Faculty of Engineering, Universitas Islam Lamongan Jln. Veteran 53A, Lamongan, East Java 62211, INDONESIA*
[3] *Centre for Mathematical Sciences Universiti Malaysia Pahang Al-Sultan Abdullah, 26300, Kuantan, Pahang, MALAYSIA*
[4] *Centre for Artificial intelligence and Data Science, Universiti Malaysia Pahang Al-Sultan Abdullah, 26300, Kuantan, Pahang, MALAYSIA*
[5] *Thales UK, Reading RG2 6GF, UNITED KINGDOM*

*Corresponding Author: zuraidin@uthm.edu.my
DOI: https://doi.org/10.30880/jscdm.2024.05.02.016

Article Info

Received: 15 April 2024
Accepted: 11 August 2024
Available online: 18 December 2024

Keywords

Automatic text classification, feature selection, genetic algorithms, sentiment analysis, deep learning models

Abstract

Automatic text classification techniques are employed in a multitude of real-world applications, including the filtering of unsolicited messages, the analysis of sentiment, and the categorization of news items. The primary challenge in text representation is the high dimensionality, which can increase the complexity and risk of overfitting the model. To address this challenge, feature selection (FS) is conducted during the data pre-processing phase with the objective of enhancing the learning accuracy and efficiency of the model. This study examines the optimization of Indonesian text sentiment analysis through the integration of feature selection using a genetic algorithm (GA) with deep learning models. The application of GA for data dimensionality reduction from 41,140 to 20,769 features, coupled with fitness evaluation based on SVM, resulted in an observed increase in accuracy by 8.10% for SVM, 36.1% for Naïve Bayes, 7.82% for LSTM, 5.47% for DNN, and 6.25% for CNN. Of the three deep learning models, LSTM demonstrated the highest accuracy, at 91.41%, while also exhibiting a notable reduction in computation time, approaching 50%.

1. Introduction

Automatic text classification techniques can be used to address real-world problems such as spam filtering, sentiment analysis, and news classification [1], [2], [3], [4]. Text is typically represented in a term-document matrix with large dimensions. Challenges associated with large datasets include complex dimensional management and the risk of overfitting models [5], [6]. Feature Selection (FS) plays a crucial role in improving learning accuracy, removing irrelevant data, and reducing dimensions [7]. FS is the most significant process for classifying text, both manually and automatically [8]. In text classification, features are typically related to the representation of subgroups of words. Many features derived from text bodies may not be relevant to text classification tasks. Therefore, FS is typically performed during the pre-processing stage of data. This is because

irrelevant features in pre-processed results can decrease the efficiency and accuracy of classification models [7]. For this reason, FS for text classification has become a popular research topic in artificial intelligence and data mining conferences and journals [9], [10], [11], [12].

Strategies for selecting features in text classification can be grouped into four main categories [13], [14]. The filter method uses statistical metrics or simple rules to select the most informative features before the model learning process. For example, statistical methods such as Information Gain [5], [15], Chi-Square [5], [16], [17], or Mutual Information [18] are used to assess the relationship between features and category labels. The classification process selects the most informative features. The wrapper approach combines a machine-learning model with a feature selection process. It uses classification models like Naive Bayes or SVM to evaluate subsets of features. This iterative process determines the best subset that optimizes the model's performance [13], [18]. Third, embedded methods can be integrated into the learning process of the model itself. For instance, in learning algorithms such as regularized models, less important features can be eliminated or weighed down during the learning process. Additionally, the hybrid model combines multiple feature selection methods to take advantage of their respective benefits. For instance, a filter approach can be utilized to narrow down the selection of features that may be most informative. Then, the wrapper approach can be employed to optimize the subset of features selected by the filter [19].

One of the most popular hybrid models is the use of the Genetic Algorithm (GA) for character selection in text classification [20], [21], [22]. Since its introduction, the algorithm has been widely adopted and has undergone improvements and modifications [23], [24], [25], [26]. Previous studies show's that this algorithm gives better performance compared to previous feature selection models. This approach uses the principle of optimization inspired by evolutionary processes in nature. GA is used to solve optimization problems by mimicking natural selection processes, in which individuals (which are solutions or representations of features in the context of character selection in text classifications) develop and adapt from generation to generation. For these reasons, this study will propose GA as an algorithm for feature selection because this approach allows GA to effectively explore feature space and find the best feature combination that can improve the performance of text classification models. Efforts continue to improve the efficiency and accuracy of models by using GA as part of a hybrid approach to feature selection in sentiment analysis.

Deep learning is a part of machine learning that has become very popular in recent years for conducting case studies of sentimental analysis [27], [28], [29], [30]. Several studies of sentimental analysis on Indonesian-language texts have used the long short-term memory model (LSTM) to analyse the Merdeka Learning program (MBKM) [31], the deep neural network (DNN) to analyse opinions against public companies [32], and the convolutional neural network (CNN) algorithm to analyse cosmetic product reviews [33]. Previous research that studied deep learning models is still implemented standardly, with no optimization efforts with modelling such as adding feature selection. Therefore, this article will propose ways to optimize the performance of such models by adding a selection of features using GA with fitness evaluation based on the support vector machine (SVM) approach.

This article's primary contribution is to assess the efficacy of incorporating feature selection through a GA with fitness evaluation based on the support vector machine (SVM) approach into a deep learning model for sentiment analysis of Indonesian language text. This research not only attempts to implement a feature selection process on a single model architecture but also engages with three distinct types of deep learning models, namely LSTM, DNN, and CNN. By adopting this methodology, this article offers a more comprehensive insight into the impact of feature selection on model performance in understanding and analysing sentiment in Indonesian.

2. Related Work

Several studies have been conducted in the last three years on sentiment analysis of Indonesian texts using deep learning models. Amali et al. [34] utilized a CNN algorithm to classify text sentiment related to telecommunications service providers. They conducted experiments on text data from Twitter after pre-processing and weighting stages using Word2Vec. The Confusion Matrix results indicate that the CNN algorithm has an average accuracy of approximately 86.22%.

Other research by Pipin et al. [31] used LSTM to review the MBKM program, the result of this research produced an LSTM model that was trained from a dataset of 658 tweets with the best accuracy value of 80.42%. The MBKM program sentiment analysis of user tweets is dominated by feelings of "Bingung" at 39.51%, then by feelings of "Senang" at 16.26%, feelings of "Sedih" at 15.80%, feelings of "Marah" at 13.98%, feeling of "Takut" at 7.29% and felling of "Terkejut" at 7.14%.. Therefore, the study is crucial to the MBKM programme to ensure that its methods and implementation are clear, resulting in students feeling confident and have positive sentiments.

Kurniasari et al. [35] conducted research proposing the use of Recurrent Neural Network (RNN) is a LSTM to classify sentiment polarity in Indonesian sentences. They assessed the algorithm using a dataset of 25,000 travel site reviews, with an equal proportion of positive and negative classes. The proposed model achieved an accuracy level of 95.0% according to the evaluation results.

Hidayat et al [32] analyses sentiment through the Twitter Application Programming Interface (API) which was then carried out pre-processing (cleansing, case folding, formalization, stemming, and tokenization). The labelling process was carried out for 3902 records using the sentiment Strength Detection application. The model training stage is carried out using the DNN algorithm with variations in the number of hidden layers, node arrangement, and learning rate values. Experiments with a training and testing data proportion of 90:10 gave the best performance results. The model is composed of 3 hidden layers with the node arrangement for each layer in the model, namely 128, 256, and 128 nodes, and using a learning rate of 0.005, the model can produce an accuracy value of 88.72%.

As for research that tries to add feature selection to machine learning by Azhar et al. [36] confirmed the effectiveness of feature selection in handling Indonesian text for sentiment analysis. They evaluated coffee shop reviews using a GA as a feature selection method, which had a positive impact on sentiment analysis using the Naïve Bayes method. They evaluated coffee shop reviews using a GA as a feature selection method, which had a positive impact on sentiment analysis using the Naïve Bayes method. The accuracy rate increased by 4.4% or 0.044.

3. Methodology

The sentiment analysis process for COVID-19 treatment discussions in Indonesian Twitter comments involves data preparation by collecting 1,918 comments with positive and negative labels [37], followed by data cleaning and normalization to ensure accuracy and relevance. The data will be processed using TF-IDF to extract word importance and features are selected using genetic algorithms to enhance model performance. Deep learning techniques are applied for sentiment analysis classification, and the results are evaluated based on accuracy, precision, recall, and F1 score, visualized using confusion matrix images to depict true negatives, false negatives, false positives, and true positives.

3.1 Data Preparation

In order to prepare the data for analysis, it was necessary to collect comments that had been used in previous studies related to the handling of the novel coronavirus (Covid-19) in Indonesia [38], [39]. The data is accessible online and has been labeled, with 874 comments labeled positive and 1,044 comments labeled negative [37]. All texts in this dataset are in Indonesian. In previous studies, this data has been analyzed using a machine learning model for sentiment analysis. This study aims to conduct sentiment analysis using a deep learning model, which is expected to show better performance with a feature selection approach utilizing the same dataset.

3.2 Clean and Normalization

The second stage involves data cleaning, which is performed to clean, correct, and arrange the data to become more accurate, complete, consistent, and relevant for further analysis or use. Data cleansing processes are an important part of effective data management [40]. With cleaner and more orderly data, organizations can make better use of their data potential and make smarter decisions based on more reliable information. In this study, this process uses some processes in general:
Here is a detailed explanation of the processes mentioned in the paragraph:

1. Case Folding: This process converts all letters in the text to lowercase. For example, "COVID" and "covid" will be treated the same after case folding. This helps standardize words that are actually the same but written with different cases.
2. Removing Non-ASCII: This process removes characters that do not belong to the ASCII standard, such as special characters or symbols that are irrelevant for text analysis. For example, unique characters from other languages or emoticon symbols may be removed.
3. Removing URLs: All URLs or links within the text are removed since they are generally irrelevant for sentiment analysis or other text processing tasks.
4. Removing Quotations: Quotation marks (both single and double) are removed from the text. Quotation marks often do not contribute significantly to text analysis and can be ignored.
5. Removing Tags: This refers to the removal of HTML tags or other tags that may be present in the text. Tags like `@`, `#`, or other formatting tags are usually irrelevant for text analysis.
6. Removing Symbols: Symbols such as punctuation marks that are not important for sentiment analysis, like exclamation points, question marks, and others, are removed. These symbols do not add meaningful information in machine learning or deep learning models.
7. Removing Numbers: Numbers present in the text are also removed because they often do not provide important information in the context of sentiment analysis unless those numbers have specific meanings that need to be retained.

8. Fixing Character Duplications: This process corrects abnormal character repetitions in words. For example, "helloooo" can be changed to "hello". Excessive repetition often occurs in informal text such as social media comments.
9. Removing Stop Words : Stop words are common words like "dan," "atau," "tapi," which often do not have significant meaning in sentiment analysis. These words are removed so that the model can focus on more important words.
10. Convert Words: This process converts certain words to a more standardized form. For example, abbreviations or slang may be changed to their original forms, such as converting "nggak" to " tidak".
11. Stemming: use Sastrawi Stemming reduces words to their base form. For example, the word "mulailah" can be reduced to "mulai". This helps to standardize different word forms that have the same basic meaning.
12. Word Segmentation: This is the process of separating words in text that are written without spaces, especially in languages that often use space-free writing. In the context of the Indonesian language, this process may be more relevant for ensuring that words are correctly separated if there are unusual word combinations. Like “mulaihidupsehat” to “mulai hidup sehat”.

3.3 Data Extraction Using Term Frequency-Inverse Document Frequencies (TF-IDF)

TF-IDF is one of the methods used in text processing and information retrieval to extract and assess the importance of a word in a document in a corpus [41], [42]. To begin this extraction, tokenize the data, then compute:

3.3.1 Term Frequency (TF)

This is a measure of how often a word appears in a document. In TF representation, each word in a document is given a weight based on its frequency.

$$tf_i = \frac{n_i}{\sum_k n_k} \tag{1}$$

- tf_i : This represents the number of times a specific term t_i appears in a document. The tf is a measure of the frequency with which a given term appears in a document relative to the total number of terms in that document.
- n_i : This represents the number of occurrences of the term t_i in the document. This represents the unfiltered tally of the number of times the term t_i appears in the document.
- $\sum_k n_k$: This represents the total number of terms present within the document. This is the sum of the frequencies of all terms *k* in the document.

3.3.2 Inverse Document Frequency (IDF)

A measure of the uniqueness of a word in the entire corpus. Words that appear in many documents have a low IDF, whereas words that occur in a few documents have a high IDF.

$$idf_i = log \frac{|D|}{|\{d_j : t_i \in d_j\}|} \tag{2}$$

- idf_i: This represents of the term t_i. IDF measures how important a word is to a document within a corpus. If a word appears in many documents, it has lower importance (low IDF value), and if it appears in fewer documents, it has higher importance (high IDF value).
- $|D|$: This refers to the total number of documents in the corpus (or dataset). It represents the size of the entire collection of documents that analysing.
- $\{d_j : t_i \in d_j\}$: This denotes the set of documents d_j in which the term t_i appears. Essentially, this counts how many documents contain the term t_i.
- $|\{d_j : t_i \in d_j\}|$:This represents the number of documents that contain the term t_i. It is the denominator in the IDF calculation, which reduces the value of terms that appear in a multitude of documents.

3.3.3 TF-IDF Score

A combination of TF and IDF results in a score for each word in a document. This score shows how important the word in the document is in the context of the entire corpus.

$$tf - idf_i = tf_i \; x \; idf_i \tag{3}$$

J. of Soft Computing and Data Mining Vol. 5 No. 2 (2024) p. 208-222

- $tf - idf_i$: The TF-IDF score represents the relative importance or weight of a specific term t_i within a document.
- tf_i : This is the TF for the term t_i, which quantifies the frequency of occurrence of the term within the document in relation to the total number of terms.
- idf_i : This is a measure of the relative frequency of a term across a corpus of documents. It assigns greater weight to less common terms and less weight to more common terms.

3.4 Features Selection Using Genetic Algorithms (FSGA)

The primary purpose of feature selection is to select the most suitable feature group to study algorithms from the original feature set so that the most useful features remain and the useless features are removed, which does not affect the classification results. In this study, we used genetic algorithms. Fig. 2 is a framework for the genetic algorithm for the selection of features of the Indonesian language text. This algorithm can be used to identify and select a subset of features from relevant or critical data to improve model performance without reducing the accuracy of results on prediction or classification capabilities. In general, this algorithm is:

Algorithm : Genetic Algorithm for Feature Selection Optimize Using SVM
1: Division of train data and test data (previously processed data)
2: Initialize population Size n_{feat} and number of generations *size*
3: Evaluate chromosome arrangement or selected features using SVM
4: Choose parent pairs based on best fitness score
5: Perform Crossover and mutation to create new individuals
6: Combine selected features sets of the generation
7: **while** *iteration < size* **do**
8: Evaluate fitness score
9: **if** *fitness score* meets termination criteria **then**
10: Terminate iteration
11: **end if**
12: **end while**

Fig. 1 *Algorithm genetic algorithm design for feature selection using SVM model*

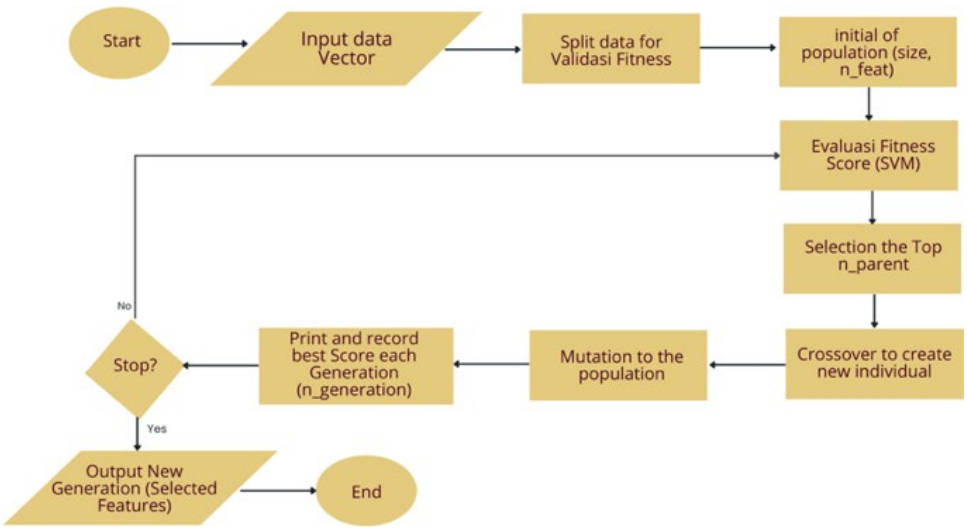


Fig. 2 *Framework feature selection using genetic algorithm*

Figure 2 depicts the methodology employed in the GA for feature selection. The process commences with the input of a data vector, which is then partitioned for the purpose of fitness validation. Subsequently, an initial population is constituted with a specific number of individuals and a predefined number of features to be selected. The fitness of each individual in the population is evaluated using a SVM Based on the aforementioned score, the most optimal individual is selected as the parent ('n_parent'), which is then subjected to a crossover process that yields a novel individual. Subsequently, mutation is conducted to introduce diversity into the population. The best score of each generation is recorded to monitor the progress of the algorithm. This process continues until a

stopping condition is reached, such as after a certain number of generations or when there is no significant improvement in the score. The outcome of this process is the selected features of the latest generation, which are ready to be used for further machine learning models. The process ends with the output of the selected features and the completion of the algorithm.

3.5 Analysis

In the fifth stage, sentiment analysis classification is performed using deep learning techniques. Here is a process plan for sentiment analysis.

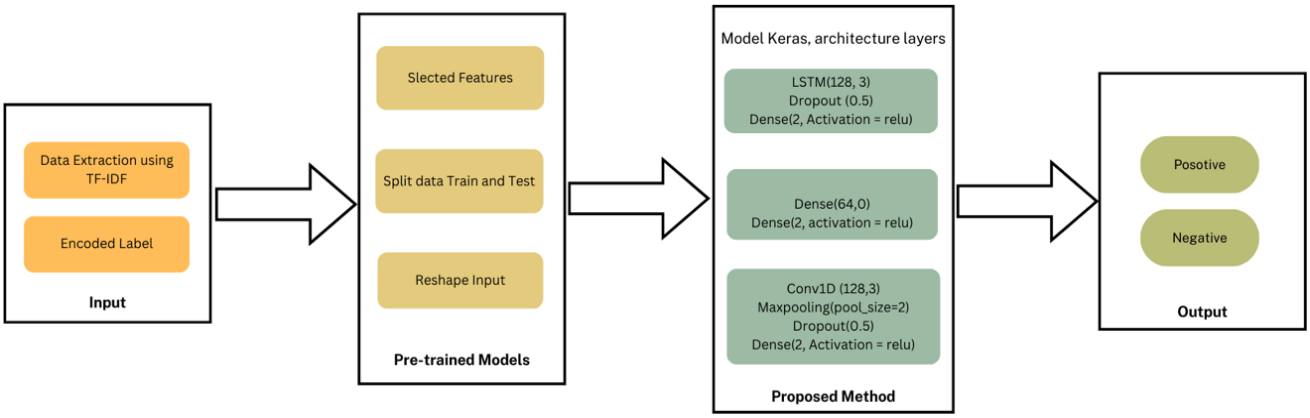


Fig. 2 *Framework sentiment analysis deep learning-based*

3.6 Evaluation and Validation

The proposed technique is evaluated based on accuracy, precision, recall, and F1 score. The evaluation parameter is given below:

3.6.1 Accuracy

Accuracy is a fundamental evaluation metric that indicates the proportion of correct predictions made by the model, relative to the total number of predictions. It provides a general measure of the model's overall effectiveness.

$$Accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \tag{4}$$

T_p : True Positives – instances where the model correctly identified a positive class.
T_n : True Negatives – instances where the model correctly identified a negative class.
F_p : False Positives – instances where the model incorrectly predicted a positive class.
F_n: False Negatives – instances where the model incorrectly predicted a negative class.

3.6.2 Precision

Precision, also referred to as positive predictive value, measures the accuracy of positive predictions made by the model. It is defined as the proportion of correctly predicted positive instances out of all instances that were predicted as positive.

$$Precision = \frac{T_p}{T_p + F_p} \tag{5}$$

T_p : True Positives – the number of correctly predicted positive instances.
F_p : False Positives – the number of negative instances incorrectly predicted as positive.

J. of Soft Computing and Data Mining Vol. 5 No. 2 (2024) p. 208-222

3.6.3 Recall

Recall, also known as sensitivity or true positive rate, is the ratio of correctly identified positive instances to the total number of actual positive instances. It provides insight into the model's ability to capture all relevant positive cases.

$$Recall = \frac{T_p}{T_p + T_n} \tag{6}$$

T_p : True Positives – the number of correctly predicted positive instances.
F_n : False Negatives – the number of actual positive instances incorrectly classified as negative.

3.6.4 F1-Score

The F1-Score is a metric that combines precision and recall, offering a single measure of performance that balances both. It is the harmonic mean of precision and recall, providing an effective measure of a model's ability to minimize both false positives and false negatives.

$$1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{7}$$

The F1-Score is particularly useful in scenarios where class distribution is imbalanced, as it provides a balanced view of precision and recall, avoiding biases that may arise when focusing on one metric alone.

3.6.5 Confusion Matrix

The confusion matrix is a fundamental instrument in the field of sentiment analysis, offering a comprehensive representation of a model's performance in categorizing sentiment. The confusion matrix provides a visual representation of the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) for each sentiment class, thus facilitating a comprehensive examination of the model's performance in predicting sentiment labels. By analyzing the confusion matrix, researchers can identify which sentiment classes are often confused with one another, thereby revealing specific areas where the model may require improvement. This information is vital for optimizing the sentiment analysis model, refining pre-processing techniques and enhancing feature selection methods in order to improve accuracy, precision, recall and F1-score. In essence, the confusion matrix represents a valuable diagnostic tool for the evaluation and interpretation of the effectiveness of sentiment analysis systems.

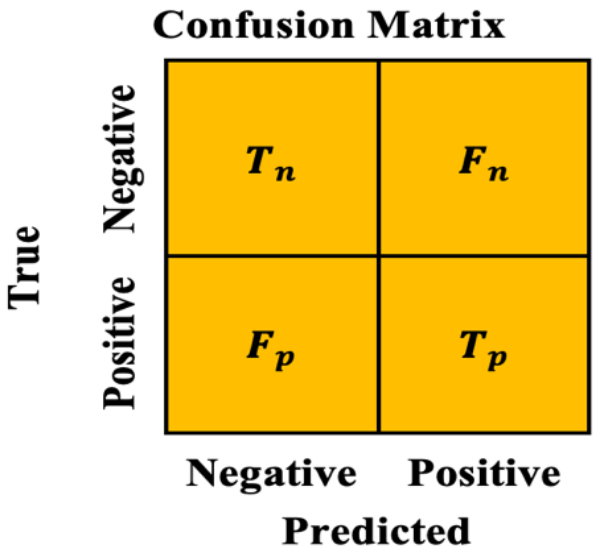


Fig. 3 *Confusion matrix design*

In this study, there are two classes of sentiment, namely positive and negative, so the confusion matrix can be illustrated in Fig. 3, where T_n is true negative, F_n is false negative, F_p is false positive, and T_p is true positive.

4. Result and Discussion

The results of the pre-processing stage have yielded 41,140 features with TF-IDF. These features will subsequently be selected to identify the most pertinent features for sentiment analysis. The genetic algorithm has been modified with a fitness function evaluation model that employs the SVM method. A GA is an optimization method inspired by the process of natural selection. In the context of feature selection, a GA is employed to identify the most pertinent combination of features, thereby enhancing the performance of a model. The following are the GA parameters utilized in this study.

Table 1 *Parameter GA*

Parameters	Values
Features (n_feat)	41.140
Features Selected (FS)	20.769
Size of population	150
Size of parent (n_parent)	30
Mutation rate	0.01
Size of generation (n_gen)	200
Model evaluation Fitness	Support Vector Machine (SVM)

A GA was employed for the purpose of feature selection, with the utilization of a number of predefined parameters. The initial dataset yielded 41,140 extracted features. Following the completion of the feature selection process, 20,769 features were identified as the most relevant. The process entailed a population of 150 individuals, with each generation comprising 30 parents selected based on their match rate. The mutation rate was set at 0.01, indicating that there was a 1% probability of a feature undergoing a change during the evolutionary process. The algorithm was executed for 200 generations to identify the optimal combination of features. In each evaluation phase, individual matches were evaluated using a SVM model to assess the efficacy of the selected subset of features in constructing an accurate predictive model. The evaluation yielded an accuracy rate of 91%. The subsequent Figure 4 illustrates the accuracy rate in each generation in relation to the individual match rate.

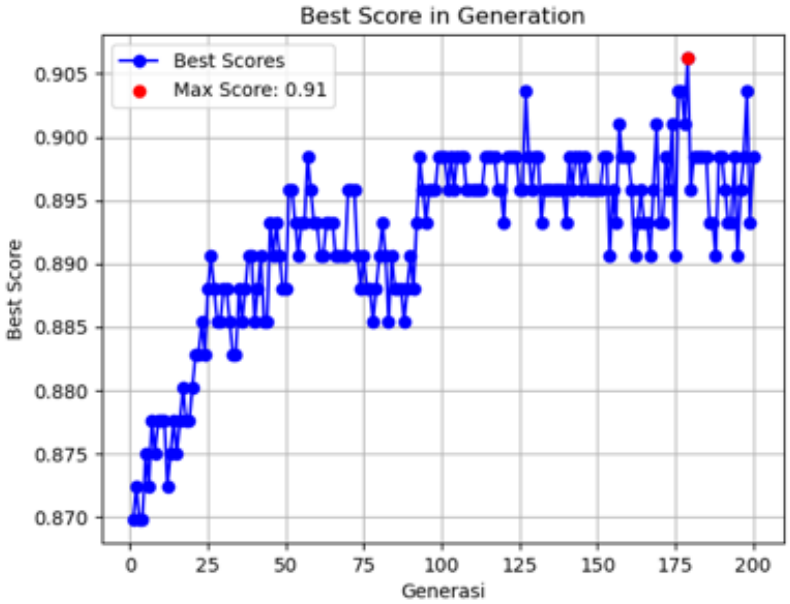


Fig. 4 *Best score 200 generation*

The sentiment analysis process is performed using a dataset comprising 20,222 features that have been selected for analysis. Subsequently, the data is divided into two distinct sets, with 80% allocated for training and 20% designated for testing. Before evaluating the efficacy of the feature selection methodology developed with deep learning models, we conducted preliminary tests on Machine Learning models to ascertain the comparative performance of this feature selection about previous studies utilizing the identical dataset. Here are the results of our tests.

J. of Soft Computing and Data Mining Vol. 5 No. 2 (2024) p. 208-222

Table 2 *Baseline model comparison from machine learning*

Author	Model	Features Selection	Accuracy %	Recall %	Precision %	F-Score %
Prasetyo, et. al[38]	SVM	-	82.00	82.01	82.24	81.84
Hadianti, et. al[39]	SVM	-	54	54	53	49
Hadianti, et. al[39]	Naïve Bayes	-	53	53	52	49
Propose	SVM	FSGA	90.10	89.38	90.56	89.80
	Naïve Bayes		89.32	88.21	90.53	88.88

The proposed approach, which employs FSGA, exhibits superior performance in all metrics (accuracy, recall, precision, and F-score) when compared to the methodologies utilized in previous studies, both for SVM and naïve Bayes models. This suggests that FSGA has been effective in markedly enhancing the model performance.

Subsequently, the performance of this FS was evaluated using three deep learning models: LSTM, CNN, and DNN. To illustrate the potential of feature selection in enhancing model performance, a comparative analysis was conducted with the deep learning models utilized in previous studies. The architectural details of the LSTM [31] , CNN [33] , and DNN [32] models are presented below.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 128)               21129728

 dropout (Dropout)           (None, 128)               0

 dense (Dense)               (None, 2)                 258

=================================================================
Total params: 21129986 (80.60 MB)
Trainable params: 21129986 (80.60 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
```

Fig. 5 *LSTM model architecture*

1. Layers:
   - LSTM Layer: The model starts with an LSTM layer, which is designed to handle sequential data and capture long-term dependencies. The output shape indicates that it outputs sequences of length 128.
   - Dropout Layer: A dropout layer follows, likely to prevent overfitting by randomly setting a fraction of input units to zero during training.
   - Dense Layer: The final dense (fully connected) layer with 2 units suggests that this is a binary classification task.
2. Total Parameters: 21,129,986 parameters, a significant number primarily due to the LSTM layer, which has a high number of trainable parameters.
3. Hyperparameters:
   - Number of Layers : The LSTM model has one LSTM layer, a dropout layer, and a dense layer, chosen to capture temporal dependencies in the data.
   - Units per Layer LSTM Model**:** 128 units in the LSTM layer, chosen to balance the need for capturing complexity without overly increasing the model size.
   - Activation Functions : Although not explicitly stated in the images, typical choices might include ReLU for intermediate layers and softmax for the final dense layer in a binary classification task.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_2 (Dense)             (None, 64)                2633024

 dense_3 (Dense)             (None, 2)                 130

=================================================================
Total params: 2633154 (10.04 MB)
Trainable params: 2633154 (10.04 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
```

Fig. 6 *DNN model architecture*

1. Layers :
   - Dense Layer 1: The model begins with a dense layer having 64 units, indicating a fully connected neural network structure.
   - Dense Layer 2: A second dense layer with 2 units, again for binary classification.
2. Total Parameters: 2,631,154 parameters, indicating a smaller model compared to the LSTM. This structure is typical for feedforward networks used in classification tasks.
3. Hyperparameters:
   - The DNN model uses two dense layers, typical for simple neural network architectures aimed at classification tasks.
   - DNN Model : 64 units in the first dense layer, likely chosen as a middle ground for model complexity.
   - Activation Functions : Although not explicitly stated in the images, typical choices might include ReLU for intermediate layers and softmax for the final dense layer in a binary classification task.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d_1 (Conv1D)           (None, 41138, 128)        512

 max_pooling1d_1 (MaxPoolin  (None, 20569, 128)        0
 g1D)

 dropout_3 (Dropout)         (None, 20569, 128)        0

 dense_11 (Dense)            (None, 20569, 2)          258

=================================================================
Total params: 770 (3.01 KB)
Trainable params: 770 (3.01 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
```

Fig. 7 *CNN model architecture*

1. Layers :
   - Conv1D Layer : The first layer is a 1D convolutional layer with 512 filters. This layer is designed to capture local patterns in the input data, which is useful for time series or sequential data.
   - MaxPooling Layer : A max-pooling layer follows, which down-samples the input representation to reduce its dimensionality.
   - Dropout Layer : Similar to the LSTM model, this dropout layer is used to reduce overfitting.
   - Dense Layer : The final dense layer with 2 units, indicating the output for binary classification.
2. Total Parameters **:** 770 parameters, making it the smallest model among the three, which suggests a more straightforward architecture.
3. Hyperparameters:
   - The CNN model has a single convolutional layer followed by pooling, dropout, and a dense layer, sufficient for capturing local patterns in the data.
   - CNN Model: 512 filters in the Conv1D layer, possibly chosen to allow the model to capture a wide range of features in the data.
   - Activation Functions : Although not explicitly stated in the images, typical choices might include ReLU for intermediate layers and softmax for the final dense layer in a binary classification task.

J. of Soft Computing and Data Mining Vol. 5 No. 2 (2024) p. 208-222

In the process of training and testing, the Adam optimization method was employed with beta 1 and 2 values of 0.001 and 0.999, respectively, and an epsilon value of 1e-8. The model was trained on a laptop with the following specifications: Intel Core i5 8265U 1.8 GHz (8 CPUs) and NVIDIA GeForce MX250 20 GB GPU. The model training process was conducted using a graphical processing unit (GPU). Subsequently, the evaluation is conducted using the Scikit-Learn library. The results of the baseline model performance evaluation, conducted with a combination of n-grams 1, epoch 20 for CNN models and 50 for LSTM and DNN models, batch size 64, and the optimal learning rate value, are presented in Table 3.

Table 3 *Baseline model results from deep learning*

Model	Learning Rate	Time Train Model / Second	Accuracy %	Recall %	Precision %	F-Score %
	0.0001	1058.803205013275	88.80	88.80	88.96	88.83
FSGA+	0.001	2530.803205013275	87.76	87.76	87.74	87.74
CNN	0.01	2516.487139225006	86.46	86.46	86.44	86.45
	0.1	2520.420850276947	85.42	85.42	85.47	85.44
	0.0001	25.886037588119507	89.58	89.58	89.58	89.56
FSGA+	0.001	26.285757541656494	87.24	87.24	87.22	87.23
DNN	0.01	25.910236597061157	85.42	85.42	85.51	85.44
	0.1	26.043387413024902	84.90	84.90	84.90	84.94
	0.0001	165.1459002494812	91.41	91.41	91.41	91.39
FSGA+	0.001	160.34843707084656	84.11	84.11	84.24	84.15
LSTM	0.01	161.27437591552734	82.29	82.29	82.71	82.36
	0.1	162.45469689369202	56.51	56.51	31.93	40.81

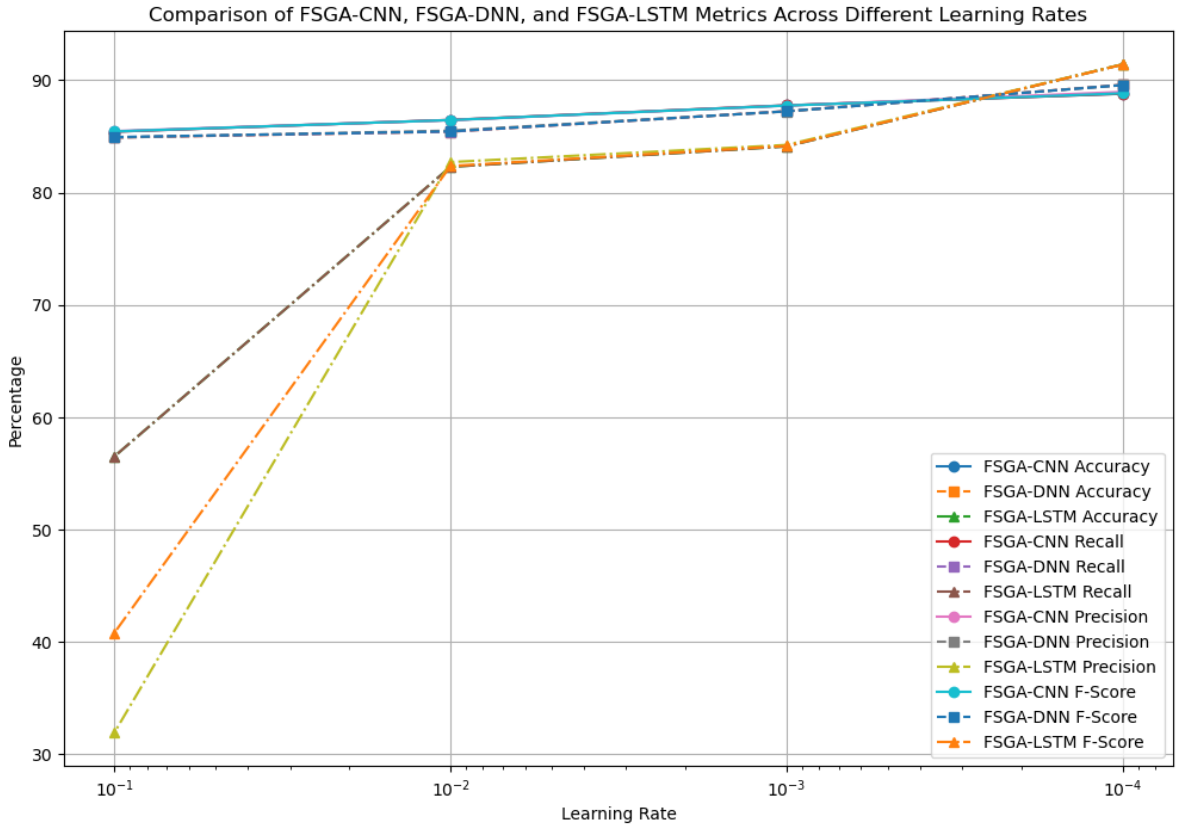
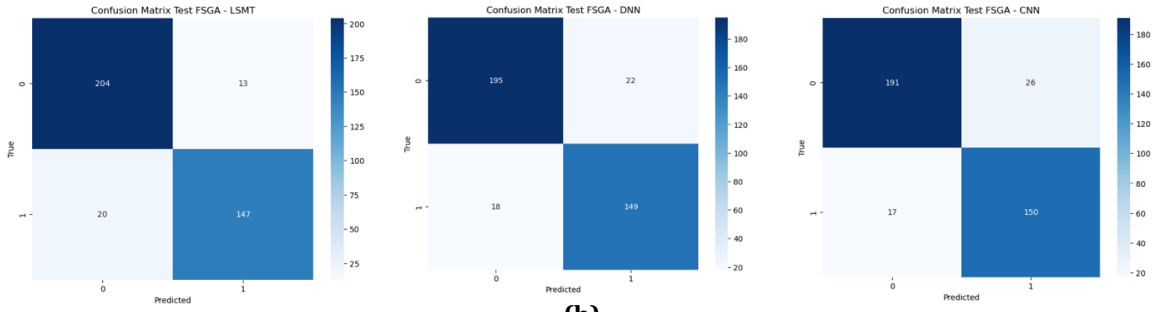


Fig. 8 *Compare performance baseline*

From the results of sentiment classification, the next stage is to analyze the results of sentiment classification in positive and negative categories. Here is a picture of the confusion matrix of the best scores from the test data.



(a) **(b)** **(c)**

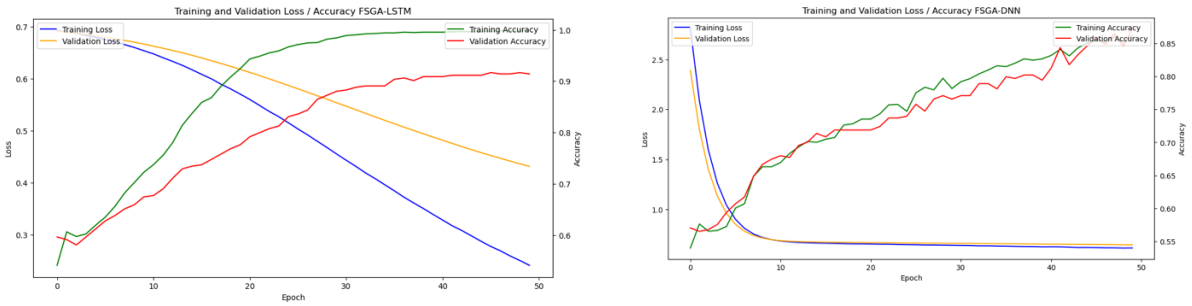
Fig. 9 *Confusion matrix deep learning-based (a) FSGA+LSTM model; (b) FSGA+DNN model; (c) FSGA+CNN model*

From the results of experiments using different learning rate optimizations, the next step is to choose the model with the best accuracy from all trials conducted. The best accuracy results will then be compared with previous research models adopted and tested with the data in this study. The following are the results of a performance comparison based on best accuracy.

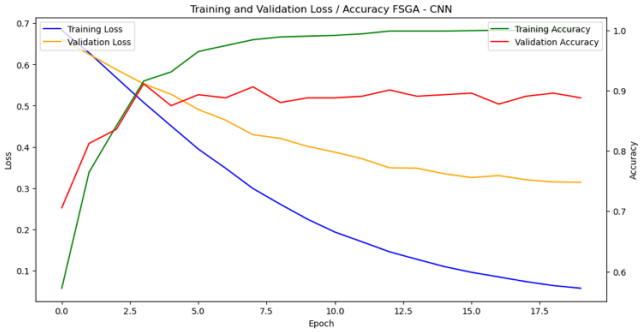
Table 4 *Baseline performance comparison*

Author	Model	Time Train Model / Second	Accuracy %	Recall %	Precision %	FScore %
Pipin et al. n.d (2022)[31]	LSTM	282.87886142730713	83.59	82.73	83.79	83.08
Hidayat *et al.* (2021)[32]	DNN	51.54164433479309	84.11	83.74	83.88	83.81
Hidayat *et al.* (2022)[33]	CNN	2059.862309432134	82.55	82.29	82.26	82.27
Proposed	FSGA+LSTM	165.1459002494812	91.41	91.41	91.41	91.39
	FSGA+DNN	25.886037588119507	89.58	89.58	89.58	89.56
	FSGA+CNN	1058.803205013275	88.80	88.80	88.96	88.83

As evidenced by the trials conducted with varying learning rates, as detailed in Table 4, the learning rate of 0.0001 yielded the optimal evaluation value among the three models employed. The models in question are the FSGA + LSTM, FSGA + DNN, and FSGA + CNN models. The LSTM model exhibited a 7.82% increase in accuracy, while the DNN and CNN models demonstrated 5.47% and 6.25% gains, respectively. Furthermore, in addition to the aforementioned improvements in accuracy and dimensionality reduction, computational efficiency was also achieved, with the average training time reduced by 50% in comparison to the model that did not undergo feature selection. The subsequent graph offers a visual representation of the outcomes yielded by the three proposed models.



(a) **(b)**



(c)

Fig. 10 *Plot training and validation (a) FSGA+LSTM model;(b) FSGA+DNN model; (c) FSGA+CNN model*

In order to prevent overfitting, a number of techniques are employed in the models. The dropout rate is set at 0.5 for LSTM and DNN models, whereas CNN models employ a dropout rate of 0.2. Furthermore, regularization (L2) is applied to LSTM and DNN models with a value of 0.01. These measures have been demonstrated to be effective, as evidenced by the training and validation loss graphs. The LSTM and DNN models demonstrate robust performance, exhibiting a reduction in loss and an increase in accuracy on both training and validation data without significant overfitting. In contrast, the CNN model employs dropout solely for the purpose of attaining optimal performance. Of the three models that were tested, the LSTM model demonstrated the most accurate results.

5. Conclusion

The research demonstrates that the optimization of Indonesian text sentiment analysis with feature selection in the deep learning model is highly effective. The efficacy of feature selection using a genetic algorithm with the incorporation of an SVM-based fitness evaluation model is demonstrated by the reduction in data dimension from 41,140 to 20,769. With a reduction in dimensions, the evaluation results demonstrate an improvement in accuracy, both in machine learning and deep learning models. In the evaluation using the same dataset, there was an increase in accuracy in the machine learning model, with the SVM method demonstrating an 8.10% improvement and the Naïve Bayes method exhibiting a 36.1% enhancement. In contrast, the deep learning model, which employed the same architectural configuration as in the preceding study but utilized a distinct dataset, demonstrated an increase in accuracy of 7.82% for the LSTM model, 5.47% for the DNN model, and 6.25% for the CNN model. The LSTM model exhibited optimal performance, with an accuracy value of 91.41%. In addition to reducing dimensions and increasing accuracy, this feature selection can also reduce training computation time. Based on the analysis, approximately 50% of the computation time can be reduced with a relatively small amount of data. The present study will be extended in the future to include further experimentation with feature selection on additional models, with an investigation of different layers and hyperparameters. Moreover, testing will be conducted on a variety of cases, including Indonesian texts with topics that are likely to become trending in the future.

Acknowledgment

This research was supported by Universiti Tun Hussein Onn Malaysia (UTHM) Tier 1 vote (Q397). The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of interest

The authors declare no conflict of interest regarding the publication of the paper.

References

[1] S. Kaddoura, G. Chandrasekaran, D. E. Popescu, and J. H. Duraisamy, "A systematic literature review on spam content detection and classification," *PeerJ Comput Sci*, vol. 8, 2022, doi: 10.7717/PEERJ-CS.830.

[2] S. D. Mahajan, "News Classification Using Machine Learning," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 9, no. 5, 2021, doi: 10.17762/ijritcc.v9i5.5464.

[3] E. Rahmanian, "Fake news: a classification proposal and a future research agenda," *Spanish Journal of Marketing - ESIC*, vol. 27, no. 1, 2023, doi: 10.1108/SJME-09-2021-0170.

[4] L. Yang, Y. Li, J. Wang, and R. S. Sherratt, "Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2969854.

[5] S. Kumar, V. B. Singh, and S. K. Muttoo, "Bug Report Classification by Selecting Relevant Features Using Chi Square, Information Gain and Latent Semantic Analysis," in *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), ICRITO 2021*, 2021. doi: 10.1109/ICRITO51393.2021.9596496.

[6] A. Y. Rahman, D. A. Aziz, A. L. Hananto, S. Sulaiman, and C. Zonyfar, "INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION journal homepage : www.joiv.org/index.php/joiv INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION Classification of Tempeh Maturity Using Decision Tree and Three Texture Features." [Online]. Available: www.joiv.org/index.php/joiv

[7] M. Zaffar *et al.*, "A hybrid feature selection framework for predicting students performance," *Computers, Materials and Continua*, vol. 70, no. 1, 2021, doi: 10.32604/cmc.2022.018295.

[8] A. Madasu and S. Elango, "Efficient feature selection techniques for sentiment analysis," *Multimed Tools Appl*, vol. 79, no. 9–10, 2020, doi: 10.1007/s11042-019-08409-z.

[9] A. O. Bajeh, B. O. Funso, and F. E. Usman-Hamza, "Performance Analysis of Particle Swarm Optimization for Feature Selection," *FUOYE Journal of Engineering and Technology*, vol. 4, no. 1, 2019, doi: 10.46792/fuoyejet.v4i1.364.

[10] Z. K. Zadeh and M. A. Z. Chahooki, "An Effective Method of Feature Selection in Persian Text for Improving the Accuracy of Detecting Request in Persian Messages on Telegram," *Journal of Information Systems and Telecommunication*, vol. 8, no. 32, 2020, doi: 10.29252/jist.8.32.249.

[11] S. S. Abdulkhaliq and A. M. Darwesh, "Sentiment Analysis Using Hybrid Feature Selection Techniques," *UHD Journal of Science and Technology*, vol. 4, no. 1, 2020, doi: 10.21928/uhdjst.v4n1y2020.pp29-40.

[12] D. Endalie and G. Haile, "Hybrid Feature Selection for Amharic News Document Classification," *Math Probl Eng*, vol. 2021, 2021, doi: 10.1155/2021/5516262.

[13] J. T. Pintas, L. A. F. Fernandes, and A. C. B. Garcia, "Feature selection methods for text classification: a systematic literature review," *Artif Intell Rev*, vol. 54, no. 8, 2021, doi: 10.1007/s10462-021-09970-6.

[14] M. B. Imani, M. R. Keyvanpour, and R. Azmi, "A novel embedded feature selection method: A comparative study in the application of text categorization," *Applied Artificial Intelligence*, vol. 27, no. 5, 2013, doi: 10.1080/08839514.2013.774211.

[15] N. M. G. D. Purnamasari, M. Ali Fauzi, Indriati, and L. S. Dewi, "Cyberbullying identification in twitter using support vector machine and information gain based feature selection," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 3, 2020, doi: 10.11591/ijeecs.v18.i3.pp1494-1500.

[16] S. Bahassine, A. Madani, M. Al-Sarem, and M. Kissi, "Feature selection using an improved Chi-square for Arabic text classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 32, no. 2, 2020, doi: 10.1016/j.jksuci.2018.05.010.

[17] A. Yoga Pratama *et al.*, "Analisis Sentimen Media Sosial Twitter Dengan Algoritma K-Nearest Neighbor Dan Seleksi Fitur Chi-Square (Kasus Omnibus Law Cipta Kerja)," *Jurnal Sains Komputer & Informatika (J-SAKTI*, vol. 5, no. 2, 2021.

[18] Y. B. Wah, N. Ibrahim, H. A. Hamid, S. Abdul-Rahman, and S. Fong, "Feature selection methods: Case of filter and wrapper approaches for maximising classification accuracy," *Pertanika J Sci Technol*, vol. 26, no. 1, 2018.

[19] O. M. Alyasiri, Y. N. Cheah, and A. K. Abasi, "Hybrid Filter-Wrapper Text Feature Selection Technique for Text Classification," in *International Conference on Communication and Information Technology, ICICT 2021*, 2021. doi: 10.1109/ICICT52195.2021.9567898.

[20] N. Bidi and Z. Elberrichi, "Feature selection for text classification using genetic algorithms," in *2016 8th International Conference on Modelling, Identification and Control (ICMIC)*, 2016, pp. 806–810. doi: 10.1109/ICMIC.2016.7804223.

[21] S. S. Hong, W. Lee, and M. M. Han, "The feature selection method based on genetic algorithm for efficient of text clustering and text classification," *International Journal of Advances in Soft Computing and its Applications*, vol. 7, no. 1, 2015.

[22] F. Iqbal *et al.*, "A Hybrid Framework for Sentiment Analysis Using Genetic Algorithm Based Feature Reduction," *IEEE Access*, vol. 7, pp. 14637–14652, 2019, doi: 10.1109/ACCESS.2019.2892852.

[23] A. Rasool, R. Tao, M. Kamyab, and S. Hayat, "GAWA–A Feature Selection Method for Hybrid Sentiment Classification," *IEEE Access*, vol. 8, pp. 191850–191861, 2020, doi: 10.1109/ACCESS.2020.3030642.

[24] O. Abayomi-Alli *et al.*, "An Improved Feature Selection Method for Short Text Classification," *J Phys Conf Ser*, vol. 1235, no. 1, p. 012021, Jun. 2019, doi: 10.1088/1742-6596/1235/1/012021.

[25] G. J. Ansari, J. H. Shah, M. C. Q. Farias, M. Sharif, N. Qadeer, and H. U. Khan, "An Optimized Feature Selection Technique in Diversified Natural Scene Text for Classification Using Genetic Algorithm," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3071169.
[26] S. Belkarkor, I. Hafidi, and M. Nachaoui, "Feature Selection for Text Classification Using Genetic Algorithm," in *Lecture Notes in Networks and Systems*, vol. 656 LNNS, 2023, pp. 69–80. doi: 10.1007/978-3-031-29313-9_7.
[27] P. Cen, K. Zhang, and D. Zheng, "Sentiment Analysis Using Deep Learning Approach," vol. 2, no. 1, pp. 17–27, 2020, doi: 10.32604/jai.2020.010132.
[28] A. Soufan, "Deep Learning for Sentiment Analysis of Arabic Text," 2019.
[29] O. J. Ying, M. Mun, N. Ramli, and U. U. Sheikh, "Sentiment analysis of informal Malay tweets with deep learning," vol. 9, no. 2, pp. 212–220, 2020, doi: 10.11591/ijai.v9.i2.pp212-220.
[30] D. A. Kristiyanti, A. Al Kaafi, E. Purwaningsih, E. Nurelasari, and B. Nisa, "Deep learning for Twitter sentiment analysis about the pros and cons of Covid-19 vaccines in Indonesia," in *AIP Conference Proceedings*, 2023, p. 030025. doi: 10.1063/5.0128686.
[31] S. J. Pipin, H. Kurniawan, | Jurnal, and S. Mikroskil, "Analisis Sentimen Kebijakan MBKM Berdasarkan Opini Masyarakat di Twitter Menggunakan LSTM," *OKTOBER 2022 IJCCS*, vol. 23, pp. 1–5.
[32] E. Y. Hidayat, R. W. Hardiansyah, and A. Affandy, "Analisis Sentimen Twitter untuk Menilai Opini Terhadap Perusahaan Publik Menggunakan Algoritma Deep Neural Network," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 7, no. 2, pp. 108–118, Sep. 2021, doi: 10.25077/teknosi.v7i2.2021.108-118.
[33] E. Y. Hidayat and D. Handayani, "Penerapan 1D-CNN untuk Analisis Sentimen Ulasan Produk Kosmetik Berdasar Female Daily Review," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 8, no. 3, pp. 153–163, Jan. 2023, doi: 10.25077/teknosi.v8i3.2022.153-163.
[34] Z. Amalia, M. Irfan, D. S. Maylawati, A. Wahana, W. B. Zulfikar, and M. A. Ramdhani, "Sentiment Analysis of the Use of Telecommunication Providers on Twitter Social Media using Convolutional Neural Network," in *2022 IEEE 8th International Conference on Computing, Engineering and Design (ICCED)*, IEEE, Jul. 2022, pp. 1–6. doi: 10.1109/ICCED56140.2022.10010357.
[35] L. Kurniasari and A. Setyanto, "Sentiment analysis using recurrent neural network-lstm in bahasa Indonesia," *Journal of Engineering Science and Technology*, vol. 15, no. 5, 2020.
[36] N. Azhar, P. P. Adikara, and S. Adinugroho, "Analisis Sentimen Ulasan Kedai Kopi Menggunakan Metode Naive Bayes dengan Seleksi Fitur Algoritme Genetika," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 8, no. 3, 2021, doi: 10.25126/jtiik.2021834436.
[37] Wildan Fariq A, "Sentimen Bahasa," Jun. 2022. Accessed: Sep. 29, 2023. [Online]. Available: https://github.com/onpilot/sentimen-bahasa/tree/master/dataset/txt
[38] P. H. Prastyo, A. S. Sumi, A. W. Dian, and A. E. Permanasari, "Tweets Responding to the Indonesian Government's Handling of COVID-19: Sentiment Analysis Using SVM with Normalized Poly Kernel," *Journal of Information Systems Engineering and Business Intelligence*, vol. 6, no. 2, 2020, doi: 10.20473/jisebi.6.2.112-122.
[39] S. Hadianti *et al.*, "ANALISIS SENTIMENT COVID-19 DI TWITTER MENGGUNAKAN METODE NAIVE BAYES DAN SVM," *Jurnal Teknologi Informasi)*, vol. 6, no. 1, [Online]. Available: www.Kaggle.com.
[40] F. Ridzuan and W. M. N. Wan Zainon, "A review on data cleansing methods for big data," in *Procedia Computer Science*, 2019. doi: 10.1016/j.procs.2019.11.177.
[41] S. M. Permataning Tyas, B. S. Rintyarna, and W. Suharso, "The Impact of Feature Extraction to Naïve Bayes Based Sentiment Analysis on Review Dataset of Indihome Services," *Digital Zone: Jurnal Teknologi Informasi dan Komunikasi*, vol. 13, no. 1, 2022, doi: 10.31849/digitalzone.v13i1.9158.
[42] S. Fransiska and A. Irham Gufroni, "Sentiment Analysis Provider by.U on Google Play Store Reviews with TF-IDF and Support Vector Machine (SVM) Method," *Scientific Journal of Informatics*, vol. 7, no. 2, 2020.