BAREKENG: Journal of Mathematics and Its Applications
September 2024 Volume 18 Issue 3 Page 1791–1802
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol18iss3pp1791-1802

COMPARISON OF SUPPORT VECTOR MACHINE BASED ON FASTTEXT WITHOUT AND WITH FIREFLY OPTIMIZATION PARAMETERS FOR DISASTER SENTIMENT ANALYSIS IN INDONESIA

Fadilah Amirul Adhel[1], Sri Astuti Thamrin[2*], Siswanto Siswanto[3]

[1,2,3]*Statistics Study Program, Department of Statistics, Faculty of Mathematics and Natural Sciences*
Universitas Hasanuddin
Jl. Perintis Kemerdekaan Km. 10 Tamalanrea, Makassar, 90245, Indonesia

Corresponding author's e-mail: ***** ***tuti@unhas.ac.id***

Article History:

Received: 9th February 2024
Revised: 3rd April 2024
Accepted: 4th July 2024
Published: 1st September 2024

Keywords:

Sentiment Analysis;
Fasttext;
Classification;
Optimization Firefly;
Support Vector Machine.

ABSTRACT

Sentiment analysis is a process for analyzing opinions, sentiments, assessments, and emotions from someone's statements regarding a domain or is also a process for entering and processing data in the form of text. Support vector machine (SVM) is a supervised machine learning technique that functions as a separator of two classes of data. SVM aims to obtain numerical vectors using fasttext. SVM cannot choose appropriate parameters so the use of parameters is not optimal. To obtain optimal parameters with better classification results, firefly optimization was carried out. This research compares the fasttext-based SVM method without and with firefly optimization parameters using data from tweets with the keyword "Indonesian disaster" which was crawled using the Twitter application. The results of this research obtained 128 dimensions that form the weight of each word. This means that each word is represented in a 128-dimensional vector space. The evaluation of the SVM classification model with and without firefly optimization provides an accuracy of 89.1% and 61.3% respectively. This shows that the SVM classification method with firefly optimization provides quite good classification performance compared to the SVM model without optimization.

How to cite this article:

F. A. Adhel, S. A. Thamrin and Siswanto., "COMPARISON OF SUPPORT VECTOR MACHINE BASED ON FASTTEXT WITHOUT AND WITH FIREFLY OPTIMIZATION PARAMETERS FOR DISASTER SENTIMENT ANALYSIS IN INDONESIA", *BAREKENG: J. Math. & App.*, vol. 18, iss. 3, pp. 1791-1802, September, 2024.

Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*

Research Article • Open Access

1. INTRODUCTION

Indonesia's geographical location which is at the confluence of three active plates, namely Indo-Australia, Eurasia, and the Pacific, makes this country a high level of vulnerability to geological and hydro-climatological disasters. The impact of these catastrophic events varied widely, including damage, losses, and even casualties [1]. This illustrates the need for disaster preparedness in disaster management in Indonesia by focusing on historical factors of past events. Indonesia is a disaster-prone country. Natural disasters that often occur in Indonesia include earthquakes, tsunamis, floods, landslides, volcanic eruptions, and others. The event prompted social media users or disaster management agencies to upload information on the disaster situation at the disaster site [2].

Advances in information and communication technology make information related to natural disasters more quickly spread. The media plays an important role in natural disasters. Information about the type of disaster, when the disaster occurred, the location of the disaster, the impact, and the needs of victims of natural disasters can all be recorded and conveyed through news media [3]. Social media provides data that is used as a source of information to respond to natural disasters in a region. Research that has been conducted by Wu, has provided a new concept by using social media as a social network [4].

Twitter is one of the well-known social media used by the public to convey the feelings of its users. Through posts on Twitter, people can share and get the latest information about disasters that are happening. Using data from social media such as Twitter, public opinion and opinion regarding disasters in Indonesia can be analyzed by grouping them into negative and positive categories through sentiment analysis. Sentiment analysis is the process of extracting, understanding, and processing data in the form of unstructured text automatically to obtain sentiment information contained in opinions or opinion sentences [5]. The application of sentiment analysis using machine learning methods and several methods that are often used [6]. On text data, a feature extraction is required to convert text data into numeric data. There are several types of word embedding extraction features such as word2vec, glove, and fasttext.

The extraction feature used in this study is fasttext which is a word embedding model developed by Facebook. Fasttext is a development of the word2vec word embedding model. One of the main advantages of fasttext is its ability to account for sub-word information in words by dividing words into sub-word units (such as syllables or characters), fasttext can better represent words not found in dictionaries or non-common words. This is useful for languages with related words and complex morphologies. Fasttext is able to achieve outstanding performance in word representation and sentence classification by utilizing character-level information, especially in the case of rare words [7]. After the extraction feature process, a model will be created using the Support vector machine (SVM) method.

Sentiment analysis often involves high-dimensional data. For example, when converting text to a feature vector through processes such as word embedding or tf-idf, there are a great many features (proportional to the size of the vocabulary). SVM has good performance in processing such high-dimensional data. SVM is a supervised machine learning technique used to analyze data for classification. SVM operates by classifying data points from the training data set. The purpose of the SVM algorithm is to find a hyperplane in n-dimensional space (number of features) that can classify data points clearly[8].

There are two cases of classification, linear and non-linear. In the linear case, the data are perfectly separated, so the classifier only needs to find a linear line as a separator between classes. To do this, a method is needed that can increase the dimensions of the data so that the classifier can separate the data perfectly. The problem of nonlinear classification can be solved by implementing kernel functions to support vector machines. There are various kernels available for the parameter selection process, including radial base (RBF), sigmoid, and polynomial functions. In this study, the RBF kernel will be used because the RBF kernel can handle linear separation in high-dimensional nonlinear input data, such as linear separation in text classification [9].

SVM cannot select the appropriate parameters, resulting in less than the optimal parameter usage. The use of appropriate parameters is expected to improve the accuracy of SVM. Fireflies are a meta-heuristic approach within a swarm intelligence swarm (SI), which uses a method in which a group of fireflies engages in social behavior and communication through light on their tails. Optimization is the process of finding the best solution to a particular problem of interest, and this search process can be done using multiple agents. This essentially forms an ever-evolving system of agents [10] [11].

Research that has been conducted by [12] regarding sentiment analysis of handling covid-19 using the long short-term memory method on Twitter social media using fasttext, words vectored using fasttext, the aim is to convert string data types into vectors. Research that has been conducted by [13] on SVM parameter optimization based on firefly optimization on film opinion data, by finding the optimal SVM parameter value based on accuracy. The results showed that firefly optimization was able to get the optimal combination of SVM parameters based on accuracy, so there was no need for trial and error to get this value. This is evidenced by the evaluation results in the highest accuracy of 87.84%. In another study, the Support Vector Machine-Firefly algorithm film opinion data classification was carried out [14]. The research results show that firefly optimization can help SVM to obtain parameter combinations that are suitable based on accuracy with a shorter execution time and get an accuracy value of 87.15%. From the description above, this research aims to obtain Indonesian disaster sentiment classification results using SVM firefly optimization with fasttext extraction and model evaluation results using accuracy.

2. RESEARCH METHODS

2.1.Sentiment Analysis

Sentiment analysis is the process of analyzing opinions, sentiments, judgments, and emotions from one's statements about a particular field, or it is also the process of extracting and processing data in text form automatically to understand opinion tendencies. Objections, whether they tend to have positive or negative opinions [15] [16].

2.2. Text Preprocessing

Text preprocessing is the process that occurs before data is selected or filtered to obtain important words while retaining the thematic features of the text. In addition to acquiring important words in the topic of the text, text preprocessing can also effectively reduce high-dimensional features and reduce noise, thereby saving data processing time and improving accuracy. Here are the stages of text preprocessing:

1. Case Folding

Case folding is a process in text preprocessing that is carried out to homogenize characters in data. The case folding process is the process of converting all letters into lowercase. In this process, the A-Z characters contained in the data are converted into a-z characters [8].

2. Filtering

Filtering is a step to eliminate illegal characters in documents such as punctuation, symbols, numbers, html, and mentions. The process of removing illegal characters can be called filtering.

3. Stemming

Stemming is the process of mapping word variations to basic forms. The Stemming process is done by removing affixes, both prefixes and suffixes from a word to get its root word. Stemming commonly used in Indonesian texts uses a literary stemmer library.

4. Stopword Removal

Stopword removal is the stage at which important words are extracted and words that are considered unimportant are removed. How to get rid of unimportant words is called stopword removal. The removal of stopwords aims to eliminate words that appear frequently but do not contribute to the data analysis process. The removal of stopwords seeks to reduce the dimensions of the data and speed up computation time [17].

5. Tokenization

Tokenization is the stage of breaking a sentence into parts called tokens. A token is considered a form of a word, phrase, or meaningful element. This stage also removes certain characters such as punctuation marks and converts all tokens to lowercase. Tokenization can break up a document into words, phrases, symbols, or other elements that have meaning. In the tokenization process, the review text data is broken down into tokens consisting of one meaningful word and stored in a word array [16].

2.3. Word Embedding

Word embedding is a term used for the technique of converting a word into a vector or array consisting of a collection of numbers. When creating a machine learning model that accepts input text, of course, machine learning cannot directly accept the raw text it has, the word must first be converted into numbers with a reference to a word dictionary. Usually, if you do not use word embedding, each word will be converted into a number in integer form according to the position of the number in the dictionary [18]. The architecture of the skip-gram network can be seen in Figure 1 as follows:

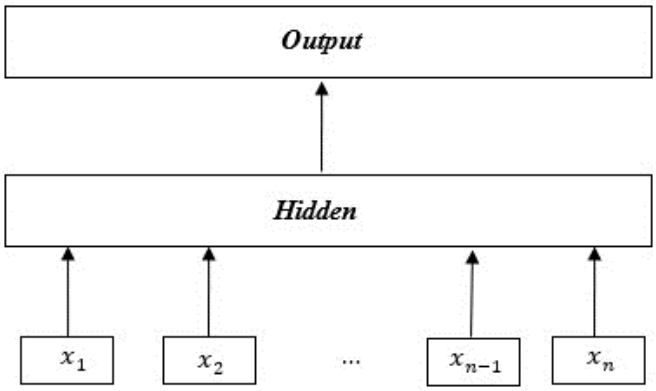

Figure 1. Fasttext illustration

$$v(w) = z(w) + \sum_{i=1}^{n} x_i \quad (1)$$

with:
$v(w)$: representation of the word w
$z(w)$: feature vector for word w
x_i : feature vector for words $n-gram, i = 1, 2, \cdots, n$

The use of fasttext in this study was done to convert words into vectors. The advantage of fasttext is it can handle words that have never been encountered before because the process utilizes the sub-word of each word [19] [20].

2.4. Support Vector Machine

Support vector machine (SVM) is a method developed by Boser, Guyon, and Vapnik that was first presented in 1992. The special concept of SVM is to minimize empirical misclassification and maximize geometric margins [21], [22]. Therefore, SVM is called maximum margin classifier. Suppose the data in the training dataset is denoted as $x_i \in \Re^q$ while the labels of each class are expressed as linear $y_i \in \{-1, +1\}$ models in general used in the SVM method to produce the hyperplane is written in Equation (2) as follows:

$$y = f(x) = w^T x_i + b, i = 1,2 \ldots, n \quad (2)$$

The data class label y_i is estimated using a weight vector value perpendicular to the hyperplane $(\boldsymbol{w})$, in the form of a vector measuring $p \times 1$ plus error or bias (b) which is a scalar. The main goal of the SVM method is to find a hyperplane that can separate two classes with a maximum separation distance [23]. The optimization problem for determining the best SVM hyperplane is written in Equation (3):

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i \cdot \boldsymbol{x}_j) \quad (3)$$

with constraints:

$$\sum_{j=1}^{n} \alpha_i y_i = 0, \forall_i \text{ with } \alpha_i > 0 \quad (4)$$

to calculate the value of w and b, you can use the equation from the soft margin SVM by changing the value x to $\Phi(x)$, so the Equation (5) is obtained as follows:

$$b = \frac{1}{s}\sum_{j=1}^{s}\left(y_k - w \cdot \Phi(x_k)\right) \quad (5)$$

Data that is a support vector $x_k = [x_{k1}, x_{k2}, \dots, x_{kq}]$ and a value α containing support vector (α_k) will be used to calculate the value of b, where s is the number of pedestal vectors and $K(x_i, x_j)$ is a radial basis function (RBF) whose parameter is γ (gamma) which is written in **Equation (6)** as follows:

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right) \quad (6)$$

2.5. Firefly Optimization

Firefly optimization is population-based metaheuristic optimization and is included in the Nature Inspired Algorithms (NIA) group. Firefly Optimization first developed **[24]**. Firefly optimization simulates the social behavior of fireflies in nature to solve optimization problems. The main advantage of Firefly optimization is its effectiveness in the non-linear problem. Firefly optimization is proven to be effective in handling the complexity of non-linear problem **[25]**. Based on Firefly optimization, the objective function is derived from the following **Equation (7)**:

$$\max f(x), x = (x_1, \dots \dots x_d)^T \quad (7)$$

Two formulations in firefly optimization are the change in light intensity and the attractiveness. The brightness of the firefly will be determined by the objective function, and the attractiveness is proportional to the brightness so that every two fireflies blink, the darker firefly will move towards the lighter firefly. The light intensity of fireflies is influenced by objective functions. The level of light intensity for the problem of minimizing a firefly x can be seen in the following **Equation (8)**:

$$I(x) = \frac{1}{f(x)} \quad (8)$$

In addition, the intensity of light will decrease from the source because it is absorbed by media, such as air. So that attractiveness (β) can be determined by distance r, this can be seen in **Equation (9)** as follows:

$$\beta = \beta_0 e^{-\gamma r^2} \quad (9)$$

Fireflies that have a higher light intensity tend to attract other, weaker fireflies, and this can also be affected by the distance between them. Therefore, the distance between fireflies is one of the factors influencing the movement behavior and exchange of information between fireflies in this optimization. The Euclidean distance is formulated in **Equation (10)** as follows:

$$r_{ij} = \| x_i - x_j \| = \sqrt{\sum_{k=1}^{n}\left(x_i^k - x_j^k\right)^2} \quad (10)$$

Fireflies approaching other fireflies with higher light intensities will try to get closer further to that firefly in an attempt to improve its solution. This may involve changing positions (movements) within the search space to approach a better solution. Then the position of the firefly or the solution of the firefly will change according to the formula in **Equation (11)** below:

$$x_i' = x_i + \beta_0 e^{-\gamma r_{ij}^2}\left(x_j - x_i\right) + a\left(rand - \frac{1}{2}\right) \quad (11)$$

2.6. Stage of Analysis

The stages of data analysis carried out in this research are as follows:

1. Crawling tweet data from Twitter using the API and then saving it in the form of a CSV file.
2. Labeling data using Microsoft Excel software for positive and negative sentiment with the following criteria for each class:

   Positive sentiment criteria

a. Contains words or phrases of support, hopes, and prayers from the community.
b. Contains praise or respect for an individual or organization certain organizations.

Negative sentiment criteria

a. Contains complaints and dissatisfaction felt by the community related to disasters.
b. Contains harsh or sarcastic words, insults, or hate speech
c. Contains expressions of intimidation against certain groups.
d. Contains expressions of discrimination against certain groups.

3. Preprocessing text data on tweets using case folding, filtering, stemming, stopword removal, and tokenizing methods.
4. Convert words into vectors using word embedding with the fasttext method.
5. Split the dataset or divide the data into training data and test data with a ratio of 80% training data and 20% test data.
6. Build an SVM model with training data
7. Perform sentiment classification on test data using a support vector machine model
8. Evaluate model accuracy using a confusion matrix
9. Optimize parameters C and γ
10. Compare the accuracy obtained from SVM and SVM optimized using the firefly optimization method.

3. RESULTS AND DISCUSSION

3.1 Text Preprocessing Indonesia's Disaster

The data in this study is primary data obtained from xplore.pustakadata.id application. This application is able to crawl using an application programming interface (API) with the keyword 'Indonesian disaster' in Indonesian-language tweets uploaded between January 01 and February 28, 2023. The data obtained was 1653 tweets in text form. The data structure used in this research is data in the form of tweet text consisting of predictor variables (message) and response variables (sentiment).

a) Case Folding

Case folding is one of the preprocessing stages to change all letters in a document or sentence to lowercase. Case folding is used to simplify data processing. Not all data is consistent in the use of capital letters. The following **Table 1** results from the case folding stages are as follows:

Table 1. Case Folding Stage Results

No	Before *Case Folding*	After *Case Folding*
1	Tadi sore gempa lagi, walaupun kecil mau tidur aja nggak tenang Tidur ini emosi kebangun tuh kepala pusing banget Ini aja udah pusing, mana hari pertama pula Paksa tidur malah tambah pusing anjing gempa terus ðŸ˜ðŸ˜ðŸ˜	tadi sore gempa lagi, walaupun kecil mau tidur aja nggak tenang tidur ini emosi kebangun tuh kepala pusing banget ini aja udah pusing, mana hari pertama pula paksa tidur malah tambah pusing anjing gempa terus Ã°ÂŸÂ˜ÂÃ°ÂŸÂ˜Â-Ã°ÂŸÂ˜Â
⋮	⋮	⋮
1653	Ada â€œbadaiâ€• baru gempa, su mode berserah tadi subuh ðŸ˜¹ minta perlindungan tuhan	ada Ã¢Â€Âœbadai Ã¢Â€Â• baru gempa, su mode berserah tadi subuh Ã°ÂŸÂ˜Â¹ minta perlindungan tuhan

Based on **Table 1**, the first text initially contains complaints about the earthquake that occurred, with expressions of emotions such as confusion and annoyance. After case folding, the text still maintains the same meaning and expression, only the letters have been changed to lowercase. Number two initial text mentions the uncertain weather situation with a new earthquake. Religious expressions are also expressed by asking for God's protection. After case folding, the text retains the same meaning, only the letters have been changed to lowercase. Thus, the results of case folding produce text that is more uniform in terms of the use of lowercase letters, making further processing and analysis easier.

b) Filtering

Filtering in the preprocessing stage of text data refers to the process of removing or modifying less relevant text elements, such as punctuation symbols, special characters, common words that do not have special meaning (stopwords), or URL links. The results of the filtering stages can be seen in **Table 2** below.

Table 2. Filtering Stage Result

No	Before Filtering	After Filtering
1	tadi sore gempa lagi, walaupun kecil mau tidur aja nggak tenang tidur ini emosi kebangun tuh kepala pusing banget ini aja udah pusing, mana hari pertama pula paksa tidur malah tambah pusing anjing gempa terus Ã°ÂŸÂ˜ÂÃ°ÂŸÂ˜ÂÃ°ÂŸÂ˜Â	tadi sore gempa lagi walaupun kecil mau tidur aja nggak tenang tidur ini emosi kebangun tuh kepala pusing banget ini aja udah pusing mana hari pertama pula paksa tidur malah tambah pusing anjing gempa terus.
⋮	⋮	⋮
1653	ada Ã¢Â€Âœbadai Ã¢Â€Â• baru gempa, su mode berserah tadi subuh Ã°ÂŸÂ˜Â¹ minta perlindungan tuhan	ada badai baru gempa su mode berserah tadi subuh minta perlindungan tuhan

In **Table 2**, symbols such as emojis have been removed, and extra spaces have been removed between words. The text has been cleaned of irrelevant elements. Thus, the filtering results in text are cleaner and more relevant for further processing, because unnecessary elements have been removed or modified.

c) Stemming

Stemming is a process that involves removing affixes or suffixes from words so that only the root word remains. For example, words like "read," "read aloud," and "read" can all be simplified to the root word "read" through stemming techniques. The results of the Stemming stages can be seen in **Table 3** as follows:

Table 3. Stemming Stage Result

No	Before *Stemming*	After *Stemming*
1	tadi sore gempa lagi walaupun kecil mau tidur aja nggak tenang tidur ini emosi kebangun tuh kepala pusing banget ini aja udah pusing mana hari pertama pula paksa tidur malah tambah pusing anjing gempa terus	tadi sore gempa lagi walaupun kecil mau tidur aja nggak tenang tidur ini emosi bangun tuh kepala pusing banget ini aja udah pusing mana hari pertama pula paksa tidur malah tambah pusing anjing gempa terus
⋮	⋮	⋮
1653	ada badai baru gempa su mode berserah tadi subuh minta perlindungan tuhan	ada badai baru gempa su mode serah tadi subuh minta lindung tuhan

Based on **Table 3**, The words in the first tweet have been simplified by removing auxiliary words such as "ke", so that only basic words remain, such as "bangun" rather than "kebangun", while for the second tweet, the words in the text have been simplified by removing unimportant prefixes or suffixes, such as "berserah" becomes "serah" and "perlindungan" becomes "lindung", so that only the basic words remain.

d) Stopwords Removal

If you omit stopwords in Indonesian text, then words that have no significance in the context of text analysis will be omitted. The results of the stopwords removal stages can be seen in **Table 4** below:

Table 4. Stopwords Removal Stage Result

No	Before Stopwords Removal	After Stopwords Removal
1	tadi sore gempa lagi walaupun kecil mau tidur aja nggak tenang tidur ini emosi bangun tuh kepala pusing banget ini aja udah pusing mana hari pertama pula paksa tidur malah tambah pusing anjing gempa terus	tadi sore gempa walaupun kecil mau tidur aja tenang tidur emosi bangun tuh kepala pusing banget aja udah pusing mana hari pertama paksa tidur malah tambah pusing anjing gempa terus
⋮	⋮	⋮
1653	ada badai baru gempa su mode serah tadi subuh minta lindung tuhan	badai baru gempa su mode serah tadi subuh minta lindung tuhan

Based on **Table 4**, the first tweet words such as "lagi", "ini", "tuh", "aja", "udah", "malah", "pula", "mana", and "terus" which are stopwords have been removed from the text. This results in text that focuses more on important words that have meaning in the context of the analysis. A second tweet such as "ada" was removed, as the word did not contribute significantly to the overall meaning of the text. The result is text that is simpler and easier to understand.

e) Tokenizing

Tokenizing is commonly used in natural language processing and computer text processing to simplify data analysis and modeling. The main purpose of tokenizing is to convert raw text into separate units called tokens. The results of the tokenizing stage can be seen in **Table 5** as follows:

Table 5. Tokenizing Stage Result

No	Before Tokenizing	After Tokenizing
1	tadi sore gempa walaupun kecil mau tidur aja tenang tidur emosi bangun tuh kepala pusing banget aja udah pusing mana hari pertama paksa tidur malah tambah pusing anjing gempa terus	tadi,sore,gempa,walaupun,kecil,mau,tidur,aja,tenang,tidur,emosi,bangun,tuh,kepala,pusing,banget,aja,udah,pusing,mana,hari,pertama,paksa,tidur,malah,tambah,pusing,anjing,gempa,terus
⋮	⋮	⋮
1653	badai baru gempa su mode serah tadi subuh minta lindung tuhan	badai,baru,gempa,su,mode,serah,tadi,subuh,minta,lindung,tuhan

Based on **Table 5** The initial text has been divided into separate tokens by commas (,). Each word or phrase in the text becomes its own token, making it easier to analyze each unit separately.

3.2 Prediction Class Determination

Determining the prediction class with SVM is carried out using **Equation (12)**:

$$f(x) = \sum_{i=1}^{n} a_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b \quad (12)$$

Data prediction classes can be defined. If the value of $f(x) \geq 0$, the data will be predicted as a positive class. Whereas if the value $f(x) < 0$, the data will be predicted as a negative class. The following are the prediction results for the first data in the test data set:

$$f(x) = \sum_{i=1}^{n} a_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + 0.93762$$
$$= \sum_{i=1}^{1322} a_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + 0.93762$$
$$= \big((0.1)(1)K(x_1.x_1) + \cdots + (0)(1)K(x_{1322}.x_1)\big) + 0.93762$$
$$= (0.02034 + \cdots + 0 + 0.93762)$$
$$= 1.06974$$

Based on the prediction results, 1.06974 is obtained and it meets the conditions $f(x) \geq 0$, then the data will be classified into positive class. The results of the classification stages are presented in **Table 6**.

Table 6. SVM Prediction Stage Result

No	Message	Sentiment	Predictions
1	trauma gempa lumayan yaa moga semua beri selamat	1	1
2	weh ada gempa sana stay safe semua nya	1	1
3	gempa terus weh tolong dong dunia lindung tuhan	1	1
4	besok galungan gue overthink gempa udah x	0	1
⋮	⋮	⋮	⋮
331	distribusi bantu masyarkat banjir ternate moga dampak manfaat semua korban	1	1

Table 6 Messages with positive sentiment (value 1) in the "Sentiment" column have been correctly predicted by the SVM model, where the "Predictions" value is also 1. However, there is one misprediction in the fourth row. A message with a negative sentiment (value 0) in the "Sentiment" column is predicted as a positive sentiment (value 1) by the SVM model. Out of 331 processed messages, the majority are correctly predicted as positive sentiment (value 1), which is consistent with the text trends in this dataset. This indicates that the SVM model has provided good results in predicting sentiment in text, although there are some prediction errors. Further evaluation may be needed to improve the model's performance, such as parameter adjustments or the use of additional features.

3.3 Evaluation Results of Confusion Matrix SVM

Confusion matrices are generally used in machine learning and pattern recognition and are used as a performance evaluation method in various types of classification models such as logistic regression, naive Bayes, decision trees, and so on. By using the confusion matrix, more detailed information about model performance is obtained, and evaluated the strengths and weaknesses of the model. Based on the results of SVM predictions on data that matches the actual labels, the following are obtained:

1. True Positive (TP) was 203
2. False Positive (FP) was 128
3. True Negative (TN) is 0
4. False Negative (FN) is 0

The SVM prediction results obtained using the confusion matrix are in **Figure 2**.

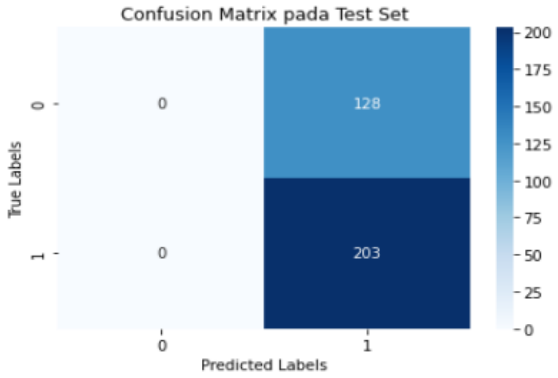

Figure 2. Confusion Matrix SVM

Based on **Figure 2**, 203 data had positive sentiment and were predicted correctly (true positive), 0 data had negative sentiment and were predicted correctly (true negative), 0 data should have positive sentiment but were incorrectly predicted as negative sentiment (false negative), and 128 data should have negative sentiment but it was wrongly predicted as positive sentiment (false positive). One of the causes of prediction errors in false negative and false positive data can be caused by removing stop words in the text data preprocessing process. From the results of the confusion matrix above, performance measurements can be carried out on the SVM model. The following are the performance measurement results on the SVM model using accuracy:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{230 + 0}{203 + 0 + 128 + 0} = 0.613$$

The classification results obtained using SVM showed an accuracy of 61.3%, which indicates the percentage of data with positive and negative sentiment that was predicted correctly.

3.4 Evaluation Results of Confusion Matrix SVM Firefly Optimization

Firefly optimization is a metaheuristic optimization inspired by the behavior of fireflies in attracting their mates by producing light. This optimization is used to find optimal solutions to complex optimization problems. In the context of SVM, firefly optimization can be used to find the best parameters that maximize classification accuracy on training data. In this case, the parameters to be optimized are the values of parameters C and γ in the RBF kernel, initiation $C = 0.1$ and $\gamma = 0.1$.

SVM firefly optimization will choose the parameters that produce the best accuracy and use them in the next iteration. The results obtained after 10 iterations are parameters $C = 8.320$ and $\gamma = 14.523$ which produces the best accuracy of 0.89 on the tested dataset. This means that this parameter is the best parameter that can be used in the SVM model to achieve the best performance on the dataset that has been tested.

The optimization process becomes more efficient and faster compared to other parameter search methods. In this example, SVM firefly optimization managed to find the best parameters after going through 10 iterations. The results of the SVM prediction using firefly optimization on the data corresponding to the actual label are obtained as follows:

1. True Positive (TP) as much as 187
2. False Positive (FP) as much as 20
3. True Negative (TN) as much as 108
4. False Negative (FN) as many as 16

Thus, the results of the Support vector machine prediction using the confusion matrix in **Figure 4** were obtained.

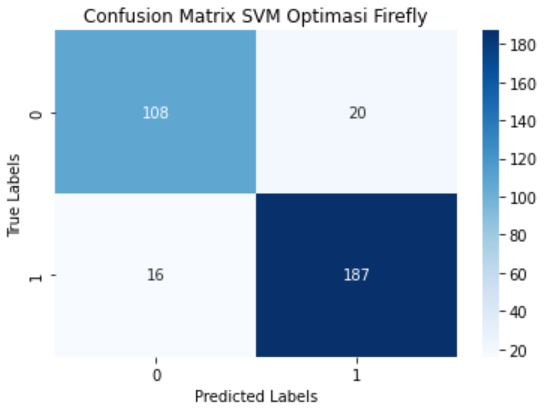

Figure 3. Confusion Matrix SVM Firefly Optimization

Based on **Figure 3**, 187 data have positive sentiment predicted correctly (true positive), 108 data have negative sentiment predicted correctly (true negative), there are 16 data that should have positive sentiment but predicted as negative sentiment (false negative), and 20 data that should have negative sentiment but predicted as positive sentiment (false positive). The following are the performance measure results on the Firefly optimization SVM model using accuracy:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{187 + 108}{197 + 108 + 20 + 16} = 0.891$$

The accuracy results obtained were 89.1%, which shows that the SVM model using the Firefly optimization SVM parameters is able to classify correctly on the dataset that has been tested.

4. CONCLUSIONS

The results of classifying disaster sentiment in Indonesia using SVM without firefly optimization were 203 true positives, 128 false positives, 0 true negatives, and 0 false negatives with a model evaluation accuracy of 61.3%. Meanwhile, the classification results using SVM firefly optimization with fasttext feature extraction obtained 187 true positives, 20 false positives, 108 true negatives, and 16 false negatives with a model evaluation accuracy of 89.1%. Based on the model evaluation results, the accuracy of the SVM model was 61.3% and the SVM model optimized with firefly optimization was 89.1% on 331 test data. It can be seen that the accuracy of the SVM model optimized with firefly optimization was higher than the SVM model that was not optimized. This shows that the optimization process using firefly optimization can improve the performance of the SVM model in predicting test data. Therefore, the SVM model that has been optimized with firefly optimization can be a better choice for use in predicting new data.

REFERENCES

[1] M. R. Pahleviannur, "Edukasi Sadar Bencana Melalui Sosialisasi Kebencanaan Sebagai Upaya Peningkatan Pengetahuan Siswa Terhadap Mitigasi Bencana," *Jurnal Pendidikan Ilmu Sosial*, vol. 29, no. 1, pp. 49–55, Jun. 2019, doi: 10.23917/jpis.v29i1.8203.

[2] N. Giarsyani, "Komparasi Algoritma Machine Learning dan Deep Learning untuk Named Entity Recognition: Studi Kasus Data Kebencanaan," *Indonesian Journal of Applied Informatics*, vol. 4, no. 2, p. 138, Aug. 2020, doi: 10.20961/ijai.v4i2.41317.

[3] Kasbawati, "Kontrol Optimal Upaya Pencegahan Infeksi Virus Flu Burung H5N1 dalam Populasi Burung dan Manusia," *Jurnal Matematika, Statistika dan Komputasi*, vol. 8, no. 1, pp. 12–24, 2011.

[4] C.-H. Wu, "Social Sensor: An Analysis Tool for Social Media," *International Journal of Electronic Commerce Studies*, vol. 7, no. 1, pp. 77–94, Jun. 2016, doi: 10.7903/ijecs.1411.

[5] F. Fitriana, E. Utami, and H. Al Fatta, "Analisis Sentimen Opini Terhadap Vaksin Covid - 19 pada Media Sosial Twitter Menggunakan Support Vector Machine dan Naive Bayes," *Jurnal Komtika (Komputasi dan Informatika)*, vol. 5, no. 1, pp. 19–25, Jul. 2021, doi: 10.31603/komtika.v5i1.5185.

[6] P. Arsi and R. Waluyo, "Analisis Sentimen Wacana Pemindahan Ibu Kota Indonesia Menggunakan Algoritma Support Vector Machine (SVM)," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 8, no. 1, pp. 147–156, 2021, doi: 10.25126/jtiik.202183944.

[7] A. R. W. Rapsanjani and E. Junianto, "Implementasi Probabilistic Neural Network Dan Word Embedding Untuk Analisis Sentimen Vaksin Sinovac," *Jurnal Responsif: Riset Sains dan Informatika*, vol. 3, no. 2, pp. 233–242, Aug. 2021, doi: 10.51977/jti.v3i2.588.

[8] M. Awad and R. Khanna, "Support Vector Machines for Classification," in *Efficient Learning Machines*, Berkeley, CA: Apress, 2015, pp. 39–66. doi: 10.1007/978-1-4302-5990-9_3.

[9] A. Noor and M. Islam, "Sentiment Analysis for Women's E-commerce Reviews using Machine Learning Algorithms," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, IEEE, Jul. 2019, pp. 1–6. doi: 10.1109/ICCCNT45670.2019.8944436.

[10] Styawati, Andi Nurkholis, Zaenal Abidin, and Heni Sulistiani, "Optimasi Parameter Support Vector Machine Berbasis Algoritma Firefly Pada Data Opini Film," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 5, pp. 904–910, Oct. 2021, doi: 10.29207/resti.v5i5.3380.

[11] K. Devi Thangavel, U. Seerengasamy, S. Palaniappan, and R. Sekar, "Prediction of factors for Controlling of Green House Farming with Fuzzy based multiclass Support Vector Machine," *Alexandria Engineering Journal*, vol. 62, pp. 279–289, Jan. 2023, doi: 10.1016/j.aej.2022.07.016.

[12] I. Pakpahan and Jasman Pardede, "Analisis Sentimen Penanganan Covid-19 Menggunakan Metode Long Short-Term Memory Pada Media Sosial Twitter," *Jurnal Publikasi Teknik Informatika*, vol. 2, no. 1, pp. 12–25, Jan. 2023, doi: 10.55606/jupti.v1i1.767.

[13] R. Parlika, S. Ilham Pradika, A. M. Hakim, and R. N. M. Kholilul, "Analisis Sentimen Twitter Terhadap Bitcoin dan Cryptocurrency Berbasis Python TextBlob," 2020. [Online]. Available: https://t.co/QaUW3P2TKc

[14] R. Nazrul and N. C. Aminuallah, "Klasifikasi Data Opini Film Algoritma Support Vector Machine-Firefly," *Portal Data*, vol. 2, no. 5, pp. 1-12, 2022.

[15] W. Chen, Z. Xu, X. Zheng, Q. Yu, and Y. Luo, "Research on Sentiment Classification of Online Travel Review Text," *Applied Sciences*, vol. 10, no. 15, p. 5275, Jul. 2020, doi: 10.3390/app10155275.

[16] S. Mulyani, S. A. Thamrin, and S. Siswanto, "Analisis Sentimen Masyarakat Pada Kebijakan Vaksinasi Covid-19 Di Twitter Menggunakan Metode Mesin Vektor Pendukung Dengan Kernel Radial Basis Function Berbasis Fitur Leksikon," *Jambura Journal of Probability and Statistics*, vol. 3, no. 2, pp. 110–119, Nov. 2022, doi: 10.34312/jjps.v3i2.16663.

[17] S. Symeonidis, D. Effrosynidis, and A. Arampatzis, "A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis," *Expert Syst Appl*, vol. 110, pp. 298–310, Nov. 2018, doi: 10.1016/j.eswa.2018.06.022.

[18] R. Yang, M. Burns, A. De Hoedt, S. Williams, S. Freedland, and Z. Klaassen, "MP29-09 Identification of Prostate Cancer Metastatic Disease for Different Risk Groups Based On Fasttext Word Embedding And Supervised Learning," *Journal of Urology*, vol. 209, no. Supplement 4, Apr. 2023, doi: 10.1097/JU.0000000000003257.09.

[19] M. D. Rahman, A. Djunaidy, and F. Mahananto, "Penerapan Weighted Word Embedding pada Pengklasifikasian Teks Berbasis Recurrent Neural Network untuk Layanan Pengaduan Perusahaan Transportasi," *Jurnal Sains dan Seni ITS*, vol. 10, no. 1, Aug. 2021, doi: 10.12962/j23373520.v10i1.56145.

[20] S. Chatterjee, L. Evenss, P. Bhattacharyya, and J. Mondal, "LSJSP at SemEval-2023 Task 2: FTBC: A FastText based framework with pre-trained BERT for NER," in *Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2023, pp. 1254–1259. doi: 10.18653/v1/2023.semeval-1.174.

[21] H. Li, "Support Vector Machine," in *Machine Learning Methods*, Singapore: Springer Nature Singapore, 2024, pp. 127–177. doi: 10.1007/978-981-99-3917-6_7.

[22] N. Rezki, S. A. Thamrin, and S. Siswanto, "Sentiment Analysis of Merdeka Belajar Kampus Merdeka Policy Using Support Vector Machine with Word2vec," *BAREKENG: Jurnal Ilmu Matematika dan Terapan*, vol. 17, no. 1, pp. 0481–0486, Apr. 2023, doi: 10.30598/barekengvol17iss1pp0481-0486.

[23] D. A. Sani and M. Z. Sarwani, "Koreksi Jawaban Esai Berdasarkan Persamaan Makna Menggunakan Fasttext dan Algoritma Backpropagation," *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 11, no. 2, pp. 92–111, Aug. 2022, doi: 10.23887/janapati.v11i2.49192.

[24] S. K. Sunori, D. K. Singh, A. Mittal, S. Maurya, U. Mamodiya, and P. K. Juneja, "Rainfall Classification using Support Vector Machine," in *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, IEEE, Nov. 2021, pp. 433–437. doi: 10.1109/I-SMAC52330.2021.9640773.

[25] H. Swapnarekha, J. Nayak, H. S. Behera, P. B. Dash, and D. Pelusi, "An optimistic firefly algorithm-based deep learning approach for sentiment analysis of COVID-19 tweets," *Mathematical Biosciences and Engineering*, vol. 20, no. 2, pp. 2382–2407, 2022, doi: 10.3934/mbe.2023112.