ICIC Express Letters
Volume **16**, Number **8**, August **2022**

ICIC International ©2022 ISSN 1881-803X
pp. **811–818**

IDENTIFYING HATE SPEECH IN BAHASA INDONESIA WITH LEXICON-BASED FEATURES AND SYNONYM-BASED QUERY EXPANSION

Atmaja Wikandiputra[1], Afiahayati[2,*] and Vincent Michael Sutanto[3]

[1]Mindimedia
Jl. Nuansa Timur X No. 9 Jimbaran, Bali 80361, Indonesia
gstmdjiwaatmaja@gmail.com

[2]Department of Computer Science and Electronics
Faculty of Mathematics and Natural Sciences
Universitas Gadjah Mada
Jalan Bulaksumur Ged Pusat UGM Lt 3, Special Region of Yogyakarta 55281, Indonesia
*Corresponding author: afia@ugm.ac.id

[3]Division of Information Science
Nara Institute of Science and Technology
8916-5 Takayama-cho, Ikoma, Nara 630-0192, Japan
vincent.michael_sutanto.vg6@is.naist.jp

Received August 2021; accepted November 2021

Abstract. *Freedom of social media users who are not controlled in giving opinions can make it easier for users to attack certain people, objects, or environments with hateful language or commonly called hate speech. According to the Indonesia Criminal Investigation Police, 80% of cybercrimes reported were expressions of hatred. Preventive actions taken by Facebook & Twitter are deemed ineffective because checking hate speech is still manually through user reports. In this study, we used a machine learning algorithm, which is Support Vector Machine (SVM), to identify whether a speech is considered as hate speech or not. We combined the SVM with the Lexicon-based Features and Synonym-based Query Expansion method. The models were trained and evaluated by calculating Accuracy, Precision, Recall, and F-measure. This study shows that the use of the Synonym-based Query Expansion method can improve the performance of the SVM model with Lexicon-based as its feature.*

Keywords: Hate speech, Classification, Support Vector Machine, Lexicon-based, Synonym

1. **Introduction.** Hate speech is a form of speech that targets certain group characteristics, such as ethnic origin, religion or beliefs, gender, or sexual orientation [1]. Since the Indonesian National Police issued a Circular Letter No. SE/6/X/2015 regarding the handling of hate speech in October 2015, according to Purnomo Hadi Suseno, Head of Unit 5 of Criminal Investigation Body of the Indonesian National Police, 80% of cybercrimes reported were hate speech [2]. The increase in the percentage of hate speech outperformed reports of online fraud cases in buying and selling occurring along with political momentum, such as the general election [3].

The growth of social media and microblogging web services such as Twitter makes it possible to analyze user tweets almost in real time. These Twitter's tweets can be analyzed, considering that users tend to express the level of emotion towards each event using a post or tweet [4]. This analysis is expected to be able to identify which tweets contain hate speech and the motives behind the writing of these tweets. Research that used Twitter's tweets as its data encountered several challenges, which are the classification accuracy,

DOI: 10.24507/icicel.16.08.811

the use of sarcasm within tweets, and word usage variation [5]. The reason for these issues is due to variations in the slang words and other abbreviations used, as well as the limited character of the Tweet (140 characters). Therefore, an approach is needed to solve these problems.

The use of Lexicon-based Features has been used to classify hate speech in Indonesian. By using Lexicon-based Features, various harsh words in ethnic languages in Indonesia can be grouped properly [6]. A study was conducted to classify positive and negative sentiments from sports-related tweets using only Lexicon-based Features, which shows that only by using Lexicon-based Features, the accuracy value can reach 95% [7]. However, a study indicated that the use of Lexicon-based Feature selection can reduce the evaluation value, because many words are not detected after the feature selection is carried out, where the only words that can be detected are those that are taken only in the Lexicon dictionary [8]. Also, Lexicon-based Features cannot perfectly classify hate speech, as it is unable to polarize figures of speech such as sarcasm.

A study researched on Indonesian hate speech sentiment analysis from Twitter using the Naïve Bayes Classifier and Support Vector Machine [9]. The highest accuracy results were obtained when using the Support Vector Machine classification method with unigram tokenization, Indonesian stopword lists and emoticons, with an average Accuracy value of 66.6%, a Precision value of 67.1%, and a Recall value of 66.7%. Another research used a Polynomial Support Vector Machine kernel with two degrees with Query Expansion to analyze reviews from online shop customers [10]. The study used Query Expansion to expand words in the test data that are not found in the training data by looking for synonyms of words in the test data. The final test results produce an average accuracy of 96.25% using Query Expansion and 94.75% without using Query Expansion. These studies indicate that the combination of SVM with Lexicon-based Features and with query expansion can lead into a good performance.

In this study, we propose a combination of Support Vector Machine with Lexicon-based Features and Synonym-based Query Expansion method to classify whether a tweet is a hate speech or not. To our knowledge, the combination of Lexicon-based Features and Synonym-based Query Expansion with Support Vector Machine has never been implemented before. Lexicon-based Features are used to assess the level of polarity of a word, while the Synonym-based Query Expansion method is used so that all words that do not appear in the training data are guaranteed to be processed properly by the model. Raw data preprocessing and variant transformations were carried out to handle slang, abbreviations, and other noises. Afterwards, we observed and analyzed the performance of all models. This research is an extended version of the author's thesis [11].

The following sections are arranged as follows: Section 2 describes the methods used in this research, Section 3 explains and discusses the result, and Section 4 contains the conclusion of this research.

2. **Materials and Methods.**

In this paper, the hate speech identification method based on sentiment analysis technique is applied to classifying Twitter feeds. The process is subdivided into 6 stages: (1) Data acquisition and labelling, (2) Data pre-processing, (3) Feature selection, (4) Weighting, (5) Classification, and (6) Evaluation. The proposed method is shown in Figure 1.

2.1. **Data acquisition and labelling.**

A collection program has been widely used to search for tweets by using certain keywords from the Twitter feed using the Twitter Search API [12]. The data taken is data that has keywords regarding the President and Vice President of the Republic of Indonesia for the period 2019-2024. The total amount of data obtained is 1111 data, which is then divided into 1011 training and validation data and 100 testing data.

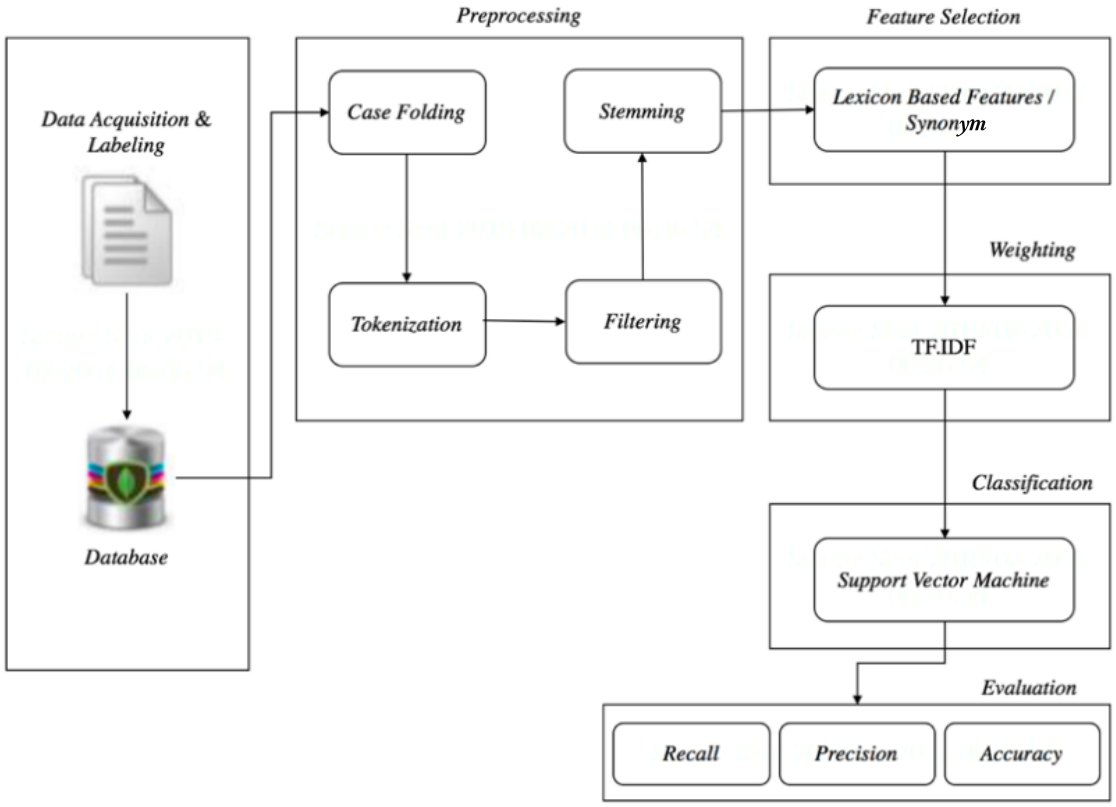

FIGURE 1. Research method

The tweet data that has been obtained is then labelled manually by language linguists who have graduated from postgraduate education. The data is labelled with 2 labels, namely the hate class and the non-hate class. From 1111 data, 791 were labelled as hate and 320 were labelled as non-hate. For the training and validation data, the number of hate classes is 733 and for the non-hate class is 278. For the testing data, 58 data were labelled as hate, and 42 data were labelled as non-hate.

2.2. **Data preprocessing.** Data preprocessing is a process that is carried out to clean and prepare the data before further usage. Preprocessing itself is often conducted based on certain steps, which are case folding, tokenization, filtering, and stemming [13].

2.2.1. *Case folding.* Case folding is the initial stage of text preprocessing which converts all characters of text letters to lowercase. In the text data taken, there are still capital letters, non-standard writing, or typing errors that cause inequality of sentence structures. With the case folding technique, "There", "NO", "WoW" will be changed to "there", "no", "wow". Therefore, case folding is needed so that the documents to be processed have the same structure to facilitate further processing.

2.2.2. *Tokenization.* Tokenization is the stage of cutting the input string based on each word that makes it up [14]. At this stage, the tweet data in the form of sentences or paragraphs are broken down into separate word collections or single words in the form of a word list. In addition, at this stage, whitespace, punctuation marks, and symbols are removed from the text data. This stage aims to tidy up each word in the text to make it easier to extract information.

2.2.3. *Filtering.* Filtering or often called stopword removal aims to select important words from the tokenization results and remove words that are not used in the identification process, such as the words "which", "in", "to", and "can". The process works by matching

pieces of the input string with the stopword dictionary. A stopword matching algorithm is performed by deleting a word in the input string if that word is in the stopword dictionary. Meanwhile, the words that are not in the stopword dictionary will be stored to continue the next process. This process is repeated gradually in each data until the output document is a set of important words.

2.2.4. *Stemming.* Stemming is the process of getting the stem or root word of a word in a sentence by separating each word from its prefix and suffix. For example, the words "together", "togetherness", "equal", will be stemmed to the root word that is "the same". The stemming algorithm for one language is different from the stemming algorithm for another language. For example, English has a different morphology than Indonesian, so the stemming algorithm for the two languages is also different. In English texts, the only process required is the process of removing suffixes. While the Indonesian language text is more complicated/complex because there are variations in the affix that must be removed to get the root word of a word.

2.3. **Feature selection.** After cleaning the tweet data, a feature selection process is carried out to determine the weight of the features that will be used in the training process. The feature extraction process in this study uses the Lexicon dictionary and synonym-based word extensions.

2.3.1. *Lexicon-based Features.* Lexicon-based Features is a method used for the sentiment analysis process, where the process uses a lexical or language source as a dictionary. The working principle of this method is to match words that are in sentiment dictionaries (data containing sentimental words) and calculate the frequency of their occurrence in text documents. The sentiment in this research is hate speech and non-hate speech. Because it only compares two sentiments, in this study, the sentiments that exist in sentiment dictionaries are only the hate speech sentiments. Examples of words with hate speech sentiment are tabulated in Table 1.

Table 1. Indonesian hate speech word sample

Hate speech word			
Bajingan	Asu	Bangsat	Goblok
Bodoh	Bunuh	Tipu	Bohong

2.3.2. *Synonym-based Feature Expansion.* This method uses the concept of Query Expansion. Query Expansion is a technique of rearranging queries by adding words to the information retrieval system query that depends on the user's initial query, in order to increase the number of relevant documents returned [15]. Synonym-based Query Expansion method is used to expand words that are not contained in the features of the training data so that features in the test data that do not appear in the training data can be replaced by using their synonyms. Synonym data used in this method is created online using the Kateglo API, an open application that provides a dictionary, thesaurus, and glossary for Bahasa Indonesia. The synonym list of words will be saved in text format and combined with the synonym list of other words.

2.4. **Term Frequency-Inverse Document Frequency weighting.** At the Term Frequency-Inverse Document Frequency (TF-IDF) weighting stage, each document is weighed to obtain the value of the data term/word that has gone through the previous preprocessing process. This weighting step is carried out by converting the document into a vector with many terms obtained from the results of the preprocessing stage. TF-IDF will calculate the weight of each term so that the term value can represent the document. Table 2

TABLE 2. Example of tweet data that will be calculated using TF-IDF

No	Tweet	Label
1	dukung jokowi presiden	Non-hate
2	presiden jokowi kunjungi singapura	Non-hate
3	presiden jokowi pki	Hate

TABLE 3. Example of TF-IDF

Term	df	TF-d1	TF-d2	TF-d3	Idf	TF-IDF1	TF-IDF2	TF-IDF3	TF-IDF
presiden	3	0.3	0.2	0.3	0	0	0	0	0
jokowi	3	0.3	0.2	0.3	0	0	0	0	0.1
dukung	1	0.3	0	0	0.4	0.1	0	0	0.2
kunjungi	1	0	0.2	0	0.4	0	0.1	0	0.2

shows examples of tweets that have not passed the TF-IDF process, while Table 3 shows the TF-IDF of tweets from Table 2.

2.5. **Classification.** Support Vector Machine (SVM) is one of the supervised classification methods to analyze and recognize patterns from data [16]. SVM is included in supervised learning, which means it requires several data that is used as training data for classification. Based on statistical theory, SVM guarantees an accurate prediction performance [17]. The concept of SVM is to look for the best hyperplanes that separate positive and negative training data. The hyperplane is called a decision boundary or decision surface. Figure 2 shows the results of SVM in linear data sets. w is a normal vector of hyperplane which has the perpendicular direction of the hyperplane [18].

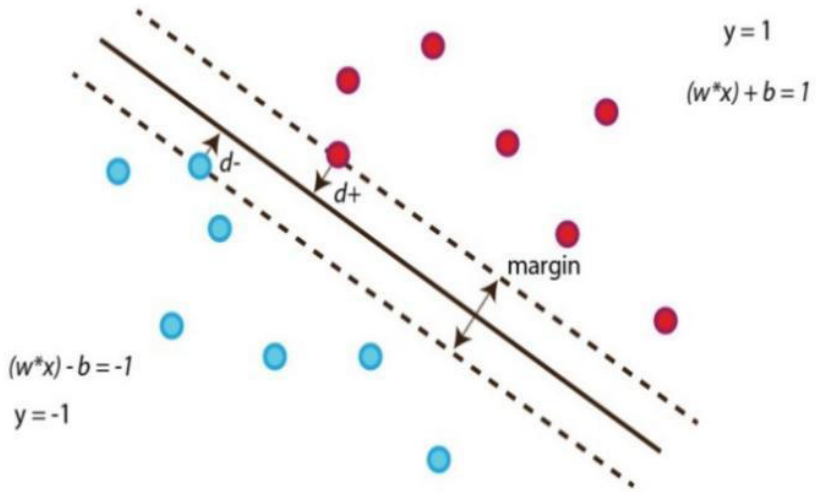

FIGURE 2. SVM in a linear dataset

SVM will try to maximize the margins of positive data and negative data. Supposing that $(w \cdot x) - b$ is a linear equation and each d_+ and d_- is the shortest distance from the separating hyperplane to the positive and negative data points, then the margin of the hyperplane is $d_+ + d_-$. An optimization problem with existing constraints must be maximized to find the maximum margin of the hyperplane (Equation (1)).

$$L_D = \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j a_i a_j (x_i \cdot x_j)$$

$$\sum_{i=1}^{n} y_i a_i = 0$$
$$0 \leq a_i \leq C,\ i = 1, 2, 3, \ldots, n \tag{1}$$

However, among all the a_i values there are only a few that reside inside the hyperplane. These data points are called support vectors. Linear separated data is, in fact, not always ideal because there is noise in the data. SVM that has been described previously cannot find a solution if there are noises in the data. Therefore, to allow errors in the data, we can loosen the constraints to the margins by adding the slack variable $\xi_i \geq 0$, as shown in Equation (2).

$$(w \cdot x) + b \geq 1 - \xi,\ \text{if } y_i = 1$$
$$(w \cdot x) + b \leq 1 - \xi,\ \text{if } y_i = -1 \tag{2}$$

With the use of the slack variable, a new optimization problem must be solved by maximizing Equation (1) within the existing constraints. After completing the optimization problem, we will get support vectors along with the a_i values of each of these data points. The values that have been obtained will be able to produce a decision rule equation that can be used to classify new data. For example, at new z data points, the decision rule is shown in Equation (3). ϕ is a function that maps its input to a high-dimensional feature-space [19].

$$class(z) = sign \sum_{i=1}^{n} y_i a_i (\phi(x) \cdot \phi(z) + b) \tag{3}$$

2.6. **Evaluation.** Before the data is fed into the classification engine, the data is divided into two, namely training data and testing data using the cross-validation method. This test is carried of a 10-Fold Cross Validation. From every time a test occurred to a fold, a matrix with a size of 2×2 will be produced as the representative of the actual class and prediction class. Table 4 explains the confusion matrix for prediction results produced by the SVM.

Table 4. Confusion matrix

	Predicted data	
Actual data	Hate speech	Non-hate speech
Hate speech	THS	FHS
Non-hate speech	FNHS	TNHS

THS is the amount of the hate speech correctly predicted hate speech, FHS is the amount of hate speech incorrectly predicted as a non-hate speech, FNHS is the amount of non-hate speech falsely predicted as hate speech, and TNHS is the amount of non-hate speech correctly predicted as non-hate speech. From the confusion matrix, Accuracy, Precision, Recall, and F-Measure can be calculated by using respectively Equation (4), Equation (5), Equation (6), and Equation (7).

$$Accuracy = \frac{THS + TNHS}{THS + FHS + FNHS + TNHS} \times 100\% \tag{4}$$
$$Precision = \frac{THS}{THS + FHS} \times 100\% \tag{5}$$
$$Recall_{hatespeech} = \frac{THS}{THS + FNHS} \times 100\% \tag{6}$$
$$F\text{-}Measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{7}$$

3. **Results and Discussion.** The training process is carried out on the dataset to form an SVM model that can classify hate and non-hate speech well. To evaluate the model, K-Fold cross-validation is used with K = 10. We trained and evaluated the model on SVM, SVM with Lexicon-based Features, SVM with Synonym, and the SVM with Synonym and Lexicon-based Features. All models that have been trained are then tested for their performance with 100 predetermined test data. The comparison of each model's performance (Accuracy, Precision, Recall, F-Measure) is tabulated in Table 5.

Table 5. Comparison of the classification results

Method	Time	Accuracy	Precision	Recall	F-Measure
SVM	42,298 s	68.84%	89.23%	42.86%	57.90%
SVM + Lexicon	39,903 s	59.11%	73.42%	28.57%	41.14%
SVM + Synonym	**54,307 s**	**70.03%**	**89.74%**	**45.24%**	**60.15%**
SVM + Lexicon + Synonym	49,269 s	59.44%	71.95%	30.95%	43.28%

From Table 5, it can be seen that the best performance is achieved when using the Synonym method, but the processing time is the longest compared to others. This is due to the need to check the Synonym dictionary. On the other side, the Lexicon-based method does accelerate processing time in exchange for lowering its performance, even lower when compared to simple SVM. This may be due to the loss of important features because these features are not contained in the Lexicon corpus. The combination of using the Lexicon-based method and Synonym managed to increase every metric except the Precision compared to the Lexicon-based method without Synonym. Overall, the Synonym method outperforms the combination of Lexicon-based method and Synonym. From these results, we believe that the use of the Synonym method plays a major role in increasing each metric of the simple SVM model.

4. **Conclusions.** Based on the research that has been conducted, the SVM works the best to classify whether a tweet is a hate speech or not when combined with word expansion, in this case, the Synonym method. The SVM with Synonym method achieved the best performance when compared to other combination, scoring Accuracy of 70.03%, Precision of 89.74%, Recall of 45.24% and F-Measure of 60.15%. On the other side, using Lexicon-based features decreased the performance of SVM. This likely due to the loss of important features, as Lexicon-based Features are unable to process word that is not contained in the Lexicon corpus. The Lexicon-based Features also enjoyed an increase in Accuracy, Recall, and F-Measure when combined with Synonym method. From these points, we concluded that the Synonym method plays a vital role in producing a good SVM model, in this case, specifically for identifying whether a tweet is considered hate speech or not. We also recommend combining more robust models such as Neural Network with Lexicon-based as its features with Synonym methods in detecting hate speech to improve the model's performance further.

REFERENCES

[1] W. Warner and J. Hirschberg, Detecting hate speech on the world wide web, *Proc. of the 2nd Workshop on Language in Social Media*, Montréal, Canada, pp.19-26, 2012.

[2] C. C. Lolowang, *Surat Edaran Kapolri No. 06/X/2015 Tentang Penanganan Ujaran Kebencian/Hate Speech (In English: Chief of Police Circular No. 06/X/2015 Concerning Handling Hate Speech)*, Master Thesis, Fakultas Hukum Unissula, Indonesia, 2015.

[3] S. Malmasi and M. Zampieri, Detecting hate speech in social media, *Proc. of Recent Advances in Natural Language Processing (RANLP)*, Varna, Bulgaria, pp.467-472, 2017.

[4] P. Burnap and M. L. Williams, Cyber hate speech on Twitter: An application of machine classification and statistical modeling for policy and decision making, *Policy & Internet*, vol.7, no.2, pp.223-242, 2015.

[5] C. Haruechaiyasak, A. Kongthon, P. Palingoon and K. Trakultaweekoon, S-Sense: A sentiment analysis framework for social media sensing, *Proc. of the IJCNLP 2013 Workshop on Natural Language Processing for Social Media (SocialNLP)*, Nagoya, Japan, pp.6-13, 2013.
[6] M. Hayaty, S. Adi and A. D. Hartanto, Lexicon-based Indonesian local language abusive words dictionary to detect hate speech in social media, *Journal of Information Systems Engineering and Business Intelligence*, vol.6, no.1, pp.9-17, 2020.
[7] F. Wunderlich and D. Memmert, Innovative approaches in sports science – Lexicon-based sentiment analysis as a tool to analyze sports-related Twitter communication, *Applied Sciences*, vol.10, no.2, 431, 2020.
[8] F. R. S. Rangkuti, M. A. Fauzi, Y. A. Sari and E. D. L. Sari, Sentiment analysis on movie reviews using ensemble features and Pearson correlation based feature selection, *2018 International Conference on Sustainable Information Engineering and Technology (SIET)*, Malang, Indonesia, pp.88-91, 2018.
[9] G. A. Buntoro, Analisis sentimen hatespeech pada Twitter dengan metode naive bayes classifier dan support vector machine (In English: Hatespeech sentiment analysis on Twitter using Naive Bayes and support vector machine), *Jurnal Dinamika Informatika*, vol.5, no.2, pp.1-21, 2016.
[10] D. J. Haryanto, L. Muflikhah and M. A. Fauzi, Analisis sentimen review barang berbahasa Indonesia dengan metode support vector machine dan query expansion (In English: Sentiment analysis of goods reviews in Bahasa Indonesia using the support vector machine method and query expansion), *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol.2, no.9, pp.2909-2916, 2018.
[11] I. G. M. J. A. Wikandiputra, *Identifikasi Ujaran Kebencian Pada Twitter Menggunakan Support Vector Machine Berbasis Lexicon Based Features dan Sinonim (In English: Identifying Hate Speech on Twitter Using a Support Vector Machine Based on Lexicon Based Features and Synonyms)*, Master Thesis, Universitas Gadjah Mada, Yogyakata, Indonesia, 2020.
[12] Twitter, *Twitter Developer Documentation*, 2014.
[13] F. Greco and A. Polli, Emotional text mining: Customer profiling in brand management, *International Journal of Information Management*, vol.51, 101934, 2020.
[14] P. Baldi, P. Frasconi and P. Smyth, *Modeling the Internet and the Web*, Chichester, Wiley, 2003.
[15] J. Singh and R. Kumar, Lexical co-occurrence and contextual window-based approach with semantic similarity for query expansion, *International Journal of Intelligent Information Technologies*, vol.13, no.3, pp.57-78, 2017.
[16] V. M. Sutanto, Z. I. Sukma and Afiahayati, Predicting secondary structure of protein using hybrid of convolutional neural network and support vector machine, *International Journal of Intelligent Engineering and Systems*, vol.14, no.1, pp.232-243, 2021.
[17] D. Fradkin and I. Muchnik, Support vector machines for classification, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol.70, pp.13-20, 2016.
[18] Afiahayati, K. Sato and Y. Sakakibara, MetaVelvet-SL: An extension of the Velvet assembler to a de novo metagenomic assembler utilizing supervised learning, *DNA Research*, vol.22, no.1, pp.69-77, 2015.
[19] L. Y. Zhong and T. H. Wang, Towards word sense disambiguation using multiple kernel support vector machine, *International Journal of Innovative Computing, Information and Control*, vol.16, no.2, pp.555-570, 2020.