Bulletin of Electrical Engineering and Informatics
Vol. 13, No. 2, April 2024, pp. 1371~1382
ISSN: 2302-9285, DOI: 10.11591/eei.v13i2.6938

Empowering hate speech detection: leveraging linguistic richness and deep learning

I Gde Bagus Janardana Abasan, Erwin Budi Setiawan
Department of Informatics Engineering, School of Computing, Telkom University, Bandung, Indonesia

Article Info

Article history:

Received Jun 9, 2023
Revised Aug 9, 2023
Accepted Sep 4, 2023

Keywords:

FastText
Feature expansion
Hate speech detection
Hybrid deep learning
Natural language processing

ABSTRACT

Social media has become a vital part of most modern human personal life. Twitter is one of the social media that was formed from the development of communication technology. A lot of social media gives users the freedom to express themselves. This facility is misused by users, so hate speech is spread. Designing a system to detect hate speech intelligently is needed. This study uses the hybrid deep learning (HDL) and solo deep learning (SDL) approach with the convolutional neural networks (CNN) and bidirectional gated recurrent unit (Bi-GRU) algorithm. There are 4 models built, namely CNN, Bi-GRU, CNN+Bi-GRU, and Bi-GRU+CNN. Term frequency-inverse document frequency (TF-IDF) is used for feature extraction, which is to get linguistic features to be analyzed and studied. FastText is used to perform feature expansion to minimize mismatched vocabulary. Four scenarios are run. CNN with an accuracy of 87.63%, Bi-GRU produces an accuracy of 87.46%, CNN+Bi-GRU provides an accuracy of 87.47% and Bi-GRU+CNN provides an accuracy of 87.34%. The ability of this approach to understand the context is qualified. HDL outperforms SDL in terms of n-gram type, where HDL can understand sentences broken down by hybrid n-gram types, namely Unigram-Bigram-Trigram which is a complex n-gram hybrid.

Corresponding Author:

Erwin Budi Setiawan
Department of Informatics Engineering, School of Computing, Telkom University
Landmark Tower, Jl. Terusan Buah Batu, Bandung 40257, Indonesia
Email: erwinbudisetiawan@telkomuniversity.ac.id

1. INTRODUCTION

Social media technology has revolutionized the landscape of both personal and professional communication, and social media platforms are now an almost vital part of most modern human personal lives [1]. Twitter is one of the social media platforms that is widely used by modern people. Twitter offers a medium for all individuals to express themselves freely and opens a place to hear various kinds of expressions and voices that are spread by many people. Ease of access must be followed by online responsibility and the ability to understand existing regulations to create a clean social media environment. These problems are difficult to handle because it is difficult to manage the activities carried out by the user. It is the responsibility of each individual. We must create a safer place for social media environments and avoid the spread of hate speech. A challenging problem that arises in this domain is crucial and requires considerable efforts to improve online responsibility and balance freedom of expression. A highly accurate hate speech detection system should be implemented as soon as possible.

To overcome this problem, some approaches have been made to detect hate speech [2]–[8]. A recent idea in the development of a hate speech detection system is to utilize hybrid deep learning and feature expansion to reduce word mismatches in datasets. However, in previous research, they still used conventional machine learning and solo deep learning in hate speech detection [9], [10]. As far as we know, there is still less research on hate speech detection that utilizes hybrid deep learning and feature expansion. Hybrid deep learning itself is a combination of two or more different deep learning methods and is very useful for training large amounts of data. Deep learning aims to mimic the human brain's ability to create and maintain representations of its environment that predict possible outcomes based on user data, allowing machines to display behavior learned from experience rather than human interaction [11]. Semantic vectors contain many linguistic features that may have features in common with one another. Feature expansion is one of the new methods to reduce vocabulary mismatches that happen in the semantic vector by identifying missing words and replacing them with semantically similar words [12].

Research on hate speech carried out by Melton *et al.* [13] proposed a combination of deep learning approaches with three different datasets. One of the exciting parts of their study is implementing an ensemble that combines convolutional neural networks (CNN), recurrent neural network (RNN), combination FC. What they don't realize is the use of pre-trained models, namely CommonCrawl and Wiki, in extraction using FastText or GloVe. The pre-trained model used is not specific for hate speech detection and, of course, consists of many languages, and the study was overly optimistic. Attention mechanisms and deep learning were used in research [14]. The author in this study uses hybrid ensemble deep learning with CNN and bidirectional gated recurrent unit (Bi-GRU). This is unique because they built a binary classification voting system. The author stated that the voting system and the addition of an attention mechanism to the hybrid layer had a major effect on increasing the accuracy of the model. The attention mechanism certainly has drawbacks in terms of computer complexity and calculations, but these deficiencies are covered by its many advantages, such as making it easier for the model to recognize slugs and slang terms in hate speech. Hybrid deep learning approaches were used in research carried out by Elzayady *et al.* [15]. Their research developed an automated method based on personality literature to identify Arabic hate speech, and they state that their research is the first in this regard.

Several studies have also been conducted on feature expansion [16]–[19]. In one of the studies on hate speech detection [18], GloVe was utilized for feature expansion. The classification still uses machine learning, namely, logistic regression (LR), random forest (RF), and artificial neural network (ANN). The result shows that feature expansion with a combination of term frequency-inverse document frequency (TF-IDF) and corpus tweets built on GloVe provides an average accuracy value of 88.59%. Feature expansions were used by Ghozali *et al.* [20] for the detection of hate speech in Indonesian languages. Their concept of feature expansion is to find synonymous words and add all the synonymous words that they find to the features. The lack of concepts carried out by this author has an impact on computer calculations and the complexity of the algorithms. The addition of another feature causes the current features to become more numerous and uncontrollable, which places a burden on the model. It is important to exercise proper feature selection and consider the trade-off between model complexity and performance. Feature expansion is a challenge and one way to select features. The selection of the correct algorithms and techniques is very necessary for improving the concept of feature expansion.

This study proposes an approach to detect hate speech using a deep learning approach with feature expansion to leverage linguistic richness. Bi-GRU and CNN are two deep learning methods used in this study. In general, we use deep learning and hybrid approaches as in previous studies, but we added a feature expansion that has a different concept from [20] and a comparison between solo deep learning (SDL) and hybrid deep learning (HDL). Our concept is that feature expansion is carried out in a semantic vector to replace missing words with semantically similar words using the help of a self-made corpus using FastText. In summary, the contribution of this paper is as follows: i) comparison between solo deep learning and hybrid deep learning in terms of understanding hate speech more comprehensively; ii) presentation of our feature expansion algorithm concept, which is performed in semantic vector to detect hate speech to minimize computer calculations and the complexity of algorithms; and iii) successfully implementing a good model without overfitting. This approach would represent a breakthrough in hate speech detection. To achieve our research target, this study takes several steps. Building a baseline, or basic model, is the first step in achieving our target. The baseline model is then used as a benchmark model for the next step, in which there are steps for feature expansion and the application of various types of n-grams with TF-IDF and dropout.

The subsequent section of this study is section 2, which will delve into the methodology employed in this research. Section 3 will encompass the findings and discussion of this study. Lastly, section 4 comprises the conclusion, recommendations, and prospects for future research endeavors aimed at enhancing the accuracy of hate speech detection.

2. METHOD

The construction of this hate speech detection system started with crawling or retrieving data from Twitter's social media. After crawling, the data obtained is labeled manually and then enters the pre-processing process. Feature extraction is carried out after all previous processes have been carried out using the n-gram and TF-IDF methods. The results of feature extraction can be used for feature expansion or directly entered into the data split process. Before carrying out the feature expansion, the extraction results should be boolean vectors, which will then be transformed into TF-IDF. The classification uses several deep learning and hybrid deep learning methods, which are described in Figure 1.

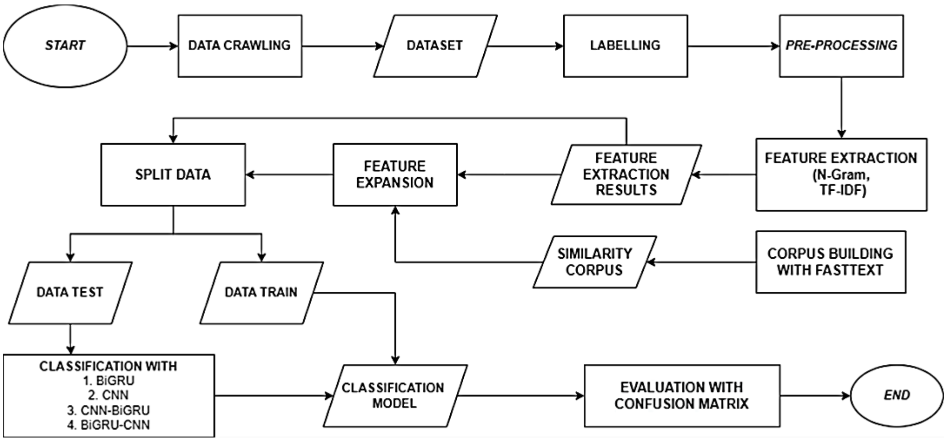


Figure 1. Proposed hate speech detection system

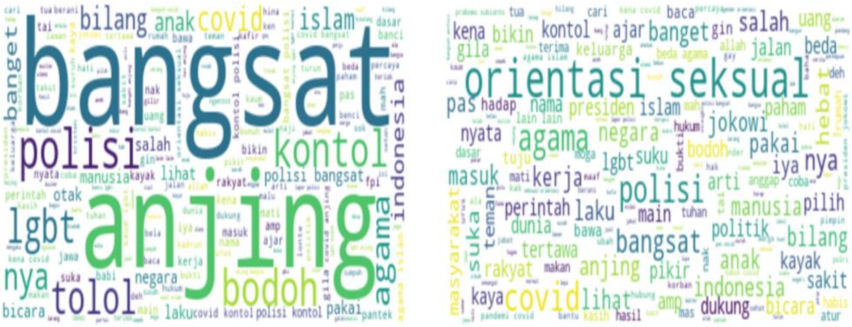
2.1. Data acquisition

Data acquisition, or crawling, is done on Twitter through the application programming interface (API) provided by Twitter. Retrieval of Indonesian-language tweets in a free-language style. Distribution of the topics described in Table 1.

Table 1. Distribution of crawled data

No	Topic	Quantity
1	Police	12,579
2	Religion	15,337
3	Politic	10,055
4	Sexual orientation	10,150
5	Covid-19	10,034
6	Race	2,500
7	Explicit words	3,329

This study uses a dataset consisting of 63,984 Indonesian-language tweets. Our research utilizes binary classification, with hate speech (HS) and non-hate speech (NHS) classes. A visualization of the spread of hate speech can be seen in Figure 2(a), and non-hate speech can be seen in Figure 2(b).



(a) (b)

Figure 2. Visualization of the spread (a) hate speech and (b) non-hate speech

2.2. Preprocessing

The study regarding text and how the text is processed mathematically will be preprocessed, which aims to correct dirty data and does not affect the interpretation or substantive conclusions of the model built [21]. We follow general text preprocessing in natural language processing (NLP), but in Indonesian languages. Here are some stages of preprocessing that we used:

a. Data cleaning: a method for cleaning noise in data (e.g., “@!”;’); Noise data includes special characters, URLs, hyperlinks, emoticons, and unnecessary words [22].
b. Stopword removal: to eliminate words that are considered unimportant in the classification process [23]. Before the stopword removal process takes place, a small dictionary is built that contains words that are not too important according to the characteristics of the dataset (e.g., *agak*, *akan*, *agar*).
c. Case folding: is a technique to change capital letters in text to lowercase letters.
d. Normalization: to fix the words that have an influence on the classification. These fixes include fixing typos, slang words, and abbreviated words.
e. Stemming: one of the steps to change the affixed words to the basic words. Stemming becomes important to retrieve information effectively and efficiently [23] (e.g., *ke-pantai* changed to *pantai*).
f. Tokenizing: is a process of converting sentences into words, phrases, and other meaningful expressions [23].

2.3. Feature extraction

Feature extraction is a technique for extracting relevant information from data so that machines can process it. In this study, feature extraction is performed on a text, which is subsequently transformed into a vector representation and used during the classification process. N-gram is used in this study to break sentences into words according to the number of n requested. From the previous process, a vector representation of the n-gram yield is obtained, which is then weighted with TF-IDF. The main idea of the TF-IDF algorithm is to identify words or phrases that often appear in a document but rarely appear in other documents, this indicates that the document is suitable for classification. The calculation of TF-IDF is formulated in (1):

$$Wij = (TF)_{ij}\ x\ (IDF)_j, where\ (IDF)_j = \log\left(\frac{N}{df}\right) \tag{1}$$

In (1) will apply in certain situations if $TF > 1$, otherwise- $W_{ij} = 0$ [24] which formulated in (2):

$$W_{ij} = \begin{cases} (TF)_{ij}\ x\ \log\left(\frac{N}{df}\right), if\ (TF)_{ij} \geq 1 \\ 0\ , otherwise \end{cases} \tag{2}$$

After performing feature extraction, the results obtained can immediately enter the splitting process or feature expansion. For the splitting process, we break the data into two parts according to the specified proportion (i.e., the data used for testing and the data used for model training).

2.4. Corpus development for feature expansion

This study uses FastText for corpus development. The corpus will be used to build a top-n rank dataset that contains similarities to each other depending on the rank. The top-n-rank corpus will then be used for feature expansion. We solved complexity and computer workload on the feature expansion concept by limiting the n-rank used to find the similarities. Corpus development was done three times with different corpora. The corpus is Twitter, IndoNews, and Twitter-IndoNews. Table 2 contains examples of the top 10 vocabulary words of *legalisasi* (legalization) constructed from corpus similarities in Twitter-IndoNews data.

Table 2. Top 10 word similar to *legalisasi* (legalization)

Word	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
Legalisasi (legalization)	*Legalitas* (legalities)	*Legalisir* (legalizer)	*Legalistic* (legalistic)	*Legalin* (legalin)	*Legal* (legal)
	Rank 6	Rank 7	Rank 8	Rank 9	Rank 10
	Terlegalisir (legalized)	*Konseptualisasi* (conceptualizes)	*Kanibalisasi* (cannibalism)	*Rasionalisasi* (rasionalist)	*Aktualisasi* (actualization)

We construct a similarity corpus based on the n-gram used. The use of n-grams depends on which n-gram performance is the best during the experiment. This will be discussed in the next section which is about experiment and result. Table 3 is an example for the number of words contained in the unigram type of corpus similarity.

Table 3. Vocabulary quantity in each corpus

Corpus similarity	Quantity
IndoNews-Twitter	76,176
Twitter	14,001
IndoNews	70,473

2.5. Feature expansion

The result of feature extraction in the previous step is a semantic vector containing the values of word occurrences. This vector contains vocabulary with the same meaning but with no values or zero values. Feature expansion resolves the vocabulary mismatch issue by changing the zero value to one if there is a word that has the same meaning as each other with a review from the similarity corpus. The algorithm in the explanation of feature expansion is described in Algorithm 1.

Algorithm 1. Feature expansion

```
Input: semantic_vector
Output: Expanded Vector
Initialization: i, j
for i=0 to size (semantic_vector) do
  v ←[ ]
  for j=0 to size (semantic_vector [i]) do
    if Vector[i][j]=0 then:
      cw ← checkWords(i,j)
      expanded_value ←[weightCheck(cw, i,j)]
      v.append(expanded_value)
    else:
      v.append(Vector[i][j])
    endif
  end for
end for
```

WeightCheck is a function to find the weight of a word, whether it's a Boolean value or a TF-IDF weight according to which method is used. CheckWords is a function to validate whether the searched words are in corpus similarity and dataset. The idea of this feature expansion algorithm comes from research [12] which have been modified. Table 4 describes the example of feature expansion concept in this research.

Table 4. Example of semantic vector for feature expansion

No	Text	*Makan* (eat)	*Cinta* (love)	*Dasar* (basic)	*Tolol* (stupid)	*Muslim* (muslim)	*Design* (design)	*Islam* (islam)
1	[*'makan','cinta','dasar','tolol','muslim','banyak','banget','bahagia'*] (['eat', 'love', 'basic', 'stupid', 'muslim', 'a lot', 'really', 'happy'])	1	1	1	1	1	0	0→1
2	[*'random', 'internet', 'alam','jago','design', 'tolol','islam'*] (['random', 'internet', 'nature','good','design', 'stupid','islam'])	0	0	0	1	0→1	1	1

As Table 4 described, our feature expansion concept is performed in semantic vector. The semantic vector contains each sentence broken down into words according to the n-gram requirements used. Our feature expansion concept also involves the use of word embedding methods to represent the semantic relationships between words in sentences. This method allows us to measure the similarity between words that have similar meanings, such as "Muslim" and "Islam" in our previous example. To get the similarities, we used the corpus built before using FastText. By doing so, we are able to replace the zero values found in texts 1 and 2 with the value of one. In contrast to the study [20], our feature expansion concept can reduce computer workload and deal with uncontrolled feature problems.

2.6. Convolutional neural networks

Figure 3 gives an illustration of CNN layer. CNN has several layers: convolutional layer; pooling layer; fully connected layer. Convolutional layers work to determine the output of connected neurons from the input layer [24]. Pooling layers help in sample reduction, allowing smaller data to be represented and making it easier to deal with overfitting [25]. The last layer is the fully connected layer, each neuron in this layer is connected to each other [25]. CNN has become a well-known method recently used in classification.

In addition, CNN has advantages in performing feature extraction and can control a high number of parameters [26]. CNN is more effective in finding more specific features because the feature set is down-sized in the network by the max-pooling layer so that it can more easily understand features that are not too sparse [27].

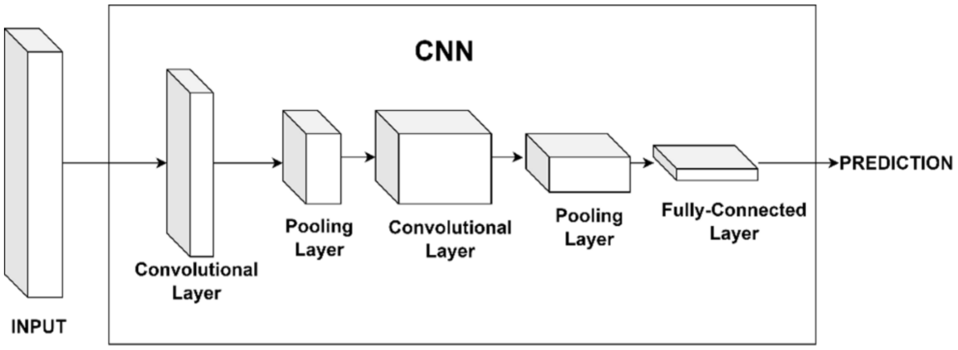


Figure 3. Illustration of CNN

2.7. Bidirectional gated recurrent unit

Bi-GRU consists of two GRUs that run forward and backward methods. In each layer of Bi-GRU, the forward layer computes the hidden layer output from front to back each time, and the backward layer computes the hidden layer output from back to forward each time [28]. GRU is an upgraded version of long short-term memory (LSTM) which has the advantage of computing speed. Within GRU, there are two additional features. reset gates that help capture short-term dependencies in sequences, and update gates that help capture long-term dependencies in sequences. Figure 4 depicts the structure of Bi-GRU.

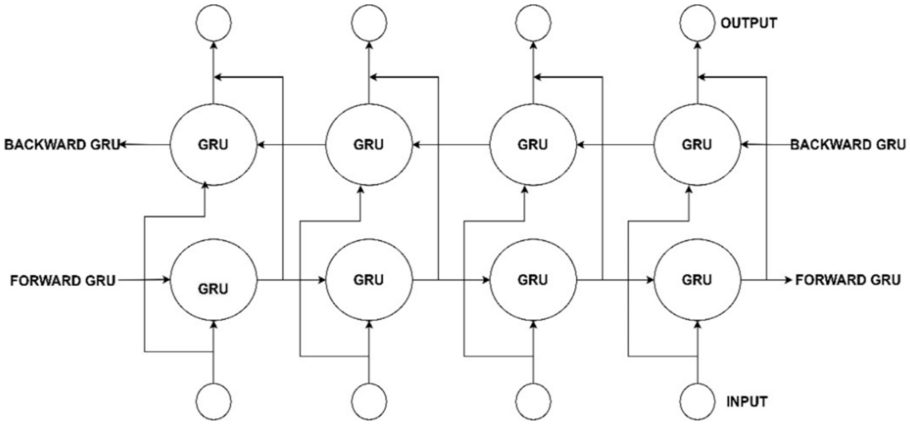


Figure 4. Illustration of Bi-GRU

As described in Figure 4, Bi-GRU consists of two GRUs that receive forward input and backward input. This means that Bi-GRU has an advantage in capturing the context of words and the relationships between words because this method understands a text sequence twice [29]. The utilization of Bi-GRU in content handling is an astute choice. Its inborn capacity to capture settings and connections between words, combined with its demonstrated viability and computational effectiveness, makes Bi-GRU a compelling choice for assignments including successive information.

2.8. Hybrid model

The process of combining two or more deep learning methods is often called hybridization [30]. The paired deep learning hybridizations in this study are CNN+Bi-GRU and Bi-GRU+CNN. Bi-GRU+CNN illustration is shown in Figure 5 and the CNN+Bi-GRU illustration is shown in Figure 6. In the CNN+Bi-GRU combination, this study uses CNN as the initial layer to extract spatial features from the input data, then connect it to Bi-GRU to model the temporal dependence. Whereas in Bi-GRU+CNN this is reversed. This study also compared model performance between HDL and SDL. By comparing the performance of HDL and SDL, we can evaluate whether a hybridization model can affect pattern recognition performance and provide performance improvements. The functions of the layers found in Bi-GRU+CNN and CNN+Bi-GRU are:

a. Embedding layer: layer that functions to change the input text into a mathematical representation. The embedding layer in this study was replaced by TF-IDF.
b. Convolutional 1-D layer: the fundamental layer on CNN which functions to extract local patterns and features from the input data.
c. Bi-GRU layer: the main layer in the development of the Bi-GRU model.
d. Max pooling layer: downsample the input and reduce spatial dimensions.
e. Average max pooling layer: a combination of average pooling and max pooling which is useful for is to preserve both the most salient features captured by the max pooling operation and the overall distribution or average information contained in the input data.
f. Fully connected layer: a fundamental component in neural networks that is useful for networks to understand complex relationships between inputs and outputs.

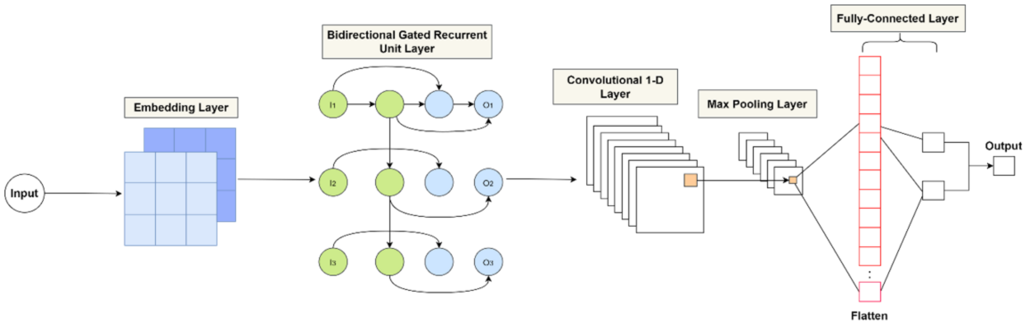


Figure 5. Proposed hybrid deep learning approaches with Bi-GRU+CNN layer

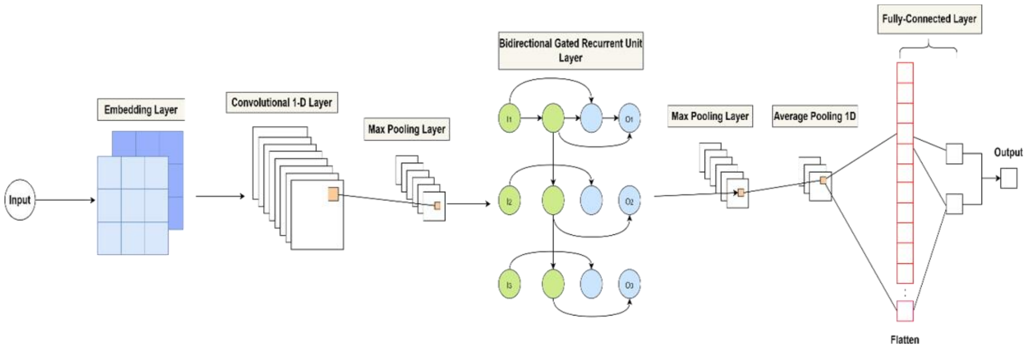


Figure 6. Proposed hybrid deep learning approaches with CNN+Bi-GRU layer

2.9. Evaluation

In this study, the model’s performance was calculated using a confusion matrix, one of which is accuracy. Confusion matrix containing precision, recall, precision and F1 scores. Accuracy is used to calculate what percentage of model inputs were successfully predicted. Recall the calculation of the model’s success rate in finding return information. Precision is for calculated input rate that detected by system. The F1 score is the average of harmonic values and recall accuracy. See formulas for accuracy, recall, precision, and F1-score (3)-(6):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

$$F1 - Score = \frac{2\ x\ (precision\ x\ recall\ )}{(precision+recall)} \quad (6)$$

This matrix evaluation is often referred to as the confusion matrix. This matrix consists of four categories: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) [23].

3. RESULTS AND DISCUSSION

This study conducted experiments in three different hate speech detection scenarios. It's because we want to get the best model performance. Each scenario depends on the previous scenario (e.g., scenario II uses scenario I as benchmark and so on). Table 5 describes all the scenarios in this study.

Table 5. Test case scenario

Scenario	Description	Objectives
I	Apply TF-IDF into the model as feature extraction, data split ratio, and looking for the best value for max features.	Get the baseline model.
II	Test the type of n-gram in TF-IDF parameters and apply a dropout.	Get the best model performance of the n-gram type and the effect of the dropout.
III	Apply feature expansion to model from scenario II using similarity corpus built by FastText.	Get the best model performance after applying the feature expansion.

3.1. Experiment and result

The first scenario is to apply TF-IDF to model as feature extraction, finding best data spit ratio, and best value of max features. The CNN parameters used are filter 32, batch size 64, and epoch 10. Bi-GRU uses parameter unit 32, batch size 128, and epoch 10. The difference is in the hybrid model, we apply a batch size of 128. This study default n-gram type is Unigram. We used a learning rate of 5e^{-5} for each model. We used a low learning rate to prevent overfitting. Those parameters also applied to HDL model.

From Table 6 it is found that for CNN, split ratio is 90:10 and max features 10,000 with an accuracy value of 85.64% is the best than others. Meanwhile, Bi-GRU split ratio is 90:10 and max features 10,000, which gives an accuracy value of 86.54%. Then, on Bi-GRU+CNN described, the highest accuracy is 86.75% obtained from max features of 10,000 and data split ratio of 80:10. CNN+Bi-GRU gives an accuracy 85.95% with 15,000 max features on 90:10 split ratio, highest max features than others. The accuracy produced not far from 85-88%, it's because we prevent overfit by limit the learning rate.

Table 6. Baseline model performance

Max features	Test size	Accuracy (%)			
		CNN	Bi-GRU	CNN+Bi-GRU	Bi-GRU+CNN
5,000	90:10	85.28	86.51	85.75	86.68
	80:20	85.08	86.26	85.39	86.69
	70:30	84.43	85.69	84.28	86.16
10,000	90:10	**85.64**	**86.54**	85.85	86.70
	80:20	83.35	86.39	85.66	**86.75**
	70:30	84.74	85.72	85.00	86.24
15,000	90:10	85.62	86.52	**85.95**	86.39
	80:20	85.31	86.44	84.73	86.49
	70:30	84.70	85.71	84.05	86.06

The results from the baseline model will be used in the next scenario, namely scenario II. To facilitate a comprehensive understanding, we provide a table to facilitate comprehensive understanding and short names for each selected model, which can be seen in Table 7. Column code is the shortened name for each model to give intuitive understanding.

Table 7. Selected baseline model for scenario II

Model	Test size	Max features	Code
CNN	90:10	10,000	Ba_1
Bi-GRU	90:10	10,000	Ba_2
CNN+Bi-GRU	90:10	15,000	Ba_3
Bi-GRU+CNN	80:20	10,000	Ba_4

Table 8 is an experiment from scenario II stage one which utilizes the n-gram type to carry out the test. U stands for Unigram, B stands for Bigram, T stands for Trigram and D stands for dropout. Testing is carried out from the baseline on the previous scenario that has been obtained. The results for the first stage,

Ba_1(CNN) obtained an accuracy value of 87.2% on the Unigram-Bigram n-gram with an increase of 1.56% from the baseline. Ba_2(Bi-GRU) has the highest accuracy of 87.2% on the n-gram Unigram-Bigram-Trigram type with an increase of 1.38% from the baseline. Ba_3(CNN+Bi-GRU) gets high performance on Unigram-Bigram-Trigram with a value of 86.19%, up 0.24% from the baseline. Then, Ba_4(Bi-GRU+CNN) got a performance of 87.07% on Unigram-Bigram-Trigram with an increase of 0.32% from the baseline.

Table 8. Scenario II stages one performance results with baseline model tested on n-gram types

Model	Baseline scores	Accuracy (%)			
		B	T	U-B	U-B-T
Ba_1	85.64	75.23(-1.04)	65.86(-19.78)	**87.20 (+1.56)**	87.02 (+1.38)
Ba_2	86.54	76.85 (-9.69)	66.15(-20.39)	87.08 (+0.54)	**87.20 (+0.66)**
Ba_3	85.95	74.63(-11.32)	60.27(-25.68)	86.1 (+0.15)	**86.19 (+0.24)**
Ba_4	86.75	76.07(-10.68)	65.96(-20.79)	87.00 (+0.25)	**87.07 (+0.32)**

Table 9 is scenario II stage two that looks for the effect of using regularization, namely dropout. Dropout is used to minimize overfitting by removing some neuron networks using certain probabilities. Ba_1(CNN) with 0.5 dropout and Unigram-Bigram type got the highest accuracy among others 87.36%, 1.72% higher than the baseline model and 0.16% increase from scenario II stage one. Ba_2(Bi-GRU) with 0.3 Dropout and Unigram-Bigram type got 87.22% more accuracy with 0.68% higher than the baseline model and 0.02% higher than scenario II model stage one. Next, Ba_3(CNN+Bi-GRU) is hybrid deep learning which got the highest accuracy on the Unigram-Bigram-Trigram dropout 0.5 of 86.19%, an increase of 0.24% from the baseline, and a fixed value from scenario II stage 1. Then, Ba_4(Bi-GRU+CNN) got the highest accuracy value on Unigram-Bigram-Trigram dropout 0.5 of 87.13%, an increase of 0.38% from baseline and 0.06% from scenario II stage 1. So that scenario II stages 2 can be used in scenario III which is the best model used that had been obtained.

Table 9. Scenario II stages two performance results with baseline model tested on n-gram types including dropout regularization

Model	Baseline scores	Accuracy (%)			
		U-B-D0.3	U-B-T-D0.3	U-B-D0.5	U-B-T-D0.5
Ba_1	85.64	87.29 (+1.65)	87.34 (+1.7)	**87.36 (+1.72)**	87.30 (+1.66)
Ba_2	86.54	**87.22 (+0.68)**	87.12 (+0.58)	87.14 (+0.6)	87.09 (+0.55)
Ba_3	85.95	85.98 (+0.03)	85.96 (+0.01)	85.17 (-0.78)	**86.19 (+0.24)**
Ba_4	86.75	87.04 (+0.29)	87.08 (+0.33)	87.01 (+0.26)	**87.13 (+0.38)**

After performing scenario II stage two, the results from that scenario will be used in scenario III. Scenario III tried to find the effect of feature expansion on the model. Table 10 describes the accuracy of the model obtained after carrying out feature expansion for each corpus similarity and ranking in the hate speech dataset. Ba stands for baseline, and S.II for scenario II. Corpus IndoNews used on Ba1+S. II(CNN) with rank 10 gives an accuracy increase in this model, which is 87.63%, 0.27% higher than the previous model. In Ba2+S. II(Bi-GRU), the provision of feature expansion increases the accuracy that was using IndoNews corpus with top 10 rankings. It gives 87.46% accuracy, a 0.24% increase from the previous model. Hybrid deep learning also provides movement in increasing accuracy after providing feature expansion. On Ba3+S. II(CNN+Bi-GRU), the increase occurred with the IndoNews corpus and with top rank 10 of 87.47% accuracy, 1.28% higher than the previous model. Then, Ba4+S. II(Bi-GRU+CNN) provides 87.34% accuracy from the IndoNews corpus with the top 5 ranks, up 0.21% from the previous model.

Table 10. Scenario III performance model results with baseline, scenario II and applying feature expansion

Model	Scenario II-2 scores	Accuracy (%)								
		Twitter			IndoNews			Twitter-IndoNews		
		Top-1	*Top-5*	*Top-10*	*Top-1*	*Top-5*	*Top-10*	*Top-1*	*Top-5*	*Top-10*
Ba_1 + S. II	87.36	87.50 (+0.14)	87.45 (+0.09)	87.42 (+0.06)	87.46 (+0.1)	87.41 (+0.05)	87.63 (+0.27)	87.51 (+0.15)	87.53 (+0.17)	87.46 (+0.1)
Ba_2 + S. II	87.22	87.10 (-0.12)	87.22 (+0)	87.18 (-0.04)	87.18 (-0.04)	87.20 (-0.02)	87.46 (+0.24)	87.18 (-0.04)	87.13 (-0.09)	87.15 (-0.07)
Ba_3 + S. II	86.19	86.53 (+0.34)	87.44 (+1.25)	87.43 (+1.24)	86.51 (+0.32)	86.44 (+0.25)	87.47 (+1.28)	86.54 (+0.35)	87.35 (+1.16)	87.37 (+1.18)
Ba_4 + S. II	87.13	87.27 (+0.14)	87.19 (+0.06)	87.27 (+0.14)	87.23 (+0.1)	87.34 (+0.21)	87.29 (+0.16)	87.19 (+0.06)	87.23 (+0.1)	87.22 (+0.09)

3.2. Discussion

Scenarios I and II are the basic foundations that must be strengthened. From the results, the highest average accuracy is not far from 85%–88%. This is due to the limitations on the learning rate that we found and the adjustments to the dataset that we have. The purpose of using a low learning rate is to build a solid foundation and keep the model from overfitting, so that when the model is given hate speech words that are new or not in the train model, then the model will guess them intelligently. Figure 7 is an example of the validation loss function in Bi-GRU methods that we use as indicators in determining whether the model is overfitting or not.

Judging from Figure 7, train loss and test loss continue to decrease. This indicates that the model is not overfitting and is well maintained. We try to keep that in mind in the next scenario and provide a pretty good model. Hybrid deep learning is proven to be able to superiorly understand the context of a sentence and provide high accuracy values for the hybrid n-gram type. As seen in scenario II, the type of n-gram used is Unigram-Bigram-Trigram. Unigram can help in the identification of individual words; Bigram can help identify the relationship between two words; and Trigram can help identify a word broadly. This feature expansion concept in this study has advantages, as we can see in Table 9. It can increase performance because it learns the context of sentences more comprehensively. The richness of linguistic features does not affect the feature expansion process at all.

The increase in the accuracy of each scenario indicates that our model is well maintained, as Figure 8 describes the accuracy improvement for each scenario in percentage. The CNN that we use provides an accuracy of 87.63%, Bi-GRU produces an accuracy of 87.46%, CNN+Bi-GRU provides an accuracy of 87.47% and Bi-GRU + CNN produces an accuracy of 87.34%. In the CNN+Bi-GRU hybrid model, for example, the increase from scenario II stage two to scenario III is very high around 1.26%. This indicates that the feature expansion can bring significant changes to the model. In short, hybrid deep learning affects performance in understanding the context of sentences, while the inclusion of Unigram, Bigram, and Trigram components with feature expansion increases the power of the system to record various levels of linguistic characteristics.

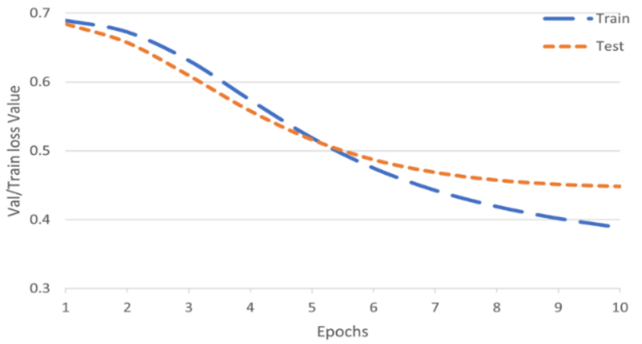
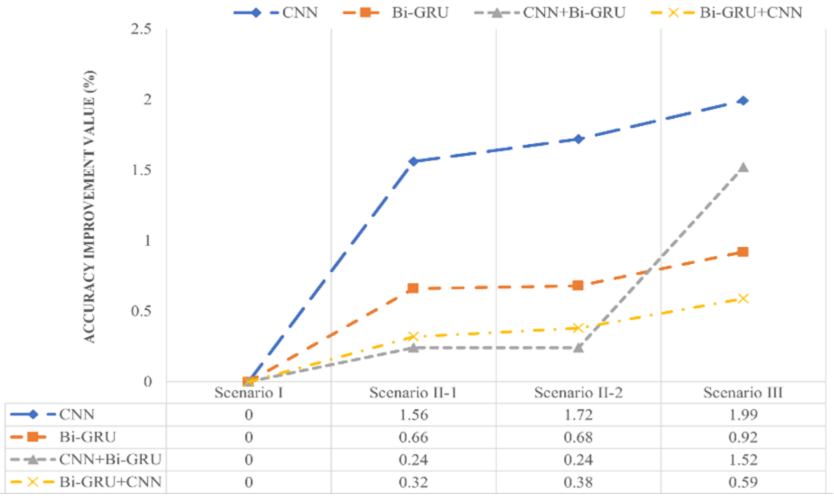


Figure 7. Validation loss function in Bi-GRU



	Scenario I	Scenario II-1	Scenario II-2	Scenario III
CNN	0	1.56	1.72	1.99
Bi-GRU	0	0.66	0.68	0.92
CNN+Bi-GRU	0	0.24	0.24	1.52
Bi-GRU+CNN	0	0.32	0.38	0.59

Figure 8. Accuracy percentage improvement graph each model

4. CONCLUSION

The detection of hate speech in this study uses 63,984 tweets in the Indonesian language which contain loose language styles. In classification using CNN, Bi-GRU, Hybrid CNN+Bi-GRU, and Bi-GRU+CNN. Data collection uses the API provided by Twitter and is labeled manually. TF-IDF is used as a feature extraction that functions to extract information which is then put into mathematical form so that it can be processed by deep learning models. The n-gram combination is used in the extraction of TF-IDF, where Unigram-Bigram and Unigram-Bigram-Trigram provide high accuracy in classification. In addition, the use of a learning rate also affects the model so that it does not overfit. Regularization such as dropout is also used as a useful network to reduce neurons with a certain probability so that the model is not overfitting. FastText is used for building a similarity corpus that is used in the feature expansion. The result we get for SDL, CNN has an accuracy of 87.63% and Bi-GRU gets 87.46% accuracy. Then for HDL, CNN+Bi-GRU model we obtained an accuracy of 87.47% and Bi-GRU+CNN obtained 87.34%. We got this result after using various scenarios. The effect of feature expansion which prove that feature expansion has an impact on the semantic vectors that are useful for training the system. At first glance, SDL is indeed superior in terms of accuracy. However, for understanding the context of sentences, HDL outperforms this. Judging from the type of n-gram used, namely the hybrid n-gram Unigram-Bigram-Trigram which is quite complex to be understood by the system. We hope that future research will focus more on the dataset used, especially in preprocessing step. In addition, we hope that further research can utilize different feature extraction, with any of upgrade version to the model by adding such as attention mechanism or genetic algorithm, maximize the performance of the feature expansion concept and use ternary or multi-class classification.

ACKNOWLEDGEMENTS

The author gratefully acknowledges the financial support provided by Telkom University for this research. Special appreciation goes to Google for generously providing free access of Cloud Computing resources, which were invaluable for training and testing our model.

REFERENCES

[1] R. Chugh and U. Ruhi, “Social media in higher education: A literature review of Facebook,” *Education and Information Technologies*, vol. 23, no. 2, pp. 605–616, Mar. 01, 2018, doi: 10.1007/s10639-017-9621-2.

[2] G. Rizos, K. Hemker, and B. Schuller, “Augment to Prevent: Short-Text Data Augmentation in Deep Learning for Hate-Speech Classification,” *CIKM ‘19: Proceedings of the 28th ACM International Conference on Information and Knowledge Management,* pp. 991–1000, 2019, doi: 10.1145/3357384.3358040.

[3] S. Zimmerman, C. Fox, and U. Kruschwitz, “Improving hate speech detection with deep learning ensembles,” *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, pp. 2546–2553, 2018.

[4] S. S. Aluru, B. Mathew, P. Saha, and A. Mukherjee, “Deep Learning Models for Multilingual Hate Speech Detection,”, *Social and Information Networks,* pp. 1–16, Dec. 2020, doi: 10.48550/arXiv.2004.06465.

[5] Y. Zhou, Y. Yang, H. Liu, X. Liu, and N. Savage, “Deep Learning Based Fusion Approach for Hate Speech Detection,” in *IEEE Access*, vol. 8, pp. 128923-128929, 2020, doi: 10.1109/ACCESS.2020.3009244.

[6] J. A. G-. Díaz, S. M. J-. Zafra, M. A. G-. Cumbreras, and R. V-. García, “Evaluating feature combination strategies for hate-speech detection in Spanish using linguistic features and transformers,” *Complex and Intelligent Systems*, vol.9, pp. 2893-2914, 2023, doi: 10.1007/s40747-022-00693-x.

[7] A. Rana and S. Jha, “Emotion Based Hate Speech Detection using Multimodal Learning,” *Machine Learning,* Feb. 2022, doi: 10.48550/arXiv.2202.06218.

[8] F. Alkomah and X. Ma, “A Literature Review of Textual Hate Speech Detection Methods and Datasets,” *Information (Switzerland)*, vol. 13, no. 6, Jun. 01, 2022, doi: 10.3390/info13060273.

[9] T. L. Sutejo and D. P. Lestari, “Indonesia Hate Speech Detection Using Deep Learning,” *2018 International Conference on Asian Language Processing (IALP),* Bandung, Indonesia, 2018, pp. 39-43, doi: 10.1109/IALP.2018.8629154.

[10] A. Marpaung, R. Rismala, and H. Nurrahmi, “Hate Speech Detection in Indonesian Twitter Texts using Bidirectional Gated Recurrent Unit,” *2021 13th International Conference on Knowledge and Smart Technology (KST)*, Bangsaen, Chonburi, Thailand, 2021, pp. 186-190, doi: 10.1109/KST51265.2021.9415760.

[11] A. Muniasamy and A. Alasiry, “Deep learning: The impact on future eLearning,” *International Journal of Emerging Technologies in Learning*, vol. 15, no. 1, pp. 188–199, 2020, doi: 10.3991/IJET.V15I01.11435.

[12] E. B. Setiawan, D. H. Widyantoro, and K. Surendro, “Feature expansion using word embedding for tweet topic classification,” *2016 10th International Conference on Telecommunication Systems Services and Applications (TSSA),* Denpasar, Indonesia, 2016, pp. 1-5, doi: 10.1109/TSSA.2016.7871085.

[13] J. Melton, A. Bagavathi, and S. Krishnan, “DeL-haTE: A Deep Learning Tunable Ensemble for Hate Speech Detection,” 2020 *19th IEEE International Conference on Machine Learning and Applications (ICMLA),* Miami, FL, USA, 2020, pp. 1015-1022, doi: 10.1109/ICMLA51294.2020.00165.

[14] V. Shah, S. S. Udmale, V. Sambhe, and A. Bhole, “A Deep Hybrid Approach for Hate Speech Analysis,” *CAIP 2021: Computer Analysis of Images and Patterns,* vol. 13052, pp. 424-433, 2021, doi: 10.1007/978-3-030-89128-2_41.

[15] H. Elzayady, M. S. Mohamed, K. M. Badran, and G. I. Salama, “A hybrid approach based on personality traits for hate speech detection in Arabic social media,” *International Journal of Electrical and Computer Engineering*, vol. 13, no. 2, pp. 1979–1988, Apr. 2023, doi: 10.11591/ijece.v13i2.pp1979-1988.

[16] R. A. Yahya and E. B. Setiawan, "Feature Expansion with FastText on Topic Classification Using the Gradient Boosted Decision Tree on Twitter," 2022 *10th International Conference on Information and Communication Technology (ICoICT)*, Bandung, Indonesia, 2022, pp. 322-327, doi: 10.1109/ICoICT55009.2022.9914896.

[17] M. F. D. Putra and E. B. Setiawan, "Influence of Sentiment on Mandiri Bank Stocks (BMRI) Using Feature Expansion with FastText and Logistic Regression Classification," *2022 International Conference on Advanced Creative Networks and Intelligent Systems (ICACNIS)*, Bandung, Indonesia, 2022, pp. 1-7, doi: 10.1109/ICACNIS57039.2022.10055450.

[18] F. Anistya and E. B. Setiawan, "Hate Speech Detection on Twitter in Indonesia with Feature Expansion Using GloVe," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 6, pp. 1044–1051, 2021, doi: 10.29207/resti.v5i6.3521.

[19] C. Yang, W. Zheng, Y. Xiao, and C. Dong, "A short text sentiment classification method based on feature expansion and bidirectional neural network," *2021 International Conference on Big Data Analysis and Computer Science (BDACS)*, Kunming, China, 2021, pp. 195-198, doi: 10.1109/BDACS53596.2021.00050.

[20] I. Ghozali, K. R. Sungkono, R. Sarno, and R. Abdullah, "Synonym based feature expansion for Indonesian hate speech detection," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 1, pp. 1105–1112, Feb. 2023, doi: 10.11591/ijece.v13i1.pp1105-1112.

[21] M. J. Denny and A. Spirling, "Text Preprocessing for Unsupervised Learning: Why It Matters, When It Misleads, and What to Do about It," *Political Analysis*, vol. 26, no. 2, pp. 168–189, Apr. 2018, doi: 10.1017/pan.2017.44.

[22] S. Gharatkar, A. Ingle, T. Naik, and A. Save, "Review preprocessing using data cleaning and stemming technique," *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, India, 2017, pp. 1-4, doi: 10.1109/ICIIECS.2017.8276011.

[23] W. Bourequat and H. Mourad, "Sentiment Analysis Approach for Analyzing iPhone Release using Support Vector Machine," *International Journal of Advances in Data and Information Systems*, vol. 2, no. 1, pp. 36–44, Apr. 2021, doi: 10.25008/ijadis.v2i1.1216.

[24] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network*," 2017 International Conference on Engineering and Technology (ICET)*, Antalya, Turkey, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.

[25] H. Khotimah, E. Budi, and I. Kurniawan, "Implementation Information Gain Feature Selection for Hoax News Detection on Twitter using Convolutional Neural Network (CNN)," *Indonesia Journal on Computing (Indo-JC)*, vol.5, no.3, 2020, doi: 10.34818/INDOJC.2020.5.3.506

[26] N. Kaur and G. Gupta, "Refurbished and improvised model using convolution network for autism disorder detection in facial images," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 29, no. 2, pp. 883–889, Feb. 2023, doi: 10.11591/ijeecs.v29.i2.pp883-889.

[27] B. Gambäck and U. K. Sikdar, "Using Convolutional Neural Networks to Classify Hate-Speech," *Proceedings of the First Workshop on Abusive Language Online*, pp. 85–90, Vancouver, BC, Canada., 2017, doi: 10.18653/v1/w17-3013.

[28] P. Li *et al.*, "Bidirectional gated recurrent unit neural network for Chinese address element segmentation," *ISPRS International Journal of Geo-information*, vol. 9, no. 11, pp. 1–19, Oct. 2020, doi: 10.3390/ijgi9110635.

[29] Y. Lee, S. Yoon, and K. Jung, "Comparative Studies of Detecting Abusive Language on Twitter," *Computation and Language*, Aug. 2018, doi: 10.48550/arXiv.1808.10245.

[30] B. Jena, S. Saxena, G. K. Nayak, L. Saba, N. Sharma, and J. S. Suri, "Artificial intelligence-based hybrid deep learning models for image classification: The first narrative review," *Computers in Biology and Medicine*, vol. 137. Elsevier Ltd, Oct. 01, 2021, doi: 10.1016/j.compbiomed.2021.104803.

BIOGRAPHIES OF AUTHORS

I Gde Bagus Janardana Abasan is pursuing a Bachelor's Degree in Computer Science at Telkom University. He is very interested in the fields of artificial intelligence, machine learning, and software engineering. His interest continued and he became part of the IT research team at Direktorat Pusat Teknologi Informasi, Telkom University. He can be contacted at email: bjanardana@student.telkomuniversity.ac.id.

Erwin Budi Setiawan is a senior-lecturer in Informatics, School of Computing, Telkom University, Bandung, Indonesia. He has more than 10 years of research and teaching experience in the domain of informatics. Currently, he is an Associate Professor. His research interests are machine learning, people analytics, modeling and simulation, and social media analysis. He can be contacted at email: erwinbudisetiawan@telkomuniversity.ac.id.