Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)
Vol. 9, No. 3, September 2023, pp. 619-631
ISSN: 2338-3070, DOI: 10.26555/jiteki.v9i3.26532

Hate Speech Detection Using Convolutional Neural Network and Gated Recurrent Unit with FastText Feature Expansion on Twitter

Kevin Usmayadhy Wijaya, Erwin Budi Setiawan
Telkom University, Jl. Terusan Buah Batu, Bandung 40257, Indonesia

ARTICLE INFO

Article history:

Received June 22, 2023
Revised July 16, 2023
Published July 20, 2023

Keywords:

Hate speech;
FastText;
Feature Expansion;
Hybrid deep learning;
Convolutional neural network;
Gated recurrent unit

ABSTRACT

Twitter is a popular social media for sending text messages, but the tweets that can send are limited to 280 characters. Therefore, sending tweets is done in various ways, such as slang, abbreviations, or even reducing letters in words which can cause vocabulary mismatch so that the system considers words with the same meaning differently. Thus, using feature expansion to build a corpus of similarity can mitigate this problem. Two datasets constructed the similarity corpus: the Twitter dataset of 63,984 and the IndoNews dataset of 119,488. The research contribution is to combine deep learning and feature expansion with good performance. This study uses FastText as a feature expansion that focuses on word structure. Also, this study uses four deep learning methods: Convolutional Neural Network (CNN), Gated Recurrent Unit (GRU), and a combination of the two CNN-GRU, GRU-CNN classification with boolean representation as feature extraction. This study uses five scenarios to find the best result: best data split, n-grams, max feature, feature expansion, and dropout percentage. In the final model, CNN has the best performance with an accuracy of 88.79% and an increase of 0.97% from the baseline model, followed by GRU with an accuracy of 88.17% with an increase of 0.93%, CNN-GRU with an accuracy of 87.47% with an increase of 1.86%, and GRU-CNN with an accuracy of 87.55% with an increase of 1.32%. Based on the result of several scenarios, the use of feature expansion using FastText succeeded in avoiding vocabulary mismatch, proven by the highest increase in accuracy of the model than other scenarios. However, this study has a limitation is that the dataset is used in Indonesian.

Corresponding Author:

Erwin Budi Setiawan, Telkom University, Jl. Terusan Buah Batu, Bandung 40257, Indonesia
Email: erwinbudisetiawan@telkomuniversity.ac.id

1. INTRODUCTION

Hate speech is speech directed at individuals or groups that do not follow the norm, threaten, abuse, insult, embarrass, cause harm, and can cause social chaos [1], [2]. In Indonesia, hate speech is regulated in the ITE Law Number 11 of 2008, with a maximum sentence of 6 years. Hate speech can occur anywhere, especially on social media, where people often express their opinions, criticisms, and many more that can lead to hatred [3]. Based on that, we need a system for hate speech detection on Twitter to create a good, safe, and disciplined environment for regulations. However, tweet messages are limited to 280 characters [4]. With the limited character length, people use various ways to convey the desired message, such as through slang, abbreviations, and reducing letters in words. This can cause vocabulary mismatch and becomes a problem when the system wants to classify text because the system can consider the same meaning as a different word [5]. To overcome the problem of vocabulary mismatches can be reduced with feature expansion. However, there are many studies have been done before regarding hate speech detection. Research development on hate speech detection systems currently uses Deep Learning and Word Embedding [6]–[12].

Research on hybrid deep learning for Hate Speech Detection has been done before [9]–[11]. Alshalan *et al.* [9] studied a hybrid deep learning method, namely CNN+GRU, and compared it with CNN, GRU, and Bidirectional Encoder Representation from Transformers (BERT) methods used. In this study, the CNN model performed best with a score of F1-Score of 0.79 and an Area Under the Receiver Operating Characteristic Curve (AUROC) of 0.89. The researchers Duwairi *et al.* [10] use deep learning and hybrid deep learning models to classify hate speech using the CNN, CNN-LSTM, and BiLSTM-CNN methods using the ArHS dataset and Combined dataset. The result of this study is that on the ArHS dataset for the binary classification task, the CNN model produces an accuracy of 0.81 which is the best result. For the ternary classification task, the best accuracy is obtained on CNN and BiLSTM-CNN with a score of 0.74. For the multi-class classification task, CNN-LSTM and BiLSTM-CNN produced the most remarkable accuracy with a score of 0.73. Then in the Combined dataset for the binary classification task, BiLSTM-CNN produced the highest score of 0.73. BiLSTM-CNN got the best accuracy for the ternary classification task at a score of 0.67. And for the multi-class classification task, CNN-LSTM and BiLSTM-CNN both produce the best performance with a score of 0.65. The studies from Zhang *et al.* [11] SVM, CNN, and CNN+GRU models were used for the classification tested into seven datasets, where CNN+GRU produced the best performance among the other models. In this study, they compared their model with some results from previous studies with the same dataset and produced the best performance in 6 datasets with an increase of 1-13% in F1-Score.

Research on feature expansion has also been done before [3], [12], [13]. Anistya *et al.* [3] studied the feature expansion method using the GloVe model to reduce the problem of vocabulary mismatch in Indonesian tweets. The classification model used in this research is LR, RF, and ANN Algorithm. The best performance is obtained by the RF model with 5000 features with TFIDF and the expansion of features from the tweet corpus, and based on several experiments in this study, a number of features have been proven to increase the system's accuracy. Dewi *et al.* [12] used Word2Vec for the feature expansion method, and the classification model used in this study are SVM and RF. The dataset in this study contained 20,571 tweets in Indonesian, where the study's results proved that feature expansion dan TFIDF for weighting could reduce the problem of vocabulary mismatch and improve the accuracy with the best result on the RF method with an accuracy of 0.88. Alhakiem *et al.* [13] used FastText as a feature expansion with the LR classifier to classify text with 16,987 data tweets. FastText managed to increase accuracy by 2.84% for sentiment classification on the signal aspect with F1-Score 0.96 and increase accuracy by 10.05% for sentiment classification on the service aspect with F1-Score 0.95.

Based on the related work, FastText is chosen to reduce vocabulary mismatches because it is suitable for foreign languages and focuses on word structure [13]. CNN is chosen, although this method is usually used for image recognition [14]–[18]. But recently, many studies have proven that CNN performs remarkably accurately for text classification [19]–[24]. Gated Recurrent Units (GRU) are chosen because it is one of the Recurrent Neural Network (RNN) methods and is a simpler method than Long Short Term Memory (LSTM) because it does not have an output gate and forget gate but still has a good performance [11], [25], [26]. As far as researchers know, Although many studies have been conducted before, no one has explored hate speech detection using hybrid deep learning and feature expansion simultaneously, especially in Indonesian Twitter. The research contributes to increasing the focus area of hate speech detection on Indonesian Twitter and using feature expansion to overcome the vocabulary mismatch problem.

This paper consists of four sections. Section 1 includes background and related work. Section 2 contains a description of the method used in this research. Section 3 consists of results and discussion. And Section 4 is the conclusion of this research.

2. METHODS

The method of this experiment consists of several stages that aim to eliminate vocabulary mismatch with feature expansion and get the best classification from the architecture model built based on several scenarios and can be used for further model research. The system design flow created for hate speech detection can be seen in Fig. 1.

2.1. Crawling Data

Crawling data is the process of collecting data online. The crawling process takes data from tweets and retweets on Twitter using the API (Application Program Interface) provided by the Twitter developers themselves [27]. The data that is crawled is in Indonesian with several topics and keywords. This topic was chosen because these topics have been used for hate speech detection in previous studies and added several

new topics [12], [28], [29]. The crawling process was carried out from November 2022 – April 2023, which produced 63,984 data tweets. Table 1 is a list of topics from the data crawling performed.

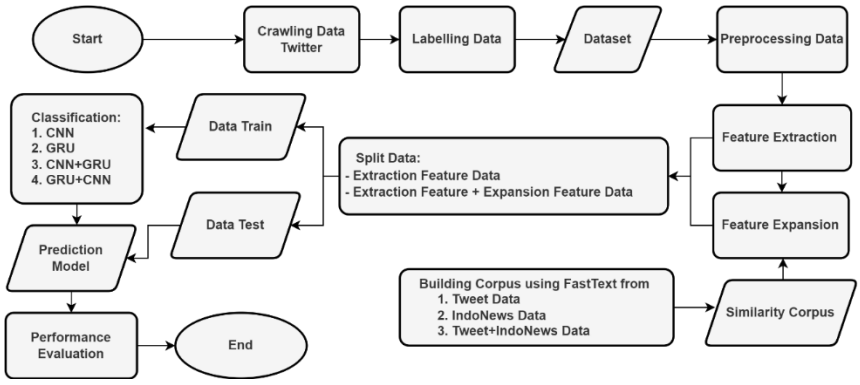

Fig. 1. Flowchart Hate Speech Detection System

Table 1. Crawling Topic List

Topic	Keyword	Total Tweets
Kepolisian	polisi, polri	12,579
Oriental Seksual	orientasi seksual, lgbt, gay, lesbian, biseksual, transgender	10,150
Politik	politik, politisi, partai, dpr, pdip, gerindra	10,059
Covid-19	covid-19, covid	10,034
Agama	agama, fpi, islam, muslim, kristen, katolik, kafir, haram, budha, muhammadiyah, injil	15,333
Ras	ras, suku, jawa, papua, batak	2,500
Explicit Word	goblok, tai, pantek, tolol, babi, lonte, kontol, gila, anjing, bangsat, bajingan	3,329

2.2. Labelling Data

The data obtained from the data crawling process has not yet been identified whether the data includes Hate Speech or not. So labeling data is the process of determining the class of data [30]. Therefore, a manual data labeling process was carried out, divided into two labels, HS (Hate Speech) and NHS (Nonhate Speech). Data labeled as HS is data contains provocation, incitement to hatred, or insults to an individual or group. The data labeling process was carried out by voting by three people. If, in a tweet, only one person labels HS and two other people label it as NHS, then the data will be considered as NHS, and vice versa. Manual labeling with voting has also been carried out in previous studies [31]. The result of labeling is there were 32.014 tweets labeled as HS and 31,970 tweets labeled as NHS.

2.3. Preprocessing Data

Crawling data produces raw data so that many have noise (data that is not important). The data preprocessing is carried out to reduce that noise [32]. There are 5 data preprocessing carried out in this study, including [33]. Table 2 is an example of data preprocessing.

1. Data Cleaning is the process of cleaning data from attributes that are not needed for data entry, such as symbols, punctuation marks, numbers, URLs, and empty data.
2. Case Folding is the process of equating text into the same form so that the same words are not considered different words, and in this study, the case folding process is changing all letters to lowercase.
3. Stop Words is the process of removing common words that have no importance or are irrelevant so that it will reduce the input data. Stop words data is taken from the nltk library.
4. Normalization is the process of normalizing text (e.g., misspelling correction, normalizing slang words, etc.) so words that don't fit are normalized will become appropriate.
5. Stemming is the process of reducing words to basic words (removing affixes). This process uses the sastrawi library.
6. Tokenizing is the process of splitting words into several word chunks (usually splitting by spaces).

After preprocessing the data, data that has null and duplicates are dropped. After the data is removed, the data becomes 54,767 tweets. Fig. 2(a) is the amount distribution of the data before preprocessing, and Fig. 2(b) is the amount distribution of the data after preprocessing.

Table 2. Example of Data Preprocessing

Preprocessing	Text
Original Tweet	@zoelfick Anjing gua tangkap dan penjarakan tuh bocah goblog nyiksa temen kaya gitu apa dia turunan setan x....polisi jangan biarkan tangkap pelaku dan teman2nya
Data Cleaning	Anjing gua tangkap dan penjarakan tuh bocah goblog nyiksa temen kaya gitu apa dia turunan setan polisi jangan biarkan tangkap pelaku dan temannya
Case Folding	Anjing gua tangkap dan penjarakan tuh bocah goblog nyiksa temen kaya gitu apa dia turunan setan polisi jangan biarkan tangkap pelaku dan temannya
Normalization	Anjing saya tangkap penjarakan tuh bocah goblog nyiksa temen kaya gitu turunan setan polisi biarkan tangkap pelaku temannya
Stop Words	anjing tangkap penjarakan tuh bocah goblog nyiksa temen kaya gitu turunan setan polisi biarkan tangkap pelaku temannya
Stemming	anjing tangkap penjara tuh bocah goblog nyiksa temen kaya gitu turun setan polisi biar tangkap laku teman
Tokenizing	['anjing', 'tangkap', 'penjara', 'tuh', 'bocah', 'goblog', 'nyiksa', 'temen', 'kaya', 'gitu', 'turun', 'setan', 'polisi', 'biar', 'tangkap', 'laku', 'teman']

(a) (b)

Fig. 2. Distribution of Labeled Data (a) Before Preprocessing and (b) After Preprocessing

2.4. Feature Extraction

Feature extraction is extracting relevant information from raw data [34]. In this study, the data were first processed by n-grams, where the words would be divided along n words. In this study, it was carried out three times, namely n=1 (Unigram), n=2 (Bigram), and n=3 (Trigram) [35]. The use of n-grams is intended so that the feature extraction that is carried out can see a series of words so that system can find out more deeply about the context of the data [29]. After dividing the word along n words, the result will be represented using a Boolean Feature vector with a fixed length. Boolean Feature converts text data into numbers 1 and 0. If the letters in the vector are in the text being extracted, they will be extracted as 1, and if not will be extracted as 0. Suppose there are five encoded vectors of words sequentially {“i”, “you”, “book”, “hate”, “love”} then the sentence “i love book” will be represented by {1, 0, 1, 0, 1} [5].

2.5. Feature Expansion

FastText is an efficient and fast open-source method developed by Facebook and first introduced in 2014 [36], [37]. FastText is an extension of the Word2Vec and follows a skip-gram model considering sub-word information [38]; e.g., the word “hate”, “hated”, and “hating” will be expressed as follows:

- hate: <h, ha, hat, hate, a, at, ate, t, te, e>
- hated: <h, ha, hat, hate, hated, a, at, ate, ated, t, te, ted, e, ed, d>
- hating: <h, ha, hat, hati, hatin, hating, a, at, ati, atin, ating, t, ti, tin, ting, i, in, ing, n, ng, g>

There are many similar characters from hate, hated, and hating, so the three words above will be considered as similar words and will be saved in the similarity corpus. With this model, the system can be able to know vocabulary mismatch [39], [40]. FastText in this study is used to create a similarity corpus based on IndoNews and Twitter. IndoNews data is taken from several media in Indonesia, namely cnnindonesia.com, detik.com, kompas.com, republika.com, sindonews.com, and tempo.co. Twitter data is taken from previous data that has been preprocessed. Table 3 is the total corpus data used. FastText will use these data. There are three rankings that will be used Top 1, Top 5, dan Top 10. The higher the word rank, the more similar the word is to the original word [31]. Table 4 is an example of the Top 10 words similar to “kapitalisme”.

Table 3. Total Data Corpus

Topic	Total
IndoNews	119,488
Tweets	54,767
IndoNews+Tweets	174,255

Table 4. Top 10 Words Similar to "kapitalisme"

Word	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
kapitalisme	liberalisme	idealisme	pluralisme	feodalisme	komunis
	Rank 6	**Rank 7**	**Rank 8**	**Rank 9**	**Rank 10**
	ironis	vonis	radikal	rompi	realistis

After obtaining the similarity corpus, feature expansion will then be carried out. Feature expansion is a method for solving vocabulary mismatch problems that aim to identify features with a value of 0 to look for similar words. If similar words have a value of 1, then the value of 0 will be replaced with a value of 1 [12]. E.g., there is the sentence "…negara penganut kapitalisme" and assuming the word "kapitalisme" is represented by the number 0 and the word "liberalisme" is represented by the number 1, then the word "kapitalisme" can be changed to 1 because it is a similar word to the word "liberalisme" based on similarity in Table 4 which has been built by FastText.

2.6. Convolutional Neural Network (CNN)

CNN is one of the deep learning methods whose primary function is used for image recognition [41], but CNN itself can have different convolution dimensions, including [42]:

- One-dimensional (Conv1D), suitable for text data, input and output data from this convolution are two-dimensional, and the convolution kernel moves in one direction.
- Two-dimensional (Conv2D), suitable for image data, input and output data from this convolution are three-dimensional, and the convolution kernel moves in two directions.
- Three-dimensional (Conv3D), suitable for 3D image data, input and output data from this convolution are four-dimensional, and the convolution kernel moves in three directions.

In this study, the CNN model is used to classify because CNN has advantages in extracting important features from each data, and the convolution used is one-dimensional (Conv1D) because used as text classification. Fig. 3 is a visualization of the input text and the layers that are traversed to produce predictive data labels. First, the Conv1D layer will receive input data which is then extracted with the help of filters. Second, MaxPooling1D will reduce the size of the output features in the previous layer to increase computational efficiency and eliminate noise. Third, Dropout is used to avoid overfitting by decreasing the percentage of feature size. Fourth, Flatten will change the resulting vector into one dimension. Finally, the Dense layer will train the resulting network to determine class labels [43].

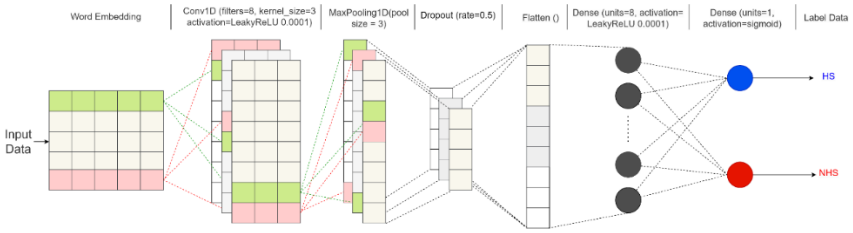

Fig. 3. CNN Architecture Built in This Study

The CNN architecture was built in this study using the TensorFlow library. This study has carried out hyperparameter tuning by optimizing the number of filters, learning rate, kernel size, and pool size. Hyperparameter tuning is used to overcome overfitting and improve model accuracy. After the hyperparameter tuning is done, there is one Conv1D layer (filter=8, kernel size=3, activation=LeakyReLU 0.0001), MaxPooling1D layer (pool size=3), Dropout layer (rate=0.5), Flatten layer, Dense Layer (units =8, activation=LeakyReLU 0.0001), and Dense Layer (units=1, activation=sigmoid).

2.7. Gated Recurrent Unit (GRU)

GRU is one of the RNN algorithms similar to the LSTM algorithm, except that GRU does not have an output gate and a forget gate but only has a reset gate and an update gate so that GRU processing and implementation is simpler [44]. GRU was developed to avoid the vanishing gradient problem. In some cases,

GRU produces better performance than LSTM and can perform calculations faster while reducing memory usage [45].

Fig. 4 is a visualization of the structure of the GRU model. I_t is the input for the current timestep, h_{t-1} is the hidden state for the previous timestep, and h_t is the hidden state for the current timestep. The GRU model has two gates. The first is the reset gate (R_t) which helps reset the main gate function and determines which information should be forgotten (value is 0), remembered (value is 1), or partially remembered (value is between 0 and 1). Then multiplied by the combined h_{t-1} and I_t then multiplied by W_r (weight reset gate) and added with b_r (bias reset gate) (1) [46].

$$R_t = \sigma([h_{t-1}, I_t] \,.\, W_r + b_r) \tag{1}$$

Then there is an update gate (U_t) which helps the model determine which information can be forwarded to the future (update the hidden state to a new state), similar to the reset gate, except that it has a different weight using weight update gate (W_u) and bias update gate (b_u), and the sigmoid function is subtracted by a vector of 1 value (2) [46].

$$U_t = \sigma([h_{t-1}, I_t] \,.\, W_u + b_u) \tag{2}$$

After the reset gate and gate update have been successfully carried out, the candidate hidden state (Ch_t) calculation is performed. The reset gate calculation results are multiplied by the weight and added with the bias in the hidden state, then calculated using the tanh activation function (3) [46].

$$Ch_t = \tanh([R_t, h_{t-1}, I_t].W_h + b_h \tag{3}$$

And finally, the hidden state is obtained from the result 1-U_t multiplied by the previous hidden state and then added with U_t, which has been multiplied by Ch_t (4) [46].

$$h_t = (1 - U_t).h_{t-1} + U_t.Ch_t \tag{4}$$

This study has carried out hyperparameter tuning by optimizing the number of units and learning rate. The GRU architecture was built in this study using the TensorFlow library. After the hyperparameter tuning is done, there is one GRU layer (unit=8, activation=LeakyReLU 0.0001), GlobalMaxPool1D layer, Dropout layer (rate=0.5), Flatten layer, Dense Layer (units=8, activation=LeakyReLU 0.0001), and Dense Layer (units=1, activation=sigmoid).

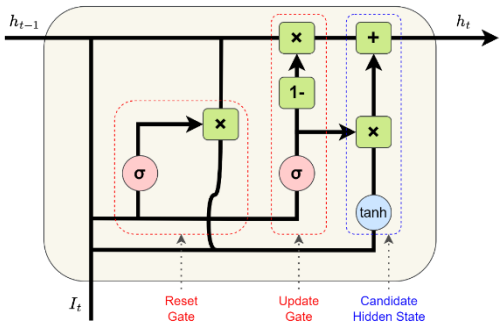

Fig. 4. Structure of the GRU

2.8. Hybrid Deep Learning

Hybrid Deep Learning is a new model obtained from a combination of two or more different deep learning methods [45]. In this study, the deep learning methods combined were CNN and GRU. Two hybrid deep learning methods are carried out, namely, CNN+GRU, where CNN is the first model to receive input, and the output results from the CNN are used to train the GRU model, and GRU+CNN, where data is input to the GRU model first then the output results will be used to be trained by the CNN model. In the architecture of CNN+GRU model, there is one Conv1D layer (filter=8, kernel size=3, activation=LeakyReLU 0.0001), one GRU layer (unit=8, activation=LeakyReLU 0.0001), GlobalMaxPool1D layer, Dropout layer (rate=0.5), Flatten layer, Dense Layer (units=8, activation=LeakyReLU 0.0001), and Dense Layer (units=1, activation=sigmoid). And also applies to GRU+CNN, except that the CNN and GRU layers are reversed.

2.9. Performance Evaluation

The performance evaluation used in this study is the Accuracy, F1-Score, and area under the receiver operating characteristic curve (AUROC). There are several terms commonly used, namely TP (True Positive), FP (False Positive), FN (False Negative), and TN (True Negative) [47]. Accuracy is the amount of data classified correctly (5). Before calculating the F1-Score, precision, and recall must be sought first. Precision is the ratio of positively predicted data to the actual positive data (6), recall is the ratio of positively predicted data to the overall predicted data (7), and F1-Score is the harmonic average of precision and recall (8) [48]. AUROC is used as an indicator of the goodness of the model to the class of labels that exist by using the curve of the TP and FP rate [10].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

$$F1 - Score = \frac{2\ x\ Precision\ x\ Recall}{Precision + recall} \quad (8)$$

3. RESULTS AND DISCUSSION

There are five scenarios in the experiment carried out in this study. This experiment was carried out to find the best performance by comparing the value of the accuracy. The accuracy obtained is the result of the average of the five tests in each scenario. Table 5 is a test scenario that was carried out in this study.

Table 5. Test Scenario

Scenario	Description	Objective
1	Test the model using the Boolean feature as feature extraction with several split ratios.	Determine the baseline with the best split ratio
2	Test the previous best scenario model by applying a combination of n-grams.	Get the type of n-grams with the best accuracy.
3	Test the previous best scenario model by applying various max features.	Get the max feature with the best accuracy.
4	Test the previous best scenario model by implementing feature expansion.	Get the best accuracy from the model by implementing feature expansion.
5	Test the previous best scenario model by implementing the ratio of the dropout layers.	Get the dropout ratio with the best accuracy.

3.1. Result

3.1.1. Scenario 1

In the first scenario, Boolean features as feature extraction with a max feature of 10000 on unigram data are used in the model to find split data with the F1-Score to determine the baseline. The ratio of splitting data is 90:10, 80:20, and 70:30. Table 6 is the result of scenario 1. The results of the first scenario obtained show that the ratio of 90:10 produces the best accuracy performance in all models with a score on the CNN model of 87.94%, the GRU model of 87.36%, the CNN+GRU model of 85.87%, and the GRU+CNN model of 86.41%. Based on this scenario, split data with a ratio of 90:10 will be used as a baseline for all models.

Table 6. Result of Scenario 1

Data Split	Accuracy (%)			F1-Score			AUROC		
	90:10	80:20	70:30	90:10	80:20	70:30	90:10	80:20	70:30
CNN	**87.94**	87.08	87.18	**0.8777**	0.8672	0.8693	**0.8794**	0.8687	0.8723
GRU	**87.36**	86.56	87.00	**0.8727**	0.8641	0.8655	**0.8733**	0.8656	0.8675
CNN+GRU	**85.87**	85.48	85.48	**0.8611**	0.8559	0.8528	**0.8591**	0.8548	0.8545
GRU+CNN	**86.41**	85.20	85.60	**0.8644**	0.8494	0.8556	**0.8638**	0.8526	0.8550

3.1.2. Scenario 2

The second scenario is carried out to find the best n-grams before performing feature extraction with Boolean. The N-grams tested are Unigram as a baseline, Bigram, Trigram, Unigram-Bigram, and Unigram-Bigram-Trigram. Table 7 is the result of scenario 2. The results of the second scenario showed that Unigram-Bigram produced the best accuracy improvement compared to other types of n-grams with an increase for the CNN model of 0.28%, GRU of 0.14%, CNN-GRU of 0.34%, and GRU+CNN of 0.08%. Based on Table 7 , Unigram-Bigram will be used for the following scenario.

Table 7. Result of Scenario 2

N-Grams	Accuracy (%)				
	Unigram	**Bigram**	**Trigram**	**Unigram-Bigram**	**Unigram-Bigram-Trigram**
CNN	87.94	77.66 (-11.69)	63.17 (-28.17)	**88.19 (+0.28)**	88.04 (+0.11)
GRU	87.36	77.20 (-11.63)	63.48 (-27.34)	**87.48 (+0.14)**	87.44 (+0.09)
CNN+GRU	85.87	75.88 (-11.63)	62.64 (-27.05)	**86.16 (+0.34)**	85.89 (+0.02)
GRU+CNN	86.41	76.24 (-11.77)	63.18 (-26.88)	**86.48 (+0.08)**	86.13 (-0.32)

3.1.3. Scenario 3

The third scenario looks for the best max features from the feature extraction of the Boolean. Max feature tested including 5000, 10000 (Baseline), 20000, and 30000. Table 8 is the result of scenario 3. The results of the third scenario show that the max feature of 5000 results in an increase in accuracy compared to other max features with a percentage increase of 0.19% for the CNN model, 0.30% for the GRU model, 0,08% for CNN+GRU, and 0.47% for GRU+CNN. Based on that, max feature 5000 will be used for the following scenario.

Table 8. Result of Scenario 3

Max Feature	Accuracy (%)			
	10000	**5000**	**20000**	**30000**
CNN	88.19	**88.36 (+0.19)**	87.67 (-0.59)	87.39 (-0.91)
GRU	87.48	**87.74 (+0.30)**	87.05 (-0.49)	86.94 (-0.62)
CNN+GRU	86.16	**86.23 (+0.08)**	85.66 (-0.58)	85.14 (-1.18)
GRU+CNN	86.48	**86.89 (+0.47)**	84.83 (-1.91)	85.34 (-1.32)

3.1.4. Scenario 4

The fourth scenario is implementing the feature expansion of the similarity corpus, built using the FastText model. The similarity is built from 3 datasets: Twitter, IndoNews, and Twitter+IndoNews. Each dataset has three tests, namely Top 1, Top 5, and Top 10. Table 9 is the test result from scenario 4. The results of scenario four show that in the CNN model, the most significant increase in the accuracy value is in the Tweet+IndoNews Corpus with Top 5 similarity with an increase of 0.49%. For the GRU model, the most significant increase is in the Tweet+IndoNews corpus in the Top 5 with an increase of 0.40%, the CNN+GRU model the biggest was in the IndoNews Corpus with Top 10 similarity with an increase of 1.44%, and the GRU+CNN model was in the Tweet+IndoNews corpus in Top 5 with an increase of 0.76%. Based on that, each model will use the best corpus for the following scenario.

Table 9. Result of Scenario 4

Feature Expansion	Best Scenario 3	Accuracy (%)								
		Tweets Corpus			**IndoNews Corpus**			**Tweets+IndoNews Corpus**		
		Top 1	**Top 5**	**Top 10**	**Top 1**	**Top 5**	**Top 10**	**Top 1**	**Top 5**	**Top 10**
CNN	88.36	88.46 (+0.11)	88.37 (+0.01)	88.37 (+0.01)	88.47 (+0.12)	88.38 (+0.02)	88.37 (+0.01)	88.57 (+0.24)	**88.79 (+0.49)**	88.39 (+0.03)
GRU	87.74	87.89 (+0.17)	87.96 (+0.25)	87.75 (+0.01)	87.96 (+0.25)	87.79 (+0.06)	88.08 (+0.39)	88.02 (+0.32)	**88.09 (+0.40)**	87.80 (+0.07)
CNN+GRU	86.23	87.16 (+1.08)	87.05 (+0.95)	87.01 (+0.90)	87.01 (+0.90)	86.60 (+0.43)	**87.47 (+1.44)**	87.05 (+0.95)	86.75 (+0.60)	87.13 (+1.04)
GRU+CNN	86.89	87.26 (+0.43)	87.21 (+0.37)	87.21 (+0.37)	87.24 (+0.40)	87.41 (+0.60)	87.25 (+0.41)	86.97 (+0.09)	**87.55 (+0.76)**	87.24 (+0.40)

3.1.5. Scenario 5

The last scenario, the baseline model + Unigram-Bigram + max feature + feature expansion that has been tested before, is used and added with several dropout percentages, including 20%, 40%, 50% (baseline), 60%, and 80%. Table 10 shows the result of the last scenario.

Table 10. Result of Scenario 5

Max Feature	Accuracy (%)				
	50	**20**	**40**	**60**	**80**
CNN	**88.79**	87.90 (-1.00)	88.43 (-0.41)	88.52 (-0.30)	88.13 (-0.74)
GRU	88.09	87.56 (-0.60)	**88.17 (+0.09)**	87.65 (-0.50)	83.10 (-5.66)
CNN+GRU	**87.47**	86.74 (-0.83)	86.20 (-1.45)	87.27 (-0.23)	84.71 (-3.16)
GRU+CNN	**87.55**	86.97 (-0.66)	87.22 (-0.38)	87.33 (-0.25)	87.40 (-0.17)

The results of the last scenario showed that the CNN, CNN+GRU, and GRU+CNN models decreased when the dropout ratio was changed, while for the GRU model, the dropout ratio of 40% increased the model by 0.09%.

3.2. Discussion

Determination of the baseline determined based on Table 6 shows that split data with 90% training and 10% testing at max feature 10000 in unigram data produces the best accuracy in each model. So that this setting can be taken as a baseline for all models. The use of n-grams has been tested to determine the best use of n-grams in the built model. Based on Table 7, it was found that for bigrams and trigrams, the accuracy decreased drastically, while for the combined n-grams (unigram-bigram and unigram-bigram-trigram), the accuracy increased. The use of the max feature has been carried out to find the best max feature. The determining of the features that are maintained is based on the ranking of the number of words so that the more these words appear, the higher the possibility of these words being maintained. Based on Table 8, 5000 max features can actually increase accuracy, so it can be concluded that the number of max features does not guarantee that it will increase accuracy. In fact, if many unimportant words appear, it will cause noise in the data. The use of feature expansion has been done to overcome the vocabulary mismatch. Based on Table 9, it is found that the use of feature expansion can increase the accuracy of the model even in all the schemes tested. What's more, the use of feature expansion has the highest increase in accuracy compared to the application of other scenarios. It can be concluded that feature expansion can overcome the problem of vocabulary mismatch and make the model able to understand similar vocabulary better so that it is not considered as a different word so that the words used for classification become more relevant.

Finally, use the layer dropout percentage. Based on Table 10, it can be seen that the use of the dropout layer percentage at the baseline is the best, except for the GRU model, which increases when the dropout percentage is 40%. The use of the dropout layer is intended as a way to overcome overfitting. Fig. 5 shows the relative increase of all test scenarios. Based on the results of the percentage increase, the use of feature expansion succeeded in producing the highest increase in accuracy compared to another scenario. The CNN+GRU model experienced the most significant percentage increase, namely 1.86% from the baseline model, followed by GRU+CNN at 1.32%, CNN at 0.97%, and GRU at 0.93%.

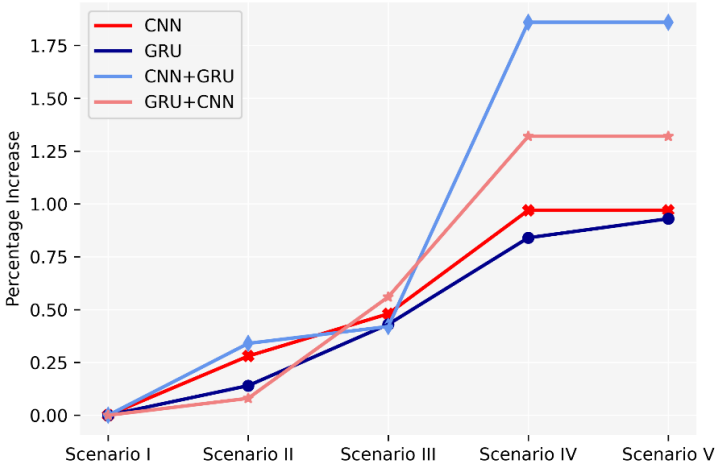

Fig. 5. Relative Increase of All Test Scenario

Statistical significance tests were used in this study to validate changes in accuracy. The significance used is the P-Value where if the P-Value < 0.05, then the comparison is statistically significant, and if the P-Value < 0.01, then the comparison is highly statistically significant. The Z-Value is also used with a 95% confidence level where if the Z-Value is > 1.96, then the comparison is statistically significant.

Table 11 is the result of a significant test where it is found that changes from scenario 3 to scenario 4 experience a statistically significant increase in all models. This proves that using feature expansion can improve the model's performance significantly. The change from scenario 1 (baseline model) to scenario 5 (best model), carried out in this study, experienced a highly statistically significant increase.

Table 12 shows the best Accuracy, F1-Score, and AUROC for each model tested, where CNN gets the best score compared to other models at baseline + Unigram-Bigram + max feature of 5000 + FastText Top 5 Tweet+IndoNews corpus + Dropout of 50% with accuracy 88.79%, F1-Score 0.8861, and AUROC 0.8879. followed by GRU at baseline + Unigram-Bigram + max feature of 5000 + FastText Top 5 Tweet+IndoNews corpus + Dropout of 40% with Accuracy 88.17%, F1-Score 0.8797, and AUROC 0.8817. CNN+GRU with baseline + Unigram-Bigram + max feature of 5000 + FastText Top 10 IndoNews corpus + Dropout of 50% with Accuracy 87.47%, F1-Score 0.8744, and AUROC 0.8748. GRU+CNN with baseline + Unigram-Bigram + max feature of 5000 + FastText Top 5 Tweet+IndoNews corpus + Dropout of 50% with Accuracy 87.55%, F1-Score 0.8742 , and AUROC 0.8756.

Table 11. Significant Test Result Between All Test Scenarios

		S1→S2	S2→S3	S3→S4	S4→S5	S1→S5
CNN	Z-Value	1.276	0.873	2.241	0.000	4.391
	P-Value	0.100	0.191	0.012	0.500	0.000
	Significant?	**False**	**False**	**True**	**False**	**True**
GRU	Z-Value	0.601	1.311	1.975	0.411	4.094
	P-Value	0.273	0.094	0.024	0.340	0.000
	Significant?	**False**	**False**	**True**	**False**	**True**
CNN+GRU	Z-Value	1.385	0.341	6.072	0	7.798
	P-Value	0.083	0.366	0.000	0.500	0.000
	Significant?	**False**	**False**	**True**	**False**	**True**
GRU+CNN	Z-Value	0.344	1.992	3.276	0	5.612
	P-Value	0.365	0.023	0.005	0.500	0.000
	Significant?	**False**	**True**	**True**	**False**	**True**

Table 12. Best Performance on Each Model

Performance	CNN	GRU	CNN+GRU	GRU+CNN
Accuracy	88.79 (+0.97)	88.17 (+0.93)	87.47 (+1.86)	87.55 (+1.32)
F1-Score	0.8861 (+0.96)	0.8797 (+0.80)	0.8744 (+1.54)	0.8742 (+1.13)
AUROC	0.8879 (+0.97)	0.8817 (+0.96)	0.8748 (+1.83)	0.8756 (+1.37)

Table 13 shows the comparison of the best performance results from several previous studies. Based on the best performance, the use of deep learning and feature expansion, as done in this study, is the highest performance compared to methods that use deep learning only or those that use feature expansion without deep learning.

Table 13. Comparison of Best Performance With Previous Studies on Hate Speech Detection

Ref	Best Model Classification	Feature Expansion	Dataset (Number of Tweets)	Best Accuracy (%)	Best F1-Score	Best AUROC
Anistya *et al.* [3]	RF	GloVe	20.601 (Indonesian)	88.59	-	-
Dewi *et al.* [12]	RF	Word2vec	20.571 (Indonesian)	88.37	-	-
Alshalan *et al.* [9]	CNN	-	9.316 (Arabic)	-	0.7900	0.8900
Duwairi *et al.* [10]	BiLSTM-CNN	-	9.833 (Arabic)	81.00	-	-
This Study	CNN	FastText	63.984 (Indonesian)	88.79	86.43	0.8879

4. CONCLUSION

This study has conducted hate speech detection on Tweets, using multiple deep learning models: CNN, GRU, CNN+GRU, and GRU+CNN. This research uses boolean representation as feature extraction and FastText to create a corpus similarity with the Top 1, Top 5, and Top 10 most similar built from Tweets with a total of 63.984, IndoNews with a total of 119.488 data, and Tweets-IndoNews, which are used for feature expansion. This study uses five scenarios: best data split, n-grams, the max feature, feature expansion, and dropout percentage.

Based on the results of the tests in this study, it was found that the best split data was found in the proportion of 90:10, the best n-grams used Unigram+Bigram, the best max feature was at 5000, and the best dropout percentage was at 50% for CNN, CNN+GRU, and GRU +CNN while for GRU at 40%. And also, the use of feature expansion has succeeded in increasing the highest accuracy compared to other scenarios. Corpus IndoNews with Top 10 similarity can increase the greatest accuracy on CNN+GRU, and for CNN, GRU, and GRU+CNN, a significant increase in accuracy occurs on corpus Tweets+IndoNews Top 5 similarity.

Based on the results of all scenarios can be concluded that the method used in this study produces outstanding performance with highly statistical significance, where CNN has the largest accuracy with a score of 88.79% and an increase of 0.97% from the baseline model, followed by GRU with an accuracy of 88.17% with an increase of 0.93%, CNN+GRU with an accuracy of 87.47% with an increase of 1.86% and GRU+CNN with a score of 87.55% with an increase of 1.32%. This increased accuracy can help make hate speech detection systems more precise because the system can avoid vocabulary mismatches.

The limitation of this study is that the dataset is used in Indonesian. So suggestions for further research, researchers can create a better hybrid deep learning model by using various features extracted, such as TF-IDF, and feature expansion, such as GloVe, Word2vec, etc. Also, use the dataset with another language to see the credibility of the use of feature expansion.

Acknowledgments

The author's praise and gratitude to Allah SWT, who has provided convenience for this research and journal preparation. The authors also thank the parents who always pray and the Supervising Professors who provided advice and knowledge to complete this research with full preparation.

REFERENCES

[1] F. Husain and O. Uzuner, "A Survey of Offensive Language Detection for the Arabic Language," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 20, no. 1. Association for Computing Machinery, pp. 1–44, 2021. https://doi.org/10.1145/3421504.

[2] J. W. Howard, "Free speech and hate speech," *Annual Review of Political Science*, vol. 22, pp. 93–109, 2019, https://doi.org/10.1146/annurev-polisci-051517-012343.

[3] 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, https://doi.org/10.29207/resti.v5i6.3521.

[4] A. B. Boot, E. Tjong Kim Sang, K. Dijkstra, and R. A. Zwaan, "How character limit affects language usage in tweets," *Palgrave Commun*, vol. 5, no. 1, 2019, https://doi.org/10.1057/s41599-019-0280-3.

[5] 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)*, pp. 1–5, 2016, https://doi.org/10.1109/TSSA.2016.7871085.

[6] Y. Zhou, Y. Yang, H. Liu, X. Liu, and N. Savage, "Deep Learning Based Fusion Approach for Hate Speech Detection," *IEEE Access*, vol. 8, pp. 128923–128929, 2020, http://dx.doi.org/10.1109/ACCESS.2020.3009244.

[7] P. Fortuna and S. Nunes, "A Survey on Automatic Detection of Hate Speech in Text," *ACM Computing Surveys*, vol. 51, no. 4, 2018, https://doi.org/10.1145/3232676.

[8] G. O. Ganfure, "Comparative analysis of deep learning based Afaan Oromo hate speech detection," *J Big Data*, vol. 9, no. 1, 2022, https://doi.org/10.1186/s40537-022-00628-w.

[9] R. Alshalan and H. Al-Khalifa, "A Deep Learning Approach for Automatic Hate Speech Detection in the Saudi Twittersphere," *Applied Sciences (Switzerland)*, vol. 10, no. 23, pp. 1–16, 2020, https://doi.org/10.3390/app10238614.

[10] R. Duwairi, A. Hayajneh, and M. Quwaider, "A Deep Learning Framework for Automatic Detection of Hate Speech Embedded in Arabic Tweets," *Arab J Sci Eng*, vol. 46, no. 4, pp. 4001–4014, 2021, https://doi.org/10.1007/s13369-021-05383-3.

[11] Z. Zhang, D. Robinson, and J. Tepper, "Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 15, pp. 745–760, 2018, https://doi.org/10.1007/978-3-319-93417-4_48.

[12] M. P. K. Dewi and E. B. Setiawan, "Feature Expansion Using Word2vec for Hate Speech Detection on Indonesian Twitter with Classification Using SVM and Random Forest," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 6, no. 2, p. 979, 2022, http://dx.doi.org/10.30865/mib.v6i2.3855.
[13] H. R. Alhakiem and E. B. Setiawan, "Aspect-Based Sentiment Analysis on Twitter Using Logistic Regression with FastText Feature Expansion," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 5, pp. 840–846, 2022, https://doi.org/10.29207/resti.v6i5.4429.
[14] U. Haruna, R. Ali, and M. Man, "A new modification CNN using VGG19 and ResNet50V2 for classification of COVID-19 from X-ray radiograph images," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 31, no. 1, p. 369, 2023, http://doi.org/10.11591/ijeecs.v31.i1.pp369-377.
[15] H. G. Ariswati and L. Soetjiatie, "Aura detection using thermal camera with convolutional neural network method for mental health diagnosis," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 31, no. 1, p. 553, 2023, http://doi.org/10.11591/ijeecs.v31.i1.pp553-561.
[16] F. D. Mirajkar, R. Fatima, and S. A. Qadeer, "Content-based image retrieval using integrated dual deep convolutional neural network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 31, no. 1, p. 77, 2023, http://doi.org/10.11591/ijeecs.v31.i1.pp77-87.
[17] C. R. Kumar, S. N, M. Priyadharshini, D. G. E, and K. R. M, "Face recognition using CNN and siamese network," *Measurement: Sensors*, vol. 27, p. 100800, 2023, https://doi.org/10.1016/j.measen.2023.100800.
[18] S. Gonwirat and O. Surinta, "DeblurGAN-CNN: Effective Image Denoising and Recognition for Noisy Handwritten Characters," *IEEE Access*, vol. 10, pp. 90133–90148, 2022, https://doi.org/10.1109/ACCESS.2022.3201560.
[19] Q.-H. Vo, H.-T. Nguyen, B. Le, and M.-L. Nguyen, "Multi-channel LSTM-CNN model for Vietnamese sentiment analysis," in *2017 9th International Conference on Knowledge and Systems Engineering (KSE)*, pp. 24–29, 2017, http://dx.doi.org/10.1109/KSE.2017.8119429.
[20] N. Nedjah, I. Santos, and L. de Macedo Mourelle, "Sentiment analysis using convolutional neural network via word embeddings," *Evol Intell*, vol. 15, no. 4, pp. 2295–2319, 2022, https://doi.org/10.1007/s12065-019-00227-4.
[21] R. Rachidi, M. A. Ouassil, M. Errami, B. Cherradi, S. Hamida, and H. Silkan, "Classifying toxicity in the Arabic Moroccan dialect on Instagram: a machine and deep learning approach," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 31, no. 1, pp. 588–598, 2023, http://doi.org/10.11591/ijeecs.v31.i1.pp588-598.
[22] X. Luo, Z. Yu, Z. Zhao, W. Zhao, and J.-H. Wang, "Effective short text classification via the fusion of hybrid features for IoT social data," *Digital Communications and Networks*, vol. 8, no. 6, pp. 942–954, 2022, https://doi.org/10.1016/j.dcan.2022.09.015.
[23] X. Chen, P. Cong, and S. Lv, "A Long-Text Classification Method of Chinese News Based on BERT and CNN," *IEEE Access*, vol. 10, pp. 34046–34057, 2022, https://doi.org/10.1109/ACCESS.2022.3162614.
[24] P. K. Roy, A. K. Tripathy, T. K. Das, and X.-Z. Gao, "A Framework for Hate Speech Detection Using Deep Convolutional Neural Network," *IEEE Access*, vol. 8, pp. 204951–204962, 2020, https://doi.org/10.1109/ACCESS.2020.3037073.
[25] M. Sajjad *et al.*, "A Novel CNN-GRU-Based Hybrid Approach for Short-Term Residential Load Forecasting," *IEEE Access*, vol. 8, pp. 143759–143768, 2020, https://doi.org/10.1109/ACCESS.2020.3009537.
[26] A. Al Wazrah and S. Alhumoud, "Sentiment Analysis Using Stacked Gated Recurrent Unit for Arabic Tweets," *IEEE Access*, vol. 9, pp. 137176–137187, 2021, https://doi.org/10.1109/ACCESS.2021.3114313.
[27] S. S. Sohail *et al.*, "Crawling Twitter data through API: A technical/legal perspective," *CoRR*, vol. 2105.10724, 2021, https://doi.org/10.1109/ACCESS.2021.3114313.
[28] H. S. Alatawi, A. M. Alhothali, and K. M. Moria, "Detecting White Supremacist Hate Speech using Domain Specific Word Embedding with Deep Learning and BERT," *IEEE Access*, vol. 9, pp. 106363–106374, 2020, https://doi.org/10.1109/ACCESS.2021.3100435.
[29] C.-C. Wang, M.-Y. Day, and C.-L. Wu, "Political Hate Speech Detection and Lexicon Building: A Study in Taiwan," *IEEE Access*, vol. 10, pp. 44337–44346, 2022, https://doi.org/10.1109/ACCESS.2022.3160712.
[30] J. Hartmann, J. Huppertz, C. Schamp, and M. Heitmann, "Comparing automated text classification methods," *International Journal of Research in Marketing*, vol. 36, no. 1, pp. 20–38, 2019, https://doi.org/10.1016/j.ijresmar.2018.09.009.
[31] N. M. Azahra and E. B. Setiawan, "Sentence-Level Granularity Oriented Sentiment Analysis of Social Media Using Long Short-Term Memory (LSTM) and IndoBERTweet Method," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 1, pp. 85–95, 2023, http://dx.doi.org/10.26555/jiteki.v9i1.25765.
[32] N. Badri, F. Kboubi, and A. H. Chaibi, "Combining FastText and Glove Word Embedding for Offensive and Hate speech Text Detection," *Procedia Comput Sci*, vol. 207, pp. 769–778, 2022, https://doi.org/10.1016/j.procs.2022.09.132.
[33] A. Amalia, D. Gunawan, Y. Fithri, and I. Aulia, "Automated Bahasa Indonesia essay evaluation with latent semantic analysis," *J Phys Conf Ser*, vol. 1235, no. 1, p. 012100, 2019, http://dx.doi.org/10.1088/1742-6596/1235/1/012100
[34] A. O. Salau and S. Jain, "Feature Extraction: A Survey of the Types, Techniques, Applications," in *2019 International Conference on Signal Processing and Communication (ICSC)*, pp. 158–164, 2019, https://doi.org/10.1109/ICSC45622.2019.8938371.

[35] S. Chotirat and P. Meesad, "Part-of-Speech tagging enhancement to natural language processing for Thai wh-question classification with deep learning," *Heliyon*, vol. 7, no. 10, 2021, https://doi.org/10.1016/j.heliyon.2021.e08216.
[36] S. Shumaly, M. Yazdinejad, and Y. Guo, "Persian sentiment analysis of an online store independent of pre-processing using convolutional neural network with fastText embeddings," *PeerJ Comput Sci*, vol. 7, p. 422, 2021, http://dx.doi.org/10.7717/peerj-cs.422.
[37] I. N. Khasanah, "Sentiment Classification Using fastText Embedding and Deep Learning Model," *Procedia Comput Sci*, vol. 189, pp. 343–350, 2021, https://doi.org/10.1016/j.procs.2021.05.103.
[38] K. Sreelakshmi, B. Premjith, and K. P. Soman, "Detection of Hate Speech Text in Hindi-English Code-mixed Data," *Procedia Comput Sci*, vol. 171, pp. 737–744, 2020, https://doi.org/10.1016/j.procs.2020.04.080.
[39] S. Ghosal and A. Jain, "Depression and Suicide Risk Detection on Social Media using fastText Embedding and XGBoost Classifier," *Procedia Comput Sci*, vol. 218, pp. 1631–1639, 2023, https://doi.org/10.1016/j.procs.2023.01.141 .
[40] W. Lu, L. Ma, H. Chen, X. Jiang, and M. Gong, "A Clinical Prediction Model in Health Time Series Data Based on Long Short-Term Memory Network Optimized by Fruit Fly Optimization Algorithm," *IEEE Access*, vol. 8, pp. 136014–136023, 2020, https://doi.org/10.1109/ACCESS.2020.3011721.
[41] C. Yu, R. Han, M. Song, C. Liu, and C.-I. Chang, "A Simplified 2D-3D CNN Architecture for Hyperspectral Image Classification Based on Spatial–Spectral Fusion," *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 13, pp. 2485–2501, 2020, https://doi.org/10.1109/JSTARS.2020.2983224.
[42] S. Ghimire, Z. M. Yaseen, A. A. Farooque, R. C. Deo, J. Zhang, and X. Tao, "Streamflow prediction using an integrated methodology based on convolutional neural network and long short-term memory networks," *Sci Rep*, vol. 11, no. 1, p. 17497, 2021, https://doi.org/10.1038/s41598-021-96751-4.
[43] M. U. Salur and I. Aydin, "A Novel Hybrid Deep Learning Model for Sentiment Classification," *IEEE Access*, vol. 8, pp. 58080–58093, 2020, https://doi.org/10.1109/ACCESS.2020.2982538.
[44] R. Dey and F. M. Salem, "Gate-variants of Gated Recurrent Unit (GRU) neural networks," in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1597–1600, 2017, https://doi.org/10.1109/MWSCAS.2017.8053243.
[45] M. U. Salur and I. Aydin, "A Novel Hybrid Deep Learning Model for Sentiment Classification," *IEEE Access*, vol. 8, pp. 58080–58093, 2020, https://doi.org/10.1109/ACCESS.2020.2982538.
[46] C. Zeng, C. Ma, K. Wang, and Z. Cui, "Parking Occupancy Prediction Method Based on Multi Factors and Stacked GRU-LSTM," *IEEE Access*, vol. 10, pp. 47361–47370, 2022, https://doi.org/10.1109/ACCESS.2022.3171330.
[47] D. Valero-Carreras, J. Alcaraz, and M. Landete, "Comparing two SVM models through different metrics based on the confusion matrix," *Comput Oper Res*, vol. 152, p. 106131, 2023, https://doi.org/10.1016/j.cor.2022.106131.
[48] S. Khan *et al.*, "BiCHAT: BiLSTM with deep CNN and hierarchical attention for hate speech detection," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 7, pp. 4335–4344, 2022, https://doi.org/10.1016/j.jksuci.2022.05.006.

BIOGRAPHY OF AUTHORS

Kevin Usmayadhy Wijaya, is a final-year student currently pursuing a bachelor's degree in computer science at Telkom University, Bandung, Indonesia. He is interested in the field of Artificial Intelligence and Data. Email: kevinusmayadhyw@student.telkomuniversity.ac.id.

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