Topic Detection on Twitter Using Deep Learning Method with Feature Expansion GloVe

Windy Ramadhanti, Erwin Budi Setiawan
Telkom University, Jl. Terusan Buah Batu, Bandung 40257, Indonesia

ARTICLE INFO

Article history:

Received July 21, 2023
Revised August 04, 2023
Published August 15, 2023

Keywords:

Topic Classification;
Twitter;
Tweet;
CNN;
RNN;
GloVe

ABSTRACT

Twitter is a medium of communication, transmission of information, and exchange of opinions on a topic with an extensive reach. Twitter has a tweet with a text message of 280 characters. Because text messages can only be written briefly, tweets often use slang and may not follow structured grammar. The diverse vocabulary in tweets leads to word discrepancies, so tweets are difficult to understand. The problem often found in classifying topics in tweets is that they need higher accuracy due to these factors. Therefore, the authors used the GloVe feature expansion to reduce vocabulary discrepancies by building a corpus from Twitter and IndoNews. Research on the classification of topics in previous tweets has been done extensively with various Machine Learning or Deep Learning methods using feature expansion. However, To the best of our knowledge, Hybrid Deep Learning has not been previously used for topic classification on Twitter. Therefore, the study conducted experiments to analyze the impact of Hybrid Deep Learning and the expansion of GloVe features on classification topics. The total data used in this study was 55,411 datasets in Indonesian-language text. The methods used in this study are Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Hybrid CNN-RNN. The results show that the topic classification system with GloVe feature expansion using the CNN method achieved the highest accuracy of 92.80%, with an increase of 0.40% compared to the baseline. The RNN followed it with an accuracy of 93.72% and a 0.23% improvement. The CNN-RN Hybrid Deep Learning model achieved the highest accuracy of 94.56%, with a significant increase of 2.30%. The RNN-CNN model also achieved high accuracy, reaching 94.39% with a 0.95% increase. Based on the accuracy results, the Hybrid Deep Learning model, with the addition of feature expansion, significantly improved the system's performance, resulting in higher accuracy.

Corresponding Author:

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

1. INTRODUCTION

Twitter is one of the most popular social networks online as a micro-blogging system [1]. Twitter can be used as a medium of information transmission and a place to freely express opinions on a topic, and it has widespread access [2]. Users can share articles or upload about anything happening in the world, so users can freely follow the topics being discussed. It becomes important because it can be used as a source of information in evaluating user responses to discussed topics, so it has a wide range of topics [3]–[5]. Word embedding was previously used in research classification [6]–[8]. Topic classification effectively explores data, links similar documents, and meaningful classification [9].

Twitter is unique, as this social media platform can only write text-based messages that do not exceed 280 characters [10]. Unlike other social media that have longer writing boundaries. Because tweets have short written limitations, tweets are often written in an unstructured grammar, using slang language, or even the content of tweets is irrelevant to a topic [5]. The use of unexplained grammar and varied vocabulary makes tweets difficult to detect, so it's a unique challenge to classify the topic on a tweet so that the information or discussion is relevant. Therefore, the extension of features in this research is used to address the problem.

Research on the classification of topics is multilabel, so the method used is machine learning or deep learning. The study [5] performed topic detection on Twitter using the Gradient Boosted Decision Tree method. In this study, the authors used the extraction of the TF-IDF feature and the expansion of the FastText feature, with an accuracy of 91.39% and an F1 score of 91.44%. Twitter used data of 30360 data and a news corpus of 97.794 data.

Similar research was carried out in the study [8] employing Word2vec for feature expansion and TF-IDF for feature extraction to extend topic detection using the Gradient Boosted Decision Tree. The data used in this research are tweets and news data. Twitter data consists of 35,605 data and news data of 142,544 data. The results of this study obtained an accuracy of 85.44%.

Research [11] conducted testing by classifying the text as an essay. This study showed that the accuracy of the results of the RNN algorithm obtained a higher accuracy compared to the CNN algorithm. RNN gets 55% accuracy, while CNN receives 50%. The study used 2000 data essays. There is no such thing as a shortage or a scale of the lowest scale.

The main contribution to this research is to apply hybrid deep learning methods with the expansion of features to classify topics. As far as the researchers searched, no study of the classification of topics using Hybrid Deep Learning was found. Therefore, the study aims to analyze the influence of the hybrid deep learning merger of the CNN and RNN algorithms as topic classification by adding the GloVe feature expansion.

The combination of the CNN and RNN algorithms is chosen because the combinations of these two methods can handle text with different lengths so that the classification process is not limited to letters alone, as well as capable of dealing with complex and non-explicit features so that better decisions can be made [5]. The purpose of expanding the GloVe feature is to address word discrepancies in tweets. GloVe employs matrix factoring to capture word co-occurrence information and global statistics, enabling semantic relationships between words in documents [12], [13]. In this study, the data used is social media Twitter using Indonesian. A total of 55,411 datasets are used and then further processed using the Term Frequency - Inverse Document Frequency (TF-IDF) extraction feature and the expansion feature using the word embedding method of GloVe.

2. METHODS

Fig. 1 shows the system's design built on this study. Starting from data collection, preprocessing, application extraction and feature expansion, sampling, model classification, and metrics evaluation.

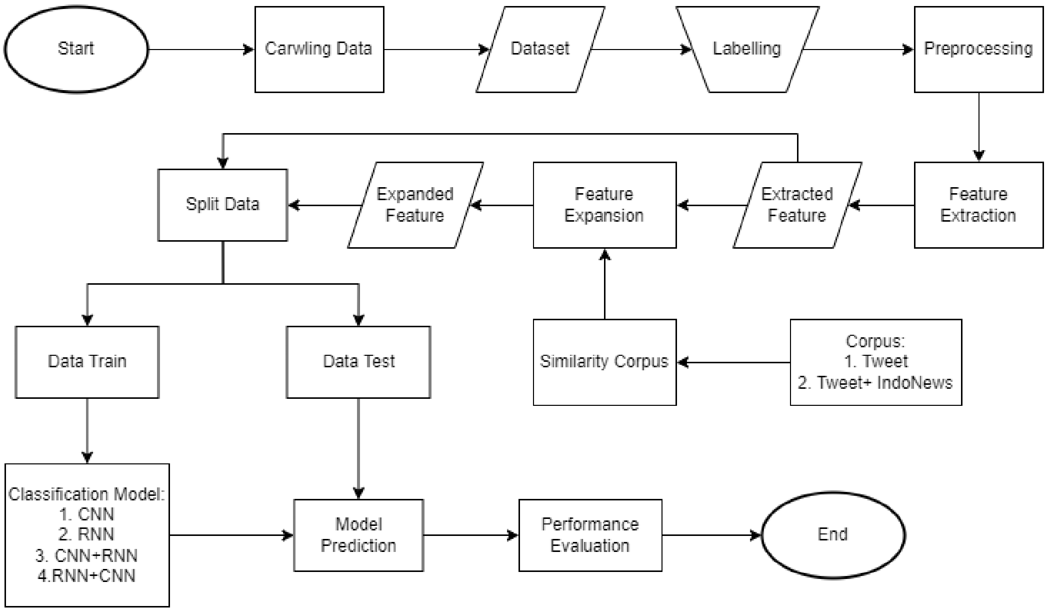


Fig. 1. Topic Detection System

2.1. Crawling Data

The data collected was obtained using the snscrape library from the Python programming language [14], [15]. The data used in this study is Indonesian-language Twitter data. Twitter data collection is taken based on specific keywords for tweet searches. Each time the crawling process will collect as much data as 1000 tweets. The data used was taken from January 2022–March 2023. A total of 56,236 Twitter datasets were collected. However, there was a decrease in data sets due to data duplication, so the number of datasets was 55,411 as the final data set before entering the preprocessing process. Table 1 is the data distribution of the crawling results.

Table 1. Data Distribution

Label	Amount	Percentage
Business	7,190	13%
Health	6,030	11%
Sport	5,978	11%
Education	5,898	10%
Automotive	5,718	10%
Entertainment	5,527	10%
Economy	5,384	10%
Lifestyle	4,905	9%
Technology	4,803	8%
Travel	3,978	8%
Total	**55,411**	**100%**

2.2. Labelling

Data labelling is done against data sets before the classification process. This tagging is done to distinguish the topic from each tweet on the dataset. In this study, there were ten different labels for each tweet. The labels are based on the most popular topics that often appear on several news portals in Indonesia. Each topic frequently appearing on several news sites is used as a reference label to classify topics in Twitter data. After examining the most popular topics on the Indonesian news portal, ten labels were taken for use in this study. The label used Business, Health, Sports, Education, Automotive, Entertainment, Economy, Lifestyle, Technology, and Travel. In this study, the process of manually labelling data [16]. To ensure the accuracy of the labelling results, each data is checked by at least three people. The method of majority votes is used in decision-making when there is disagreement in the marking process [17]. Examples of labels can be seen in Table 2.

Table 2. Example of a label on a tweet

Tweet	Label
Gimana mau keringetan kalo tiap olahraga dikit dikit cek hape lalu kebanyakan selfie. Olahraganya cuman 5 menit, duduk scroll hpnya lebih dari 1 jam, terus ngeluh udah rajin olahraga tapi badan kok gak bagus bagus, pikirin aja sendiri...	Sports
Kita dapat menyadari bahwa pijat dapat mengobati penyakit, bahkan jika kita tidak sakit dan kita secara teratur memijat selama 10-20 menit setiap hari, itu akan membantu menjaga kesehatan tubuh kita.	Health
Kenalin, #robot ikan ini bernama Gillbert! 🐟 Walaupun kecil, Gilbert mengemban tugas yang sangat besar, yaitu mengumpulkan sampah #mikroplastik 🗑️ 🌊 yang mencemari lautan. Yuk lihat bagaimana Gillbert beraksi! 👇	Technology

2.3. Data Preprocessing

Data preprocessing is a method of preparing data before it is processed. Incomplete and inconsistent raw data will be converted into a machine-understood format [18]. Text preprocessing involves processing and preparing text data before the analysis. These steps include data cleaning, converting letters into lowercase, dividing the text into words or tokens, filtering or removing irrelevant or meaningless words, and changing words into their basic form by deleting affixations [19], [20]. To help in the preprocessing process, the NLTK (Natural Language Toolkit) Python library enables tagging, stemming, classification, tokenization, and

Capabilities for parsing and semantic analysis [21], in addition to using the Pysastrawi library to assist in the stemming process [19].

The data cleaning includes removing URLs, hashtags, numbers, emoticons, and reading marks. Case folding is converting all the words with capital letters into small letters. This process is beneficial if there are variations in the use of capitalization on certain words [22]. Stop words are the process of filtering or removing words with no significant meaning, such as general words that are not included in the index or cannot be searched in computer search engines [23]. Examples of words included in a list of stopword include related words such as "dan," "atau," and "yang," etc. [23].

Stemming means removing the word reward as a basic word. The stemming process can be done by cutting off the prefixes, endings, or combinations of both. Using stemming, variations of words with the same word root are considered similar tokens. Stemming also uses a normalization process in which the slang words used in the Internet language are converted into basic word forms [5]. Tokenization is the process of dividing sentences into words, phrases, and symbols called tokens [24]. The tokens generated will help in the parsing and processing of data. The development of tokenization in this context separated the character series into basic processing units and interpreted and grouped isolated tokens to form higher-level tokens. Subsequently, raw text is processed and divided into smaller units [25].

2.4. Feature Extraction

Feature extraction is a method used to calculate the value of a feature in a document and is one of the most important techniques in data processing and text classification [26], [27]. In the feature extraction phase, a tweet representation will be performed. The tweet representation in this study uses a boolean vector feature with a fixed length with each feature that will indicate the presence or absence of a word in the tweet. This tweet representation started using the N-gram feature, including unigram, bigram, and trigram. The n-gram-based technique is very suitable for text classification, especially for language categorization [28]. Using N-grams allows the model to capture local patterns and context of words in the text, thereby improving accuracy in classifying tweets into appropriate categories in language analysis tasks. By leveraging the N-gram feature, the model can recognize combinations of words that often appear together in tweets, providing important information for further classification and understanding of the language used in those texts. For example, for word representations using the N-gram feature on a unigram, a feature vector with a length of 6 encodes the appearance of words in the order “saya”, “minum”, “enak”, “pagi”, “kopi”, and “hari”. Then tweets that contain “saya”, “minum”, “kopi”, “pagi”, “hari”, will appear as {1, 1,0, 1, 1, 1}. Examples of word representations using the N-gram feature can be seen in Fig. 2.

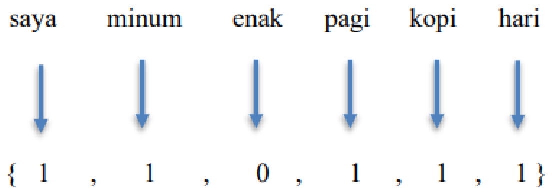


Fig. 2. Example of Tweet Representation

This study used TF-IDF (Term Frequency-Inverse Document frequency) as a feature extraction method. TF-IDF calculates the weight or value of each word (token) used on a document in the corpus. This method is widely used in information search and text development to evaluate the relationship of each word in a document set relationship [29]. This normalization process determines the weight of the terms often appearing in a document. Simply put, the document was converted to a weight calculated based on the number of its appearances [30]. This method is used to find out how often this word appears on tweet documents. TF-IDF can be formulated as follows:

$$w_{ij} = tf_{ij} \times IDF_j, with\ IDF_j \left(\log \frac{N}{df}\right) \tag{1}$$

Where w_{ij} is the weight of the word-j document, tf_{ij} represents the number of word appearances that are sought on the document, IDF_{ij} Inverse Document Frequency, N the total of documents, and df is the sum of all documents that contain the word search [31].

2.5. Feature Expansion

GloVe is one of the unsupervised learning algorithms for obtaining vector representations for words [32]. Word embedding is considered an efficient and effective method in learning vector representation of words. The advantage of the GloVe model is its ability to be trained quickly on larger amounts of data because its implementation can be parallelized [3], [32], [33]. Fig. 3 is an example of vector relations captured by GloVe.

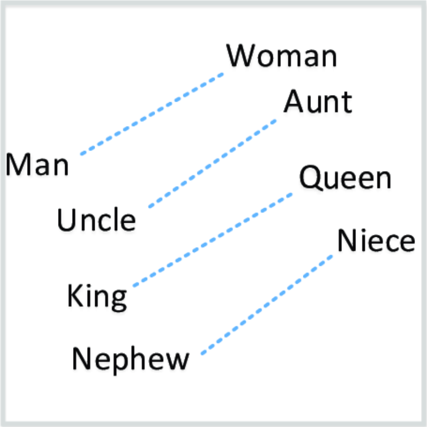


Fig. 3 Vector relations captured by GloVe [34]

The objective of using the GloVe method for word representation is to capture semantic relationships between words based on their common co-occurrence in the corpus [33]. GloVe uses a global matrix factoring method, which is a matrix that represents the presence or absence of words in a document [13]. GloVe studies relationships between words by calculating how often the words appear together in a particular corpus. The probability ratio of word appearance can encode a form of meaning and helps improve performance in word analogy problems [13]. The corpus of this study consists of tweets and IndoNews. The tweet corpus uses tweet data, while the tweet + IndoNews corpus is constructed using the combined tweet+IndoNews. Table 3 is an example of the similarity of the word “kanan” based on the corpus of Tweets with the top 15 similarities.

Table 3. Example of Top 15 Similarity of "kanan" words based on Corpus Tweets

Word	Rank 1	Rank 2	Rank 3
kanan	kiri	belah	motorik
	Rank 4	**Rank 5**	**Rank 6**
	dada	lajur	miring
	Rank 7	**Rank 8**	**Rank 9**
	Jalar	bahu	telinga
	Rank 10	**Rank 11**	**Rank 12**
	lengan	mules	bengkok
	Rank 13	**Rank 14**	**Rank 15**
	encok	tusuk	keseleo

2.6. Convolutional Neural Network

Convolutional Neural Network (CNN) is a type of simulated neural network in deep learning inspired by the human brain [35]. CNN is a type of simulated neural network architecture that can be trained in several stages and has been developed specifically for classification tasks [36]. Although its primary use was for image classification in computer vision, CNN also performed well in text classification tasks [36]. If an image is an input for the image classification, then the input for the text classification is a word vector created using word embedding [37]. Fig. 4 shows an overview of the CNN architectural design for text processing.

Several layers in the CNN model built on this research were created using the TensorFlow Keras library, including the input layer, the convolutional layer, the dense layer, the max pooling layer, the dropout layer, and the flatten layer. The input layer is a layer for entering data inputs that will be continued onto the convolutional layer [39]. This study used 1D convolutional layer because the data used is one-dimensional text data. The model can extract these patterns and more sophisticated features from the upper layers using 1D-CNN to identify simple ways in our data [40]. Layer Conv1D is used with 64 filters, each filter has a 3-kernel size and uses the ReLU activation function.

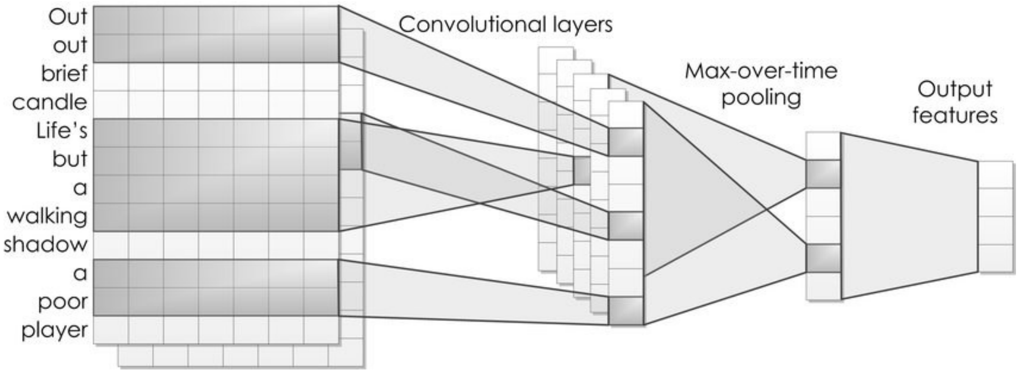


Fig. 4. CNN Architecture [38]

After the Conv1D layer, the Model has a dense layer with 32 units and uses the ReLU activation function. The dense layer connects each neuron on the previous layer with the neuron layer on the fully connected layer. This creates connections between all the neurons in the layer and allows the nerve tissue to study the more complex relationships between the features.

There is a Max Pooling1D layer. This layer performs the max pooling operation on data passed through the previous layer. Max pooling reduces the spatial dimension of the data by taking the maximum value from the pooling window. After the MaxPooling1D layer, there is a Dropout layer with a dropout rate 0.1. Dropout layers prevent overfitting by randomly turning off some neurons on the previous layer during training.

After the Dropout layer, the model has a Dense layer with 10 units and uses the ReLU activation function. A flatten layer converts one dimension of a feature matrix into a vector that continues a fully connected layer [39]. After the Flatten layer, the model has a Dense layer with 10 units and uses the softmax activation function. The optimization used is Adam loss function categorical cross entropy.

2.7. Recurrent Neural Network

A feed-forward neural network focusing on modelling in the temporal domain is known as a recurrent neural network (RNN) [41]. The ability of RNNs to send information over time is one of their distinguishing characteristics [41]–[43]. To connect the time steps that support training in the temporal domain with the exploitation of the sequential properties of the input, RNN has an additional parameter matrix in its structure [43], [44]. The RNN (Recurrent Neural Network) model has demonstrated a solid ability to learn various natural language processing tasks [45]. This model has good characteristics in modelling sequential data and leveraging sequential information to the maximum [45]. However, in using basic RNNs, there are some challenges, such as the problem of vanishing gradient, which may limit its ability to capture long-term dependencies [46]. To address this problem, the study used LSTM variants to initialize dynamic weights [47]. This variation of LSTM is used to remember long-term information with weight as Long-term memory [48].

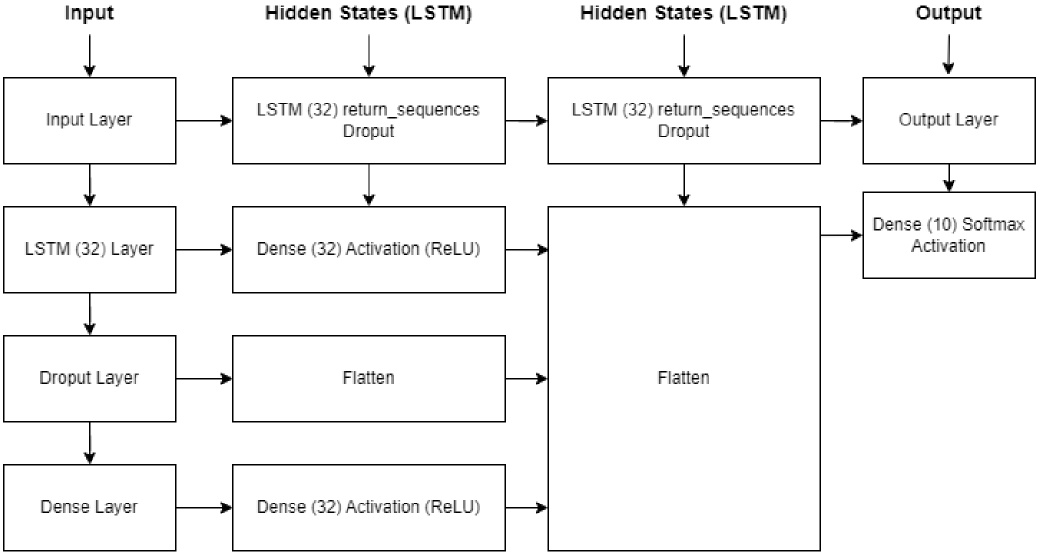

Input
Hidden States (LSTM)
Hidden States (LSTM)
Output
Input Layer
LSTM (32) return_sequences Dropout
LSTM (32) return_sequences Dropout
Output Layer
LSTM (32) Layer
Dense (32) Activation (ReLU)
Dense (10) Softmax Activation
Dropout Layer
Flatten
Flatten
Dense Layer
Dense (32) Activation (ReLU)

Fig. 5 RNN Architecture

Fig. 5 is the architectural design of RNN built on this research. The LSTM layer has 32 hidden units. A drop-out layer of 0.5 is used to prevent overfitting. The dense layer has 32 units with a ReLU activation function, and the last Dense Layer uses 10 units and softmax activation functions. The optimization is Adam, and categorical cross entropy is the Loss Function.

2.8. Hybrid

The hybrid model built on this study combined two previous methods, CNN and RNN. This model is built using TensorFlow Keras. For the hybrid model made in this study, there are two models, CNN-RNN and RNN-CNN. Fig. 6 shows the architecture of hybrid CNN-RNN.

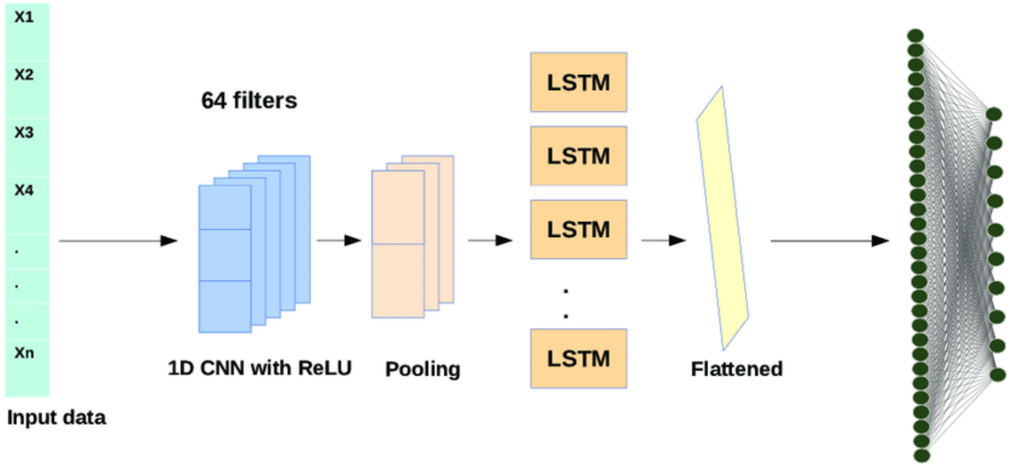


Fig. 6 Hybrid CNN-RNN Architecture [49]

For the CNN-RNN model, the first layer is the Convolution layer 1 Dimension with 32 filters. Each filter has 3 kernel sizes and uses the ReLU activation function. Padding is used to keep the output size equal to the input size. The L2 regulation with the parameter 0.01 is also applied to the Conv1D kernel to reduce overfitting.

After the Conv1D layer, the model has a Dense layer with 32 units and uses the ReLU activation function. This solid layer performs operations that are fully connected between the previous and next layers. After the Dense layer, there is the MaxPooling1D layer. This layer performs the combined operation max on the data passed through the previous layer. Max pooling reduces the spatial dimension of the data by taking the maximum value from the pooling window.

After the MaxPooling1D layer, there is the RNN layer in the variant LSTM. The RNN layer has 32 hidden units and produces outputs at each time step. LSTM is used to study sequence patterns in data. After the RNN layer, there is the Flatten layer. This layer converts the output from the previous layer into a 1-Dimensional vector to be used as an input for the next solid layer. After the Flatten layer, this model has a Dense layer with 10 units and uses the softmax activation function. This solid layer produces outputs with shapes that match the number of classes to be predicted, in this case, 10 classes.

Once a model is defined, the model is compiled using the Adam optimizer. The loss function is categorical cross entropy, suitable for multi-class classification cases. Accuracy metrics are also used to monitor model performance during training. The layer on the RNN-CNN model is the same as CNN-RNN, but the difference lies in the 3 initial layers only. On the first layer, use layer RNN, the second layer Conv1D, and the third layer maxpooling.

This study's use of the CNN-RNN and RNN-CNN hybrid model has weaknesses and challenges. The combination of these two architectures leads to the high complexity of the model and requires complex parameter settings. Problems with disappearing and exploding gradients may also arise when using RNNs in models. In addition, combining representations from CNN and RNN can be challenging, given that both types of architectures focus on different aspects of data. However, this challenge can be overcome by exploring different architectures, setting parameters carefully, and using regulatory techniques.

2.9. Performance Evaluation

Only use accuracy values to calculate the system’s performance in this study. The accuracy value can be obtained based on four terms on the confusion matrix as a result of the representation of the classification process, among them True Positive (TP), True Negative (TN), False positive (FP), and False negative (FN) [5].

However, since this research is a multi-classification, the measurement needed is only an accuracy value to measure the amount of correctly classified data compared to the whole data [5]. For the formula to calculate the accuracy value can be seen in the following equation:

$$Accuracy(y, \hat{y}) = \frac{1}{n_{sample}} \sum_{i=0}^{n_{samples^{-1}}} 1(y, \hat{y}) \tag{2}$$

3. RESULTS AND DISCUSSION

This study used CNN and RNN as model classifications, TF-IDF N-gram as baseline testing and feature extraction, and GloVe as feature expansion. The classification theme is based on four scenarios. The first scenario uses a splitting ratio, for the best results will be used for the next scenario. The second scenario is to compare the baseline, and the best results will be used in the next scenario. The third scenario improves the best baseline outcomes by applying the GloVe feature expansion. The fourth scenario improves the best baseline outcomes using a hybrid classification model and applying the GloVe feature expansion.

3.1. Results

3.1.1. Scenario 1

In the first scenario, the testing used the Uni-gram TF-IDF baseline with the application of the splitting ratio to find the best results. The split ratio used in this study is 90:10, 80:20, and 70:30. The test results can be seen in Table 4.

Table 4. Result of First Scenario

Split Ratio	Accuracy (%)	
	CNN	RNN
90:10	92.32	93.20
80:20	**92.43**	**93.50**
70:30	92.27	93.34

Table 4 shows the best accuracy result is the splitting ratio of 80:20 with the accurate value on the CNN model of 92.43% and the RNN model of 93.50%. The best results from the first scenario continued to the second scenario to compare the best baseline.

3.1.2. Scenario 2

In this scenario, the baseline comparison uses unigram, bigram, trigram, unigram+bigram, and unigram +bigram+trigram. The results of the test can be seen in Table 5.

Table 5. Result of the Second Scenario

Baseline	Accuracy (%)	
	CNN	RNN
Unigram (Baseline)	92.43	93.50
Bigram	71.34 (-22.83)	73.63 (-21)
Trigram	36.04 (-61.)	36.38 (-61)
Baseline + Bigram	76.13 (-17.62)	93.42 (-0.08)
Baseline + Bigram + Trigram	**92.50 (+0.07)**	**93.54 (+0.04)**

The table above shows that the baseline with the best accuracy results is Baseline + Bigram + Trigram, with the accurate development of the CNN model of 92.50% and RNN of 93.54%. Baseline results will continue to be tested in the third scenario.

3.1.3. Scenario 3

In the third scenario, an expansion is made using the GloVe feature expansion. In this process, the baseline and classification model will be added features with the similarity corpus built, including Top 1, Top 5, Top 10, and Top 15. There are two types of corpus: corpus tweets and corpus Tweet + IndoNews. Both corpus will be applied to the CNN and RNN classification models. Table 6 shows the results of experiments on CNN models.

Table 6. Result of Third Scenario Using CNN

Feature	Accuracy (%)		
	Baseline	Tweet	Tweet+IndoNews
Top 1	92.43	92.55 (+0.12)	92.63 (+0.21)
Top 5		**92.71 (+0.30)**	**92.80 (+0.40)**
Top 10		92.57 (+0.15)	92. 48 (+0.05)

In Table 6, the results of the experiments showed that the accuracy value of the expansion feature on the CNN classification model improved in the Top 5 features for both corpus against the baseline accuracy value. The best accuracy result was obtained on the corpus Tweet+IndoNews top 5 with a precise score of 92.80%. In the top ten, both scores have fallen.

In Table 7, the experimental results showed that the accuracy values of the expansive feature in the RNN classification model increased in the Top 1 and Top 10 features for both corpus against the baseline accuracy values. In the top 1, there was an increase in the Tweet+IndoNews corpus, while in the top 10, there was a rise in the Tweet corpus but a decrease in the Tweet+IndoNews corpus accuracy. But if you compare the top 1 with the top 10, the top 10 gets a higher accuracy of 93.72%.

Table 7. Result of Third Scenario Using RNN

Feature	Accuracy (%)		
	Baseline	Tweet	Tweet+IndoNews
Top 1	93.50	93.68 (+0.19)	**93.71 (+0.22)**
Top 5		93.70 (+0.21)	93.67 (+0.18)
Top 10		**93.72 (+0.23)**	-
Top 15		93.49 (-0.01)	-

3.1.4. Scenario 4

The fourth scenario applies a hybrid classification model by combining CNN with RNN. At this stage, testing continues to use the baseline with the best results tested on the second scenario. The feature expansion is also applied in the third scenario using two bodies with Top 1, Top 5, Top 10, and Top 15 features. This hybrid classification model's experiment aimed to analyze how hybrid deep learning influences topic classification by applying expansion and extraction features, hoping to obtain more optimal accuracy results. Two experiments will be conducted on this hybrid method, the combination of CNN-RNN and RNN-CNN.

Table 8 above shows the accuracy results of the CNN-RNN hybrid classification model by applying expansion and extraction features. In the Top 5 features, there was a rise for both corpus with accuracy values on the corpus Tweet of 94.42% and in the corpus Tweet+News of 94.56%. We can see that the higher accuracy values are on the Tweet+IndoNews corpus. In the Top 10, there is a decrease in accuracy.

Table 8. Result of the Forth Scenario Using CNN+RNN

Feature	Accuracy (%)		
	Baseline	Tweet	Tweet+IndoNews
Top 1	92.43	94.39 (+2.12)	93.30 (+0.94)
Top 5		**94.42 (+2.15)**	**94.56 (+2.30)**
Top 10		94.40 (+2.13)	93.38 (+1.02)

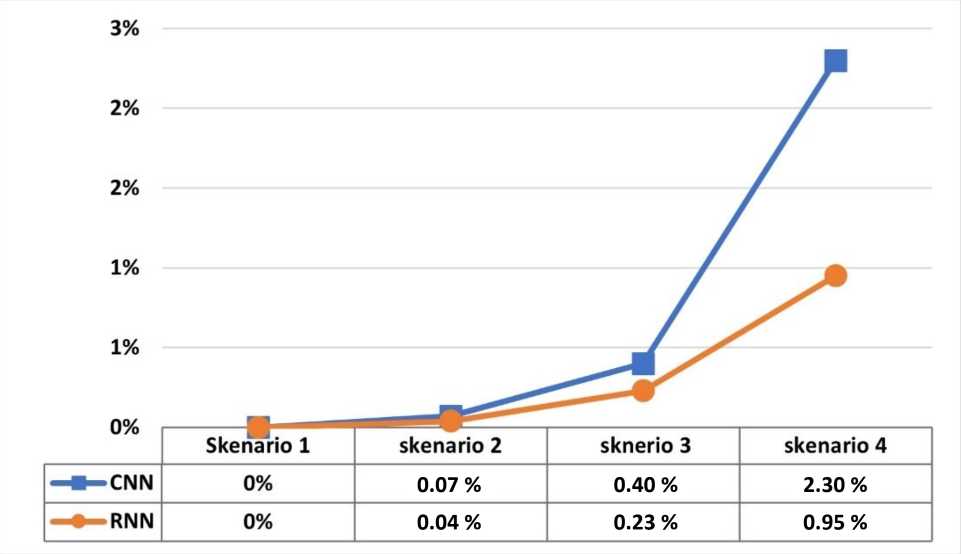
Table 9 shows the accuracy results of the RNN-CNN hybrid classification model by applying expansion and extraction features. It can be seen in the table above that the top 1 on the Tweet + IndoNews corpus experienced an increase in accuracy of 94.26%, while the top 5 also experienced a rise in the corpus Tweet by 94.39%. If we look at the accuracy values obtained in the experiment are superior to the Top 5 feature with the Tweet+IndoNews corpus. In the top 10 features experienced a decrease in accuracy.

Table 9. Result of the Forth Scenario Using RNN+CNN

Feature	Accuracy (%)		
	Baseline	Tweet	Tweet+IndoNews
Top 1	93.50	93.99 (+0.52)	**94.26 (+0.81)**
Top 5		**94.39 (+0.95)**	94.00 (+0.53)
Top 10		94.10 (+0.64)	-

3.2. Discussion

Based on the test results, all scenarios experienced performance improvement. Can be illustrated from the relative improvement chart in all the scenarios in Fig. 7.



	Skenario 1	skenario 2	sknerio 3	skenario 4
CNN	0%	0.07 %	0.40 %	2.30 %
RNN	0%	0.04 %	0.23 %	0.95 %

Fig. 7. The Highest Relative Increase in All Scenarios

The results of the chart above show that testing using the CNN, RNN, and hybrid methods with the expansion of features using TF-IDF and the extraction of relative GloVe features increased performance. In the first scenario, obtain the highest accuracy result with the application of N-gram unigram as a test baseline with a splitting ratio of 80:20. The second scenario was tested by testing 5 N-grams using a splitting ratio of 80:20 including Unigram, Bigram, Trigram, Bigram+Trigram, Unigram+Bigram + Trigram and obtaining the highest accuracy results on Unigrams + Bigrams, Trigrams. In this experiment, both classification models experienced an improvement in accuracy from the baseline, with an increase of 0.08% for CNN versus 0.04% for RNN. This is because N-gram word representation can improve accuracy by capturing relationships and contexts between words in the text in more detail. With N-grams, models can extract more complex and profound patterns from text, thereby enhancing natural language's overall classification and analysis ability and ultimately improving accuracy in text processing tasks.

In the third scenario, tests were conducted by combining the TF-IDF extraction feature with the Glove feature expansion. From the results of the experiments, the highest accuracy was achieved by the RNN model on the Top 10 feature with an accurate 93.72%. In comparison, CNN experienced a rise in the Top 5 features with an accuracy of 92.80%. If viewed in this scenario, the accuracy result has been improved due to the use of the GloVe feature expansion. GloVe provides rich word vector representations by leveraging co-occurrence statistics to capture semantic relationships between words in the text. By combining GloVe representation as an additional feature, the model can better understand semantic contexts, address low data problems, and improve overall performance on various text processing tasks.

In the fourth scenario, an experiment was conducted by combining the CNN model with RNN into a hybrid. In the testing, the writer also tried to test disproportionately, meaning that after trying to do the CNN-RNN test, the RNN-CNN test was also done. In this process, the CNN-RNN model obtained a higher accuracy than the baseline, with an accurate value of 94.56%. If we compare the single baseline results of the CNN model, the CNN-RNN hybrid model proved superior, with an increase of 2.30%. For the hybrid model, RNN-CNN also experienced a higher accuracy increase than the baseline, with 94.39%. The RNN-CNN model also increased by 0.95% compared to the baseline single RNN model.

In the fourth scenario, the accuracy of the results can be increased by combining the strengths of both methods to address the unique challenges of text data. CNN can extract features based on spatial patterns in text, thus effectively recognizing local patterns such as phrases or important words. Meanwhile, RNNs can handle temporal contexts and understand relationships between words in sequence. Combining both, hybrid deep learning can extract key features from the text in parallel and consider word sequence contexts in-depth, enhancing the ability to understand and represent text more comprehensively.

4. CONCLUSION

This study detected topics on social media Twitter using the Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) methods with the expansion of TF-IDF N-gram extraction features as baseline and GloVe expansion features. The researchers used data from 55411 tweets. Since a tweet is a short text message limited to only 280 characters, the Glove feature expansion is selected to reduce the non-conformity of words due to text data containing many variations of terms, such as abbreviations or slang languages. Ten labels are used to distinguish topics from each tweet on the dataset. On the results of this study, the CNN-RNN hybrid model combined with TF-IDF feature extraction and GloVe feature expansion achieved an accuracy of 94.56%, with an increase in accuracy of 2.30% compared to the CNN baseline. In contrast, the RNN-CNN Hybrid model, combined with TF-IDF features extraction and the GloVe features expansion, had a 94.39% accuracy increase with a 0.95% increase compared with the RNN baseline. From the results, it can be concluded that the hybrid deep learning method with the extension of feature extraction and feature expansion can increase the system's performance and achieve the best accuracy results. From the research results presented, the approach using the CNN-RNN hybrid with the expansion of the GloVe feature shows an interesting potential in text processing. However, the architectural complexity and compatibility of GloVe features can be a barrier. Future research should explore proper features, testing on more varied datasets, parameter optimization, and pre-processing techniques. Through overcoming these limitations, further research is expected to contribute significantly to understanding and analyzing texts.

REFERENCES

[1] A. S. Raamkumar, M. Erdt, H. Vijayakumar, E. Rasmussen, and Y.-L. Theng, "Understanding the Twitter usage of humanities and social sciences academic journals," *Proceedings of the Association for Information Science and Technology*, vol. 55, no. 1, pp. 430–439, 2018, https://doi.org/10.1002/pra2.2018.14505501047.

[2] A. Kumar and A. Jaiswal, "Systematic literature review of sentiment analysis on Twitter using soft computing techniques," *Concurr Comput*, vol. 32, p. e5107, 2019, https://doi.org/10.1002/cpe.5107.

[3] A. Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: a review," *Artif Intell Rev*, vol. 53, no. 6, pp. 4335–4385, 2020, https://doi.org/10.1007/s10462-019-09794-5.

[4] L. Yue, W. Chen, X. Li, W. Zuo, and M. Yin, "A survey of sentiment analysis in social media," *Knowl Inf Syst*, vol. 60, no. 2, pp. 617–663, 2019, https://doi.org/10.1007/s10115-018-1236-4.

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

[6] L. Sheng and L. Xu, "Topic Classification Based on Improved Word Embedding," in *2017 14th Web Information Systems and Applications Conference (WISA)*, pp. 117–121, 2017, https://doi.org/10.1109/WISA.2017.44.

[7] A. Meddeb and L. Ben Romdhane, "Using Topic Modeling and Word Embedding for Topic Extraction in Twitter," *Procedia Comput Sci*, vol. 207, pp. 790–799, 2022, https://doi.org/10.1016/j.procs.2022.09.134.

[8] D. T. Maulidia and E. Budi Setiawan, "Feature Expansion with Word2Vec for Topic Classification with Gradient Boosted Decision Tree on Twitter," in *2022 International Conference on Data Science and Its Applications (ICoDSA)*, pp. 87–92, 2022, https://doi.org/10.1109/ICoDSA55874.2022.9862907.

[9] A. Rafea and N. A. Gaballah, "Topic Detection Approaches in Identifying Topics and Events from Arabic Corpora," in *Procedia Computer Science*, pp. 270–277, 2018, https://doi.org/10.1016/j.procs.2018.10.492.

[10] K. Gligori´c, G. Epfl, A. Anderson, and R. West, "How Constraints Affect Content: The Case of Twitter's Switch from 140 to 280 Characters," in *Proceedings of the International AAAI Conference on Web and Social Media,* vol. 12, no. 1, 2018, https://doi.org/10.1609/icwsm.v12i1.15079.

[11] L. Parameswaran, B. K.R, and R. K, "A Text Classification Model Using Convolution Neural Network and Recurrent Neural Network," vol. 119, no. 7, 2018, http://www.acadpubl.eu/hub/.

[12] G. Carducci, G. Rizzo, D. Monti, E. Palumbo, and M. Morisio, "TwitPersonality: Computing Personality Traits From Tweets Using Word Embeddings and Supervised Learning," *Information (Switzerland)*, vol. 9, no. 5, 2018, https://doi.org/10.3390/info9050127.

[13] E. M. Dharma, F. Lumban Gaol, H. Leslie, H. S. Warnars, and B. Soewito, "The Accuracy Comparison Among Word2vec, Glove, And Fasttext Towards Convolution Neural Network (CNN) Text Classification," *J Theor Appl Inf Technol*, vol. 31, no. 2, 2022, http://www.jatit.org/volumes/Vol100No2/5Vol100No2.pdf.

[14] J. Claussen and C. Peukert, "Obtaining Data from the Internet: A Guide to Data Crawling in Management Research," *SSRN Electronic Journal*, 2019, https://doi.org/10.2139/ssrn.3403799.

[15] K. N. Alam *et al.*, "Deep Learning-Based Sentiment Analysis of COVID-19 Vaccination Responses from Twitter Data," *Comput Math Methods Med*, vol. 2021, p. 4321131, 2021, https://doi.org/10.1155/2021/4321131.

[16] F. Gargiulo, S. Silvestri, M. Ciampi, and G. De Pietro, "Deep neural network for hierarchical extreme multi-label text classification," *Appl Soft Comput*, vol. 79, pp. 125–138, 2019, https://doi.org/10.1016/j.asoc.2019.03.041.

[17] 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.

[18] M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi, and B.-W. On, “Fake News Stance Detection Using Deep Learning Architecture (CNN-LSTM),” *IEEE Access*, vol. 8, pp. 156695–156706, 2020, https://doi.org/10.1109/ACCESS.2020.3019735.

[19] M. Rosid, A. Fitrani, I. Astutik, N. Mulloh, and H. Gozali, “Improving Text Preprocessing For Student Complaint Document Classification Using Sastrawi,” *IOP Conf Ser Mater Sci Eng*, vol. 874, p. 12017, 2020, https://doi.org/10.1088/1757-899X/874/1/012017.

[20] M. Anandarajan, C. Hill, and T. Nolan, “Text Preprocessing,” in *Practical Text Analytics*, pp. 45–59, 2019, https://doi.org/10.1007/978-3-319-95663-3_4 .

[21] J. Yao, “Automated Sentiment Analysis of Text Data with NLTK,” *J Phys Conf Ser*, vol. 1187, no. 5, p. 52020, 2019, https://doi.org/10.1088/1742-6596/1187/5/052020.

[22] E. Setiawan, D. Widyantoro, and K. Surendro, “Measuring information credibility in social media using combination of user profile and message content dimensions,” *International Journal of Electrical and Computer Engineering*, vol. 10, pp. 3537–3549, 2020, https://doi.org/10.11591/ijece.v10i4.pp3537-3549.

[23] D. J. Ladani and N. P. Desai, “Stopword Identification and Removal Techniques on TC and IR applications: A Survey,” in *6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 466–472, 2020, https://doi.org/10.1109/ICACCS48705.2020.9074166.

[24] D. Merlini and M. Rossini, “Text categorization with WEKA: A survey,” *Machine Learning with Applications*, vol. 4, p. 100033, 2021, https://doi.org/10.1016/j.mlwa.2021.100033.

[25] A. Kadhim, “An Evaluation of Preprocessing Techniques for Text Classification,” *International Journal of Computer Science and Information Security,* vol. 16, no. 6, pp. 22–32, 2018, https://www.slideshare.net/IJCSISResearchPublic/an-evaluation-of-preprocessing-techniques-for-text-classification.

[26] A. I. Kadhim, “Term Weighting for Feature Extraction on Twitter: A Comparison Between BM25 and TF-IDF,” in *International Conference on Advanced Science and Engineering (ICOASE)*, pp. 124–128, 2019, https://doi.org/10.1109/ICOASE.2019.8723825.

[27] P. M. Prihatini, I. K. Suryawan, and I. N. Mandia, “Feature extraction for document text using Latent Dirichlet Allocation,” *J Phys Conf Ser*, vol. 953, no. 1, p. 12047, 2018, https://doi.org/10.1088/1742-6596/953/1/012047.

[28] N. Nasser, L. Karim, A. El Ouadrhiri, A. Ali, and N. Khan, “n-Gram based language processing using Twitter dataset to identify COVID-19 patients,” *Sustain Cities Soc*, vol. 72, p. 103048, 2021, https://doi.org/10.1016/j.scs.2021.103048.

[29] S. W. Kim and J. M. Gil, “Research paper classification systems based on TF-IDF and LDA schemes,” *Human-centric Computing and Information Sciences*, vol. 9, no. 1, 2019, https://doi.org/10.1186/s13673-019-0192-7.

[30] A. I. Kadhim, “Survey on supervised machine learning techniques for automatic text classification,” *Artif Intell Rev*, vol. 52, no. 1, pp. 273–292, 2019, https://doi.org/10.1007/s10462-018-09677-1.

[31] R. Dzisevič and D. Šešok, “Text Classification using Different Feature Extraction Approaches,” in *Open Conference of Electrical, Electronic and Information Sciences (eStream)*, pp. 1–4, 2019, https://doi.org/10.1109/eStream.2019.8732167.

[32] M. Birjali, M. Kasri, and A. B-Hssane, “A Comprehensive Survey on Sentiment Analysis: Approaches, Challenges and Trends,” *Knowl Based Syst*, vol. 226, p. 107134, 2021, https://doi.org/10.1016/j.knosys.2021.107134.

[33] C. I. Eke, A. Norman, L. Shuib, F. B. Fatokun, and I. Omame, “The Significance of Global Vectors Representation in Sarcasm Analysis,” in *International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS)*, pp. 1–7, 2020, https://doi.org/10.1109/ICMCECS47690.2020.246997.

[34] K. Kowsari *et al.*, “Text Classification Algorithms: A Survey,” *Information (Switzerland)*, vol. 10, 2019, https://doi.org/10.3390/info10040150.

[35] D. Alsaleh and S. Larabi-Marie-Sainte, “Arabic Text Classification Using Convolutional Neural Network and Genetic Algorithms,” *IEEE Access*, vol. 9, pp. 91670–91685, 2021, https://doi.org/10.1109/ACCESS.2021.3091376.

[36] S. V Georgakopoulos, S. K. Tasoulis, A. G. Vrahatis, and V. P. Plagianakos, “Convolutional Neural Networks for Toxic Comment Classification,” *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*, pp. 1-6, 2018, https://doi.org/10.1145/3200947.3208069.

[37] N. I. Widiastuti, “Convolution Neural Network for Text Mining and Natural Language Processing,” *IOP Conf Ser Mater Sci Eng*, vol. 662, no. 5, p. 52010, 2019, https://doi.org/10.1088/1757-899X/662/5/052010.

[38] E. Tutubalina, Z. Miftakhutdinov, S. Nikolenko, and V. Malykh, “Medical Concept Normalization in Social Media Posts with Recurrent Neural Networks,” *J Biomed Inform*, vol. 84, 2018, https://doi.org/10.1016/j.jbi.2018.06.006.

[39] E. C. Nisa and Y. D. Kuan, “Comparative assessment to predict and forecast water-cooled chiller power consumption using machine learning and deep learning algorithms,” *Sustainability (Switzerland)*, vol. 13, no. 2, pp. 1–18, 2021, https://doi.org/10.3390/su13020744.

[40] M. Azizjon, A. Jumabek, and W. Kim, “1D CNN based network intrusion detection with normalization on imbalanced data,” in *International Conference on Artificial Intelligence in Information and Communication*, pp. 218–224, 2020, https://doi.org/10.1109/ICAIIC48513.2020.9064976.

[41] H. Mohaouchane, A. Mourhir, and N. S. Nikolov, “Detecting Offensive Language on Arabic Social Media Using Deep Learning,” in *Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pp. 466–471, 2019, https://doi.org/10.1109/SNAMS.2019.8931839.

[42] G. K. Soon, C. K. On, N. M. Rusli, T. S. Fun, R. Alfred, and T. T. Guan, “Comparison Of Simple Feedforward Neural Network, Recurrent Neural Network And Ensemble Neural Networks In Phishing Detection,” *J Phys Conf Ser*, vol. 1502, no. 1, p. 012033, 2020, https://doi.org/10.1088/1742-6596/1502/1/012033.

[43] E. Singh, N. Kuzhagaliyeva, and S. M. Sarathy, “Chapter 9 - Using deep learning to diagnose preignition in turbocharged spark-ignited engines,” in *Artificial Intelligence and Data Driven Optimization of Internal Combustion Engines*, pp. 213–237, 2022, https://doi.org/10.1016/B978-0-323-88457-0.00005-9.

[44] G. Tanaka *et al.*, “Recent Advances In Physical Reservoir Computing: A Review,” *Neural Networks*, vol. 115, pp. 100–123, 2019, https://doi.org/10.1016/j.neunet.2019.03.005.

[45] C. Du, L. Huang, C. Du, and L. Huang, “Text Classification Research with Attention-based Recurrent Neural Networks,” *International Journal of Computers Communications & Control,* vo. 13, no.1, pp. 50-61, 2018, https://doi.org/10.15837/ijccc.2018.1.3142.

[46] H. Jelodar, Y. Wang, R. Orji, and S. Huang, “Deep Sentiment Classification and Topic Discovery on Novel Coronavirus or COVID-19 Online Discussions: NLP Using LSTM Recurrent Neural Network Approach,” *IEEE J Biomed Health Inform*, vol. 24, no. 10, pp. 2733–2742, 2020, https://doi.org/10.1109/JBHI.2020.3001216.

[47] X. Liu, Y. Su and B. Xu, "The Application of Graph Neural Network in Natural Language Processing and Computer Vision," *3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, pp. 708-714, 2021, https://doi.org/10.1109/MLBDBI54094.2021.00140.

[48] S. V. Bhoir, “An Efficient Fake News Detector,” in *International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1-9, 2020, https://doi.org/10.1109/ICCCI48352.2020.9104177.

[49] R. Hamad, L. Yang, W. L. Woo, and B. Wei, “Joint Learning of Temporal Models to Handle Imbalanced Data for Human Activity Recognition,” *Applied Sciences*, vol. 10, p. 5293, 2020, https://doi.org/10.3390/app10155293.

BIOGRAPHY OF AUTHORS

Windy Ramadhanti is an undergraduate student in the School of Computing at Telkom University. Her research is interested in Data Science. Email: windyramadhanti@student.telkomuniversity.ac.id.

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