Accredited Ranking SINTA 2
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 6 No. 6 (2022) 1037-1046 ISSN Media Electronic: 2580-0760

Using Social Media Data to Monitor Natural Disaster: A Multi Dimension Convolutional Neural Network Approach with Word Embedding

Mohammad Reza Faisal[1], Irwan Budiman[2], Friska Abadi[3], Muhammad Haekal[4], Mera Kartika Delimayanti[5], Dodon Turianto Nugrahadi[6]
[1,2,3,4,6]Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lambung Mangkurat
[5]Program Studi D4 Teknik Informatika, Jurusan Teknik Informatika dan Komputer, Politeknik Negeri Jakarta
[1]reza.faisal@ulm.ac.id*, [2]irwan.budiman@ulm.ac.id, [3]friska.abadi @ulm.ac.id, [4]muhammadhaekal77@gmail.com, [5]mera.kartika@tik.pnj.ac.id, [6]dodonturianto@ulm.ac.id

Abstract

Social media has a significant role in natural disaster management, namely as an early warning and monitoring when natural disasters occur. Artificial intelligence can maximize the use of natural disaster social media messages for natural disaster management. The artificial intelligence system will classify social media message texts into three categories: eyewitness, non-eyewitness and don't-know. Messages with the eyewitness category are essential because they can provide the time and location of natural disasters. A common problem in text classification research is that feature extraction techniques ignore word meanings, omit word order information and produce high-dimensional data. In this study, a feature extraction technique can maintain word order information and meaning by using three-word embedding techniques, namely word2vec, fastText, and Glove. The result is data with 1D, 2D, and 3D dimensions. This study also proposes a data formation technique with new features by combining data from all word embedding techniques. The classification model is made using three Convolutional Neural Network (CNN) techniques, namely 1D CNN, 2D CNN and 3D CNN. The best accuracy results in this study were in the case of earthquakes 78.33%, forest fires 81.97%, and floods 78.33%. The calculation of the average accuracy shows that the 2D and 3D v1 data formation techniques work better than other techniques. Other results show that the proposed technique produces better average accuracy.

Keywords: *natural disaster, word embedding, convolutional neural network, twitter, social media*

1. Introduction

The existence of social media currently plays an essential role in assisting every activity in the disaster management cycle. During the pre-disaster stage, social media can serve as an early warning before a disaster occurs [1]. At the scene when the disaster occurred, social media users who were eyewitnesses shared information about the situation at that time. This information can be used by related parties, such as volunteers or the government to deal with the impact of disasters. Meanwhile, in the post-disaster stage, social media users share messages containing information on assistance that has been carried out or reports on locations that have not received aid [2].

Using social media messages related to natural disasters for natural disaster management can be maximized with the help of artificial intelligence. Artificial intelligence can help find messages related to natural disasters more quickly [3]. The artificial intelligence system will classify social media messages into three categories: *eyewitness*, *non-eyewitness*, and *don't-know* [4]. Messages with the *eyewitness* category are natural disaster messages posted by eyewitnesses at the scene when the disaster occurred. *Non-eyewitness* category messages are messages about natural disasters posted by users who are not eyewitnesses. Meanwhile, messages in the *don't-know* category are messages with the keyword natural disasters, but their meaning has nothing to do with natural disasters.

Classification of social media messages in natural disasters has technical similarities with other text classifications, consisting of feature extraction and classification processes. The feature extraction technique used in text classification is divided into three types: vector space representation, lexicon-based, and word embedding vector-based. The following process is classification using structured data, which is the output of the feature extraction process. Two types of

Accepted: 22-09-2022 | Received in revised: 05-12-2022 | Published: 29-12-2022

classification techniques are used in text classification: shallow learning and deep learning.

Space representation [5],[6] and lexicon-based [7] feature extraction techniques have been used to classify natural disaster messages. However, both of these techniques have the disadvantage of eliminating word order information. In addition, it produces high-dimensional data and ignores the meaning of words for the space representation technique. Lexicon-based and word embedding vector-based techniques keep to word meanings. However, the word order information is lost when the sentence vector is formed by adding up each word vector [8]. This result will affect the meaning of the sentence because the meaning of the sentence can be different if the order of the words in it changes.

The word embedding-based feature extraction technique is formed by concatenating word vectors into 1-dimensional (1D) data [9] and arranging the word vectors into a two-dimensional matrix (2D data) [10], [11]. The three 2D data formed by each word embedding technique, such as Wod2vec, Glove and fastText can be combined into three layers to create three-dimensional (3D) data [12].

After the data is extracted into structured data, the data can be processed by a classification algorithm. A classification algorithm that can process 1D, 2D, and 3D data is a Convolutional Neural Network (CNN). For text classification with 1D CNN with feature extraction technique based on word2vec [9]. The application of the 2D CNN technique to the classification of forest fire natural disaster messages produces a good accuracy of 81.97% [11]. This study used three-word embedding techniques to create 2D data, namely word2vec, fastText and Glove. The application of text data classification with 3D CNN is made by combining 2D data based on word embedding which is arranged in three layers. Each layer is 2D data that are generated by using word2vec, Glove, and fastText [12].

As with other classification algorithms, the input data processed by CNN must have the same feature size. Text data generally has a different number of words, so it needs to be processed to equalize the number of words before being processed at the word embedding vector-based feature extraction stage. One way to equalize is by using the maximum number of words in the data [10].

From the explanation above, word embedding can be used to build multi-dimensional input data and then processed by the CNN technique to construct a classification model. This study conducted a comprehensive analysis of the implementation of 1D, 2D, and 3D CNN for the classification of natural disaster messages by feature extraction using three-word embedding techniques, namely word2vec, fastText, and Glove. In addition, the three extraction results of the three-word embedding techniques were also combined for 1D and 2D data as a new feature formation technique. And the formation of two types of 3D data. In this study, to equate the features of the input data, word padding was used based on statistical calculations, namely the mean, median, and mode.

The results of this study can provide classification accuracy of 1D, 2D, and 3D data generation techniques and their combinations. This study also investigated word padding techniques' effect on improving classification performance so that the final results of this study can contribute to determining the best technique for classifying natural disaster messages.

2. Research Methods

The research procedures can be seen in Figure 1.

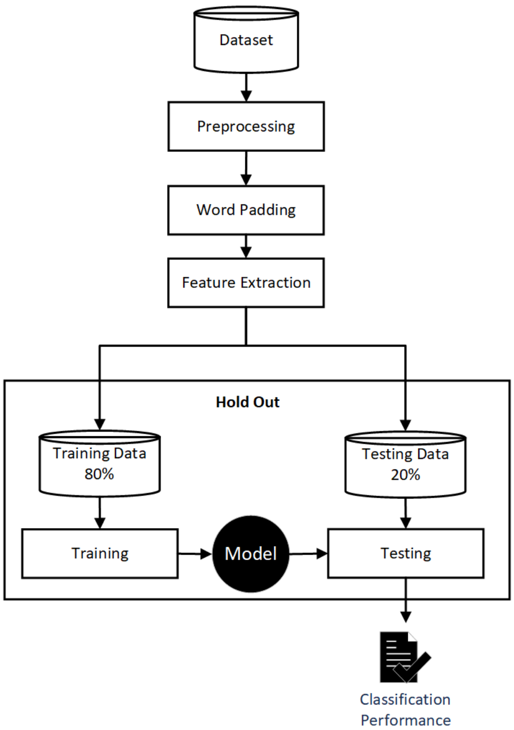

Figure 1. Research procedure.

2.1. Dataset

This study uses three datasets with details that can be seen in Table 1. Each dataset consists of 3000 Twitter messages. These three datasets can be downloaded at the following link:

https://github.com/rezafaisal/NaturalDisasterOnTwitter.

Table 1. Dataset

Dataset	Class Label		
	eyewitness	non-eyewitness	don't-know
Earthquakes [7]	1000	1000	1000
Forest fires [11]	1000	1000	1000
Floods [5], [6]	1000	1000	1000

DOI: https://doi.org/10.29207/resti.v6i6.4525

Each dataset is divided into three class labels or categories: *eyewitness*, *non-eyewitness*, and *don't-know*. Each category consists of 1000 messages. The three datasets are balanced because each class label has the same number.

2.2. Preprocessing and Word Padding

Then the three datasets are pre-processed with steps commonly performed in text classification, namely removing double spaces, punctuation, numbers and non-alphanumeric characters. [6], [7], [13].

The next step is to count the number of words in each message. Then the mean, mode and median values are calculated based on the data on the number of words. The implementation of word padding is done to equalize the number of words in each message in the text data. The output of this stage produces nine datasets. The nine datasets will be processed one by one in the next stage.

2.3. Feature Extraction

The next stage is feature extraction. There are three groups of techniques based on the dimensions of the output data, namely, one dimension (1D), two dimensions (2D), and three dimensions (3D). Each of these techniques will use three-word embedding techniques that are popularly used in classification studies, namely word2vec [14], fastText [15], and Glove [16], [17]. By using the three-word embedding models, data structures with different dimensions are created. The formation of 1D data is explained as follows.

v_i	=	V1	V2	V2	…	V98	V99	V100

If v_i is a word vector, and n is the number of words in a sentence, then 1D data is a sentence vector v_s which is formed by combining all word vectors with formula 1. The value of n corresponds to the number of words searched for using statistical-based word padding techniques, namely mean, mode or median.

$$v_s = \bigcup_{i=1}^{n} v_i \tag{1}$$

The method of forming 1D data which is a combination of the three word embedding techniques can be seen in Formula 2.

$$v_{all} = v_{s\,word2vec} \cup v_{s\,fastText} \cup v_{s\,Glove} \tag{2}$$

Formation of combined 1D data by combining 1D data using the word padding technique with the same number of words.

An explanation of how to create 2D data with each word embedding technique is as follows. If v_1, v_2, to v_n are the word vectors produced by a word embedding technique, m is the value of the word vector, and n is the number of words in a sentence. Then 2D data is formed by creating a two-dimensional matrix $m \times n$ as shown in Figure 2, where m is 100.

v_1	=	V1	V2	V2	…	V98	V99	V100
v_2	=	V1	V2	V2	…	V98	V99	V100
…	…	…	…	…	…	…	…	…
v_n	=	V1	V2	V2	…	V98	V99	V100

Figure 2. Data 2D.

In this study, 2D data was also formed from the three-word embedding techniques. The combined 2D data can be seen in Figure 3. This 2D data is a two-dimensional matrix $m \times N$, where m is 100 and N is $3 \times n$.

word2vec	V1	V2	V2	…	V98	V99	V100
	V1	V2	V2	…	V98	V99	V100
	…	…	…	…	…	…	…
	…	…	…	…	…	…	…
	V1	V2	V2	…	V98	V99	V100
fastText	V1	V2	V2	…	V98	V99	V100
	V1	V2	V2	…	V98	V99	V100
	…	…	…	…	…	…	…
	…	…	…	…	…	…	…
	V1	V2	V2	…	V98	V99	V100
Glove	V1	V2	V2	…	V98	V99	V100
	V1	V2	V2	…	V98	V99	V100
	…	…	…	…	…	…	…
	…	…	…	…	…	…	…
	V1	V2	V2	…	V98	V99	V100

Figure 3. 2D data combined with three-word embedding techniques.

The creation of 3D data is done by combining 2D data from three different word embedding techniques but with the same word length. There are two ways of generating 3D data. The way of forming the first 3D data can be seen in Figure 4. 3D data type 1 is generated in 3 layers. The first layer is 2D data from the word2vec technique, followed by 2D data from the fastText and Glove techniques.

If 3D data type 1 is represented as a matrix, the dimensions are $m \times n \times z$. Where m is 100, n is the number of words in a sentence, and z is 3. The method of forming 3D data type 1 follows the technique of forming 3D image data consisting of 3 RGB color channels.

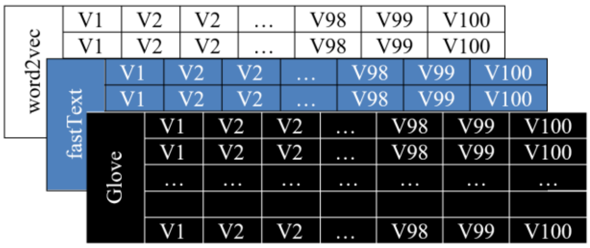

Figure 4. 3D Data type 1.

The way to form type 2 3D data is to follow the formation of 3D data in the form of a video. A video consists of a series of frames or pictures. For this study, a frame is formed by three vectors of a word from three-

DOI: https://doi.org/10.29207/resti.v6i6.4525

word embedding techniques. Then the next layer represents the second word, and so on. So that this data will have n layers, the number of words in the sentence. If 3D data type 2 is represented as a matrix, the dimensions are $m \times z \times n$. Where m is 100, z is the number of word embedding techniques, namely 3, and n is the number of words in the sentence.

2.4. Classification

In the next stage, each data created in the previous step is divided using the hold-out technique into training and testing data with a ratio of 80:20 [18]. The training data will be used to create a classification model with a Convolutional Neural Network (CNN) [19], [20]. The general architecture used in this study can be seen in Figure 5.

This architecture consists of 7 layers. The first layer is the convolutional layer which processes structured data with N dimensions (N-D data), namely 1D, 2D and 3D. In Figure 5, this layer is named Conv-N-D. This layer functions as a filter; this filter's value becomes an update parameter in the learning process. The next layer is pooling which functions to reduce input by reducing the number of parameters. The pooling method that is commonly used is max-pooling, which is to take the largest value in that section. This layer is named MaxPooling-N-D. The layers in the box with the dotted line will be adjusted according to the data dimensions.

The next layer is the dropout which functions to discard neurons randomly. The technique used is regularising artificial neural networks where some neurons are randomly selected and not used during learning. Unused neurons are temporarily suspended from the network, and new weights are also not applied to neurons during backpropagation. This layer also serves to prevent overfitting and speed up the learning process. Then the output from this layer is combined in the concatenate and flatten layers. The flatten layer aims to change the feature map produced in the previous process into a multi-dimensional array reshaped into a vector. Next, the process is continued by the dropout layer. The last layer is dense which functions as a fully connected layer where each neuron receives input from all neurons from the previous layer.

CNN implementation in this study uses the Python programming language, the TensorFlow library, and the Keras framework. Details of the functions and configurations used can be seen in Table 2.

In Table 2, there are four columns. The first column is the layer names of the CNN architecture used in this research. Then the second column is the names of functions and configurations for the 1D CNN architecture. The third and fourth columns are the names of the functions and configurations for the 2D CNN and 3D CNN architectures.

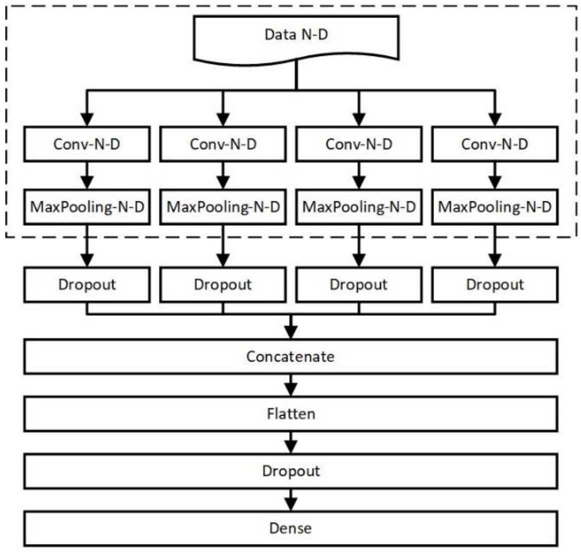

Figure 5. N-D CNN Architecture.

Table 2. Functions and configurations in CNN implementation.

Layer	1D CNN	2D CNN	3D CNN
Convolutional	**Conv1D** kernel initializer: orthogonal; bias initializer: glorot_uni-form; actication: tanh;	**Conv2D** kernel initializer: orthogonal; bias initializer: glorot_uni-form; actication: tanh;	**Conv3D** kernel initializer: orthogonal; bias initializer: glorot_uni-form; actication: tanh;
Pooling	MaxPool1D pool size: maxword – 1; padding: valid;	MaxPool2D pool size: maxword – 1; padding: valid;	MaxPool3D pool size: maxword – 1; padding: valid;
Dropout		p: 0.25	
Concatenante		-	
Flatten		-	
Dropout		p: 0.35	
Dense		unit: 3 activation: softmax	

The classification model with the highest accuracy will be used to predict the class label from the testing data. The accuracy of the classification model is calculated based on commonly used techniques, namely the confusion matrix [21], [22]. Then the accuracy of the classification model of each dataset will be stored. The results of all classification accuracy or performance will be compared to answer the questions or objectives of this study.

3. Results and Discussions

3.1. Results

The three datasets used in this research are messages about floods, forest fires, and earthquakes. The three datasets are input to the word padding process using statistical techniques, namely the mean, mode and median. The output of this word padding process can be seen in Table 3.

The first column in Table 3 is the input dataset in the form of natural disaster messages which have previously been described in Table 1. The second column is the word padding technique to equalize the number of words in the message based on statistical techniques. While the third column is the number of words obtained using the word padding technique. The fourth column is the output dataset code.

For example, the earthquake dataset is processed using the mean-based word padding technique. As a result, the earthquake dataset will have the same number of words, namely 19 words. Furthermore, this dataset is given the code E1. Moreover, the earthquake dataset is equated with the number of words with word padding based on the median and mode, producing an output dataset with codes E2 and E3.

Table 3. Datasets that are generated by the word padding process.

Input Dataset	Word Padding	Number of Words	Output Dataset
Earthquakes	mean	19	E1
	median	17	E2
	modus	16	E3
Forest fires	mean	8	FF1
	median	7	FF2
	modus	6	FF3
Floods	mean	8	F1
	median	7	F2
	modus	6	F3

The feature extraction stages are used to process the nine data in Table 3. The 1D data feature extraction process with three-word embedding techniques and their combination produces data, as shown in Table 4.

Table 4. 1D Datasets.

Input Dataset	Word Embedding Technique	1D Features	Output Dataset	Accuracy (%)
Earthquakes				
E1	Word2vec	1900	1D_E1W	71.00
	fastText	1900	1D_E1F	71.00
	Glove	1900	1D_E1G	71.67
	Word2vec, fastText, Glove	5700	1D_E1_All	73.33
E2	Word2vec	1700	1D_E2W	69.00
	fastText	1700	1D_E2F	**74.33**
	Glove	1700	1D_E2G	72.67
	Word2vec, fastText, Glove	5100	1D_E2_All	72.67
E3	Word2vec	1600	1D_E3W	67.33
	fastText	1600	1D_E3F	71.33
	Glove	1600	1D_E3G	69.67
	Word2vec, fastText, Glove	4800	1D_E3_All	73.67
Forest Fires				
FF1	Word2vec	800	1D_FF1W	**81.14**
	fastText	800	1D_FF1F	76.63
	Glove	800	1D_FF1G	80.32
	Word2vec, fastText, Glove	2400	1D_FF1_All	80.73
FF2	Word2vec	700	1D_FF2W	75.81
	fastText	700	1D_FF2F	79.50
	Glove	700	1D_FF2G	78.68
	Word2vec, fastText, Glove	2100	1D_FF2_All	**81.14**
FF3	Word2vec	600	1D_FF3W	75.81
	fastText	600	1D_FF3F	75.81
	Glove	600	1D_FF3G	77.86
	Word2vec, fastText, Glove	1800	1D_FF3_All	77.45
Floods				
F1	Word2vec	800	1D_F1W	70.00
	fastText	800	1D_F1F	69.67
	Glove	800	1D_F1G	68.33
	Word2vec, fastText, Glove	2400	1D_F1_All	**74.00**
F2	Word2vec	700	1D_F2W	68.67
	fastText	700	1D_F2F	67.00
	Glove	700	1D_F2G	64.33
	Word2vec, fastText, Glove	2100	1D_F2_All	66.67
F3	Word2vec	600	1D_F3W	63.67
	fastText	600	1D_F3F	66.67
	Glove	600	1D_F3G	68.33
	Word2vec, fastText, Glove	1800	1D_F3_All	67.00

The first column is the dataset that has been generated in Table 3. The second column is the word embedding feature extraction technique used to process the input dataset. The third column is the number of features produced after the input dataset is processed by the word embedding technique. The fourth column is the output dataset code from the feature extraction process.

For example, the E1 dataset is earthquake text data which has been equated with 19 words. The feature extracted E1 dataset using the word embedding technique Word2vec produces structured data with 1900 features. The output dataset of this feature extraction process is coded 1D_E1W. Code 1D_E1W means that 1D is the data dimension, E1 is the data source, and W is the Word2vec technique. The feature extraction output using fastText and Glove produces 1D data, namely 1D_E1F and 1D_E1G. Meanwhile, the combined 1D data results from the three datasets are stored as 1D_E1_All. Merging is done by concatenating data 1D_E1W, 1D_E1F, and 1D_E1G.

The results of the feature extraction process for 2D data can be seen in Table 5. In the third column, you can see the dimensions of the feature extracted data, for example, for the 2D_E1W dataset, the dimension is a 100×19 matrix where 100 is a vector of words, and 19 is the number of words in a message. In the fourth

DOI: https://doi.org/10.29207/resti.v6i6.4525

column is the generated dataset code. The 2D_E1F code means that 2D is the data dimension, E1 is the data source, and F is the fastText technique.

Table 5. 2D Datasets.

Input Dataset	Word Embedding Technique	(2D Features	Output Dataset	Accuracy (%)
Earthquakes				
E1	Word2vec	100 x 19	2D_E1W	70.67
	fastText	100 x 19	2D_E1F	71.00
	Glove	100 x 19	2D_E1G	**78.33**
	Word2vec, fastText, Glove	100 x 57	2D_E1_All	**78.33**
E2	Word2vec	100 x 17	2D_E2W	70.00
	fastText	100 x 17	2D_E2F	69.67
	Glove	100 x 17	2D_E2G	71.00
	Word2vec, fastText, Glove	100 x 51	2D_E2_All	74.33
E3	Word2vec	100 x 16	2D_E3W	72.00
	fastText	100 x 16	2D_E3F	71.33
	Glove	100 x 16	2D_E3G	71.67
	Word2vec, fastText, Glove	100 x 48	2D_E3_All	73.67
Forest fires				
FF1	Word2vec	100 x 8	2D_FF1W	76.23
	fastText	100 x 8	2D_FF1F	77.87
	Glove	100 x 8	2D_FF1G	78.28
	Word2vec, fastText, Glove	100 x 24	2D_FF1_All	81.15
FF2	Word2vec	100 x 7	2D_FF2W	81.15
	fastText	100 x 7	2D_FF2F	76.23
	Glove	100 x 7	2D_FF2G	76.23
	Word2vec, fastText, Glove	2100	2D_FF2_All	77.46
FF3	Word2vec	100 x 6	2D_FF3W	76.64
	fastText	100 x 6	2D_FF3F	79.51
	Glove	100 x 6	2D_FF3G	**81.97**
	Word2vec, fastText, Glove	100 x 18	2D_FF3_All	75.81
Floods				
F1	Word2vec	100 x 8	2D_F1W	69.33
	fastText	100 x 8	2D_F1F	69.67
	Glove	100 x 8	2D_F1G	68.33
	Word2vec, fastText, Glove	100 x 24	2D_F1_All	**78.33**
F2	Word2vec	100 x 7	2D_F2W	70.33
	fastText	100 x 7	2D_F2F	65.33
	Glove	100 x 7	2D_F2G	66.00
	Word2vec, fastText, Glove	100 x 21	2D_F2_All	71.00
F3	Word2vec	100 x 6	2D_F3W	66.00
	fastText	100 x 6	2D_F3F	68.33
	Glove	100 x 6	2D_F3G	61.33
	Word2vec, fastText, Glove	100 x 18	2D_F3_All	65.33

Output datasets with codes ending in _All are combined by concatenation datasets with the endings W, F, and G. For example, the dataset 2D_E1_All is the result of concatenation datasets with codes 2D_E1W, 2D_E1F, and 2D_E1G.

The results of the feature extraction process for 3D data type 1 can be seen in Table 6. 3D data type 1 is formed in the manner described in Figure 4. In the third column, you can see the dimensions of the feature-extracted data used as shown in the second column. For example, a dataset with code 3D_E1_v1 is a dataset of E1 text data extracted with three-word embedding techniques, namely Word2vec, fastText, and Glove. So that we get data with 100×19 based on word vectors which total 100 values and as many as 19 words for each word embedding technique. Because there are three techniques, we finally get 3D data with dimensions $100 \times 19 \times 3$.

Table 6. 3D type 1 datasets.

Input Dataset	Word Embedding Technique	3D Features	Output Dataset	Accuracy (%)
Earthquakes				
E1	Word2vec, fastText, Glove	100 x 19 x 3	3D_E1_v1	**77.00**
E2	Word2vec, fastText, Glove	100 x 17 x 3	3D_E2_v1	68.00
E3	Word2vec, fastText, Glove	100 x 16 x 3	3D_E3_v1	75.00
Forest fires				
FF1	Word2vec, fastText, Glove	100 x 8 x 3	3D_FF1_v1	76.22
FF2	Word2vec, fastText, Glove	100 x 7 x 3	3D_FF2_v1	**81.55**
FF3	Word2vec, fastText, Glove	100 x 6 x 3	3D_FF3_v1	69.67
Floods				
F1	Word2vec, fastText, Glove	100 x 8 x 3	3D_F1_v1	**72.33**
F2	Word2vec, fastText, Glove	100 x 7 x 3	3D_F2_v1	66.67
F3	Word2vec, fastText, Glove	100 x 6 x 3	2D_F3_v1	71.00

The results of the feature extraction process for 3D data type 2 can be seen in Table 7. 3D data type 2 is created by following the formation of 3D data in the form of a video consisting of a series of frames or images. So the dimensions of 3D data type 2 are $100 \times 3 \times 19$.

The number of datasets generated from various feature extraction techniques is 90 datasets. Furthermore, one by one of the 90 datasets are processed at the classification stage to produce 90 performance classification models. The performance results of the classification model of each feature extraction technique can be seen in the accuracy column in those tables.

Table 7. 3D type 2 datasets.

Input Dataset	Word Embedding Technique	3D Features	Output Dataset	Accuracy (%)
Earthquakes				
E1	Word2vec, fastText, Glove	100 x 3 x 19	3D_E1_v2	72.33
E2	Word2vec, fastText, Glove	100 x 3 x 17	3D_E2_v2	**74.33**
E3	Word2vec, fastText, Glove	100 x 3 x 16	3D_E3_v2	67.00
Forest fires				
FF1	Word2vec, fastText, Glove	100 x 3 x 8	3D_FF1_v2	75.00
FF2	Word2vec, fastText, Glove	100 x 3 x 7	3D_FF2_v2	**81.15**
FF3	Word2vec, fastText, Glove	100 x 3 x 6	3D_FF3_v2	78.69
Floods				
F1	Word2vec, fastText, Glove	100 x 3 x 8	3D_F1_v2	**71.00**
F2	Word2vec, fastText, Glove	100 x 3 x 7	3D_F2_v2	68.67
F3	Word2vec, fastText, Glove	100 x 3 x 6	2D_F3_v2	61.00

3.2. Discussion

A comparison of the best results on the earthquake natural disaster messages dataset can be seen in Figure 6. The results obtained in making a classification model with 2D CNN are the same two accuracy values, namely 2D_E1G and 2D_E1_All.

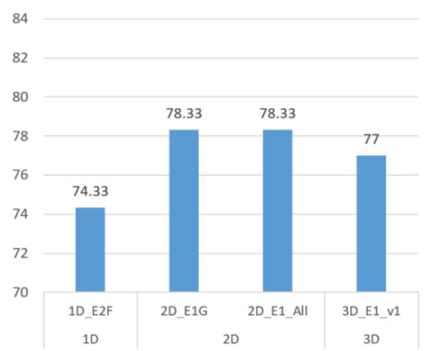

Figure 6. Comparison of the accuracy of earthquake natural disaster messages.

A comparison of the best results on the forest fire natural disaster messages dataset can be seen in Figure 7. In these results, the 1D CNN classification model values have the same two accuracies, namely 1D_FF1W and 1D_FF2_All.

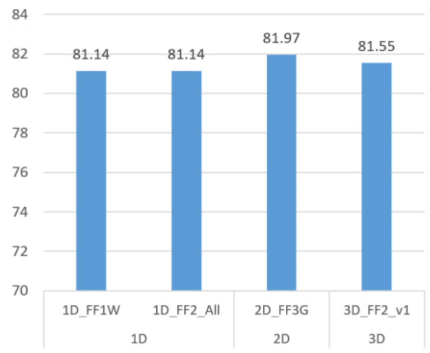

Figure 7. Comparison of message accuracy for forest fire natural disasters.

A comparison of the best results on the flood natural disaster message dataset can be seen in Figure 8.

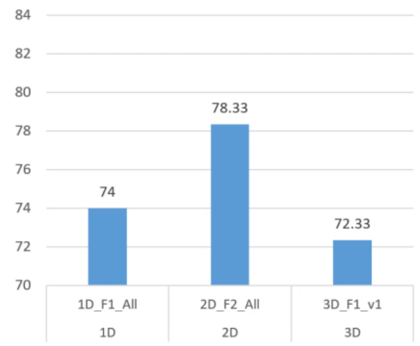

Figure 8. Comparison of the accuracy of flood natural disaster messages.

The results of this study indicate that the best results shown above mean that 2D CNN consistently gives better accuracy values than the results of 1D and 3D CNN.

The average accuracy was calculated for all 1D CNN, 2D CNN, and 3D CNN to determine the best feature extraction technique and classification method. The results can be seen in Figure 9. Comparison of average accuracy based on feature extraction and classification techniques. This calculation distinguishes between the formation of 3D data types 1 and 2. The results of this comparison provide information that the formation of 2D and 3D data type 1 has the same average accuracy and is the best compared to the formation of 1D data. The results of this comparison also show that the formation of 3D data by following the method of forming images based on three colour channels is better than the method of forming 3D data based on frames in the video.

The average accuracy of each technique used is measured to see how the statistical technique of implementing word padding influences classification performance. The results can be seen in Figure 10.

DOI: https://doi.org/10.29207/resti.v6i6.4525

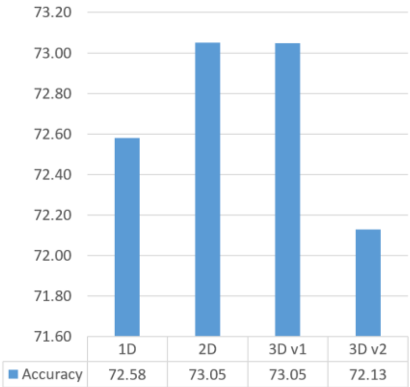

Figure 9. Comparison of average accuracy based on feature extraction and classification techniques.

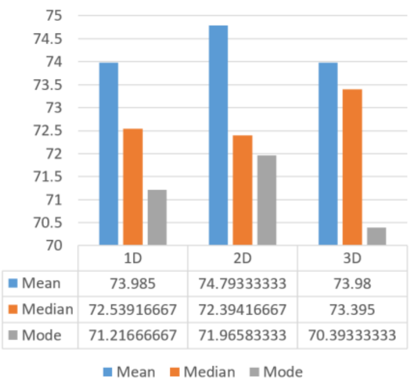

Figure 10. Comparison of average accuracy based on word padding technique in sentences.

From this comparison, it can be seen that the implementation of word padding with a mean value always gives a better value than the median and mode techniques. If you look at the data on the number of words in Table 3, it shows that the number of words produced by the mean technique is more than the other techniques, so when feature extraction is carried out, it will produce a greater number of features. This gives the data has more information so that it increases the accuracy value compared to data made with the median and mode techniques.

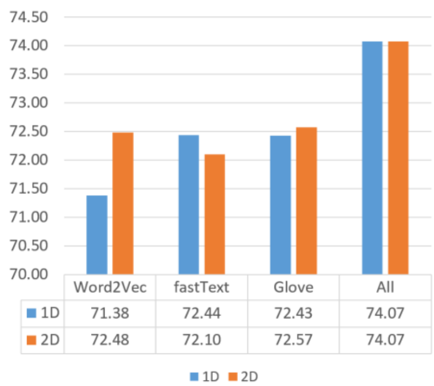

Figure 11. Comparison of the average accuracy based on the word embedding technique.

The results of this study can also show which word embedding technique can provide the best accuracy value. The results of the comparison can be seen in Figure 11. These results show that the structured data formed by word2vec, fastText, and Glove structures provide approximately the same accuracy. As mentioned in the Introduction section, the three-word embeddings used have their respective strengths and weaknesses. By combining structured data from the feature extraction output from each technique, the shortcomings of one technique are covered by other techniques. As a result, combining structured data using the three-word embedding techniques produces an accuracy of around 74.07%, which is better than the three-word embeddings that stand alone.

Table 8 shows the performance of the classification model from previous studies using the same dataset.

Table 8. Classification performance results from previous research.

Dataset	Method		Accuracy (%)
	Feature Extraction	Classification	
Earthquakes [7]	Lexicon	SVM	64.13
Forest fires [11]	word embedding vector	2D CNN	81.87
Floods [6]	vector space representation	SVM	77.87

Figure 12 shows a comparison of the performance of the classification model carried out in this study with previous research. The accuracy value displayed in the research results section is the best accuracy value for each natural disaster case.

Accuracy values for cases of earthquakes [7] and flood [6] obtained in this study are better when compared to previous studies.

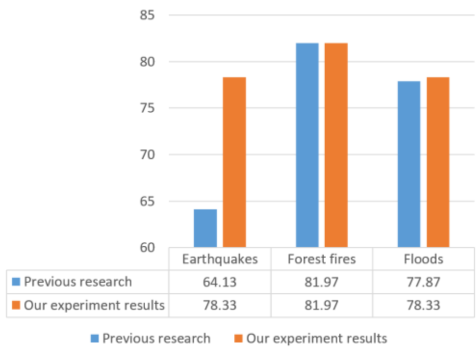

Figure 12. Classification performance results from previous research.

Meanwhile, when compared with research [11] for cases of forest fires, the accuracy values obtained from this study are the same. In previous studies, the number of words used in feature extraction was the maximum number of words, while this study only used six words. This means that the accuracy obtained in this study was

DOI: https://doi.org/10.29207/resti.v6i6.4525

obtained with a smaller number of features compared to previous studies [11].

4. Conclusion

To carry out a comprehensive study of classification techniques based on 1D CNN, 2D CNN, and 3D CNN in the case of natural disaster message classification, multi-dimensional data formation based on three-word embedding was carried out, namely Word2vec, fastText and Glove. In addition, the word padding technique is also applied to produce text messages with the same number of words. The combination of these techniques resulted in 90 structured data, which were then used to create a classification model. By calculating the performance of the classification model, the best technique for classifying natural disaster messages can be identified.

The mean is the technique of equalizing the number of words (word padding) used to produce a classification model with the best performance. This technique produces fewer features than the mode or median techniques. Structured data with a small number of features theoretically makes computation time faster. This is of more value than the word padding technique with the mean.

2D data and 3D type 1 data dimensions provide better performance when used on 2D CNN and 3D CNN compared to 1D and 3D type 2 data. This can be seen from the higher average accuracy of both. In contrast, the word embedding technique that can provide the best classification performance is Glove. However, the technique proposed in this study, namely the formation of 1D and 2D data by combining three-word embedding techniques, works better than data formed by a single-word embedding technique.

In this study, the 2D data formation technique by combining three-word embedding and 3D techniques can be explored more deeply to improve the performance of the classification model. In future research, another architecture will be used for 2D CNN and 3D CNN and perform hyperparameter tuning. In addition, an implementation and comparison study was also carried out with a classification method based on the Recurrent Neural Network (RNN) [23] and the Bidirectional Encoder Representations from Transformers model (BERT) [24].

Acknowledgement

The computer used in this study was provided by the Data Science Laboratory, Computer Science Department, Faculty of Mathematics and Natural Sciences, Lambung Mangkurat University. This work was supported by Program Dosen Wajib Meneliti (PDWM) from Grant of PNBP Lambung Mangkurat University.

References

[1] D. Wu and Y. Cui, “Disaster early warning and damage assessment analysis using social media data and geo-location information,” Decis. Support Syst., vol. 111, pp. 48–59, 2018.
[2] K. Muniz-Rodriguez et al., “Social media use in emergency response to natural disasters: a systematic review with a public health perspective,” Disaster Med. Public Health Prep., vol. 14, no. 1, pp. 139–149, 2020.
[3] A. Devaraj, D. Murthy, and A. Dontula, “Machine-learning methods for identifying social media-based requests for urgent help during hurricanes,” Int. J. Disaster Risk Reduct., vol. 51, p. 101757, 2020.
[4] K. Zahra, M. Imran, and F. O. Ostermann, “Automatic identification of eyewitness messages on twitter during disasters,” Inf. Process. Manag., vol. 57, no. 1, p. 102107, 2020, doi: 10.1016/j.ipm.2019.102107.
[5] K. Y. Firlia, M. R. Faisal, D. Kartini, R. A. Nugroho, and F. Abadi, “Analysis of New Features on the Performance of the Support Vector Machine Algorithm in Classification of Natural Disaster Messages,” in 2021 4th International Conference of Computer and Informatics Engineering (IC2IE), 2021, pp. 317–322.
[6] M. K. Delimayanti, R. Sari, M. Laya, M. R. Faisal, Pahrul, and R. F. Naryanto, “The effect of pre-processing on the classification of twitter’s flood disaster messages using support vector machine algorithm,” Proc. ICAE 2020 - 3rd Int. Conf. Appl. Eng., no. February 2021, 2020, doi: 10.1109/ICAE50557.2020.9350387.
[7] S. M. Nooralifa, M. R. Faisal, F. Abadi, R. A. Nugroho, and M. Aziz, “Identifikasi Otomatis Pesan Saksi Mata Pada Media Sosial Saat Bencana Gempa,” KLIK-KUMPULAN J. ILMU Komput., vol. 8, no. 2, pp. 129–138, 2021.
[8] M. R. Faisal, R. A. Nugroho, R. Ramadhani, F. Abadi, R. Herteno, and T. H. Saragih, “Natural Disaster on Twitter: Role of Feature Extraction Method of Word2Vec and Lexicon Based for Determining Direct Eyewitness,” Trends Sci., vol. 18, no. 23, p. 680, 2021.
[9] B. Jang, M. Kim, G. Harerimana, S. Kang, and J. W. Kim, “Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism,” Appl. Sci., vol. 10, no. 17, p. 5841, 2020.
[10] N. A. Hasanah, N. Suciati, D. Purwitasari, and others, “Pemantauan Perhatian Publik terhadap Pandemi COVID-19 melalui Klasifikasi Teks dengan Deep Learning,” J. RESTI (Rekayasa Sist. Dan Teknol. Informasi), vol. 5, no. 1, pp. 193–202, 2021.
[11] Rinaldi, M. R. Faisal, M. I. Mazdadi, R. A. Nugroho, F. Abadi, and Others, “Eye Witness Message Identification on Forest Fires Disaster Using Convolutional Neural Network,” J. Data Sci. Softw. Eng., vol. 2, no. 02, pp. 100–108, 2021.
[12] J. Ochoa-Luna and D. Ari, “Word Embeddings and Deep Learning for Spanish Twitter Sentiment Analysis,” in Information Management and Big Data, 2019, pp. 19–31.
[13] D. Alita and A. R. Isnain, “Pendeteksian Sarkasme pada Proses Analisis Sentimen Menggunakan Random Forest Classifier,” J. Komputasi, vol. 8, no. 2, pp. 50–58, 2020.
[14] Y. Goldberg and O. Levy, “word2vec Explained: deriving Mikolov et al.’s negative-sampling word-embedding method,” arXiv Prepr. arXiv1402.3722, 2014.
[15] S. Khomsah, R. D. Ramadhani, S. Wijaya, and others, “The Accuracy Comparison Between Word2Vec and FastText On Sentiment Analysis of Hotel Reviews,” J. RESTI (Rekayasa Sist. Dan Teknol. Informasi), vol. 6, no. 3, pp. 352–358, 2022.
[16] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

DOI: https://doi.org/10.29207/resti.v6i6.4525

[17] F. Anistya, E. B. Setiawan, and others, “Hate Speech Detection on Twitter in Indonesia with Feature Expansion Using GloVe,” J. RESTI (Rekayasa Sist. Dan Teknol. Informasi), vol. 5, no. 6, pp. 1044–1051, 2021.

[18] J. He and X. Fan, “Evaluating the performance of the k-fold cross-validation approach for model selection in growth mixture modeling,” Struct. Equ. Model. A Multidiscip. J., vol. 26, no. 1, pp. 66–79, 2019.

[19] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A survey of convolutional neural networks: analysis, applications, and prospects,” IEEE Trans. neural networks Learn. Syst., 2021.

[20] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, “1D convolutional neural networks and applications: A survey,” Mech. Syst. Signal Process., vol. 151, p. 107398, 2021.

[21] D. Kartini, A. Farmadi, D. T. Nugrahadi, P. Pirjatullah, and others, “Perbandingan Nilai K pada Klasifikasi Pneumonia Anak Balita Menggunakan K-Nearest Neighbor,” J. Komputasi, vol. 10, no. 1, pp. 47–53, 2022.

[22] A. Nugroho, A. B. Gumelar, A. G. Sooai, D. Sarvasti, P. L. Tahalele, and others, “Perbandingan Performansi Algoritma Pengklasifikasian Terpandu Untuk Kasus Penyakit Kardiovaskular,” J. RESTI (Rekayasa Sist. Dan Teknol. Informasi), vol. 4, no. 5, pp. 998–1006, 2020.

[23] S. Bodapati, H. Bandarupally, R. N. Shaw, and A. Ghosh, “Comparison and analysis of RNN-LSTMs and CNNs for social reviews classification,” in Advances in Applications of Data-Driven Computing, Springer, 2021, pp. 49–59.

[24] H. Jwa, D. Oh, K. Park, J. M. Kang, and H. Lim, “exBAKE: Automatic Fake News Detection Model Based on Bidirectional Encoder Representations from Transformers (BERT),” Appl. Sci., vol. 9, no. 19, 2019, doi: 10.3390/app9194062.

DOI: https://doi.org/10.29207/resti.v6i6.4525