Terakreditasi SINTA Peringkat 2
Surat Keputusan Dirjen Penguatan Riset dan Pengembangan Ristek Dikti No. 10/E/KPT/2019
masa berlaku mulai Vol. 1 No. 1 tahun 2017 s.d Vol. 5 No. 3 tahun 2021

Terbit online pada laman web jurnal: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 4 No. 4 (2020) 648 – 654 ISSN Media Elektronik: 2580-0760

Hoax Detection on Twitter using Feed-forward and Back-propagation Neural Networks Method

Crisanadenta Wintang Kencana[1], Erwin Budi Setiawan[#2], Isman Kurniawan[3]
[1]School of Computing, Informatics, Telkom University
[2,3]Research Center of Human Centric Engineering, Telkom University
[1]crisanadenta@student.telkomuniversity.ac.id, [2]erwinbudisetiawan@telkomuniversity.ac.id,
[3]ismankrn@telkomuniversity.ac.id

Abstract

Social media is one of the ways to connect every individual in the world. It also used by irresponsible people to spread a hoax. Hoax is false news that is made as if it is true. It may cause anxiety and panic in society. It can affect the social and political conditions. This era, the most popular social media is Twitter. It is a place for sharing information and users around the world can share and receive news in short messages or called tweet. Hoax detection gained significant interest in the last decade. Existing hoax detection methods are based on either news-content or social-context using user-based features. In this study, we present a hoax detection based on FF & BP neural networks. In the developing of it, we used two vectorization methods, TF-IDF and Word2Vec. Our model is designed to automatically learn features for hoax news classification through several hidden layers built into the neural network. The neural network is actually using the ability of the human brain that is able to provide stimulation, process, and output. It works by the neuron to process every information that enters, then is processed through a network connection, and will continue learning to produce abilities to do classification. Our proposed model would be helpful to provide a better solution for hoax detection. Data collection obtained through crawling used Twitter API and retrieve data according to the keywords and hashtags. The neural networks highest accuracy obtained using TF-IDF by 78.76%. We also found that data quality affects the performance.

Keywords: *hoax, Twitter, feed-forward, back-propagation, TF-IDF, Word2Vec.*

1. Introduction

Social media is a platform used by the public to share information. People turn to social media as a communication place. Social media has exploded as a category of online discourse where people create, share, and bookmark content at a prodigious rate. Examples include Facebook, MySpace, Digg, Twitter, and JISC listservs on the academic side. Because of easy to use, speed, and reach, social media is fast changing the public discourse in society and setting trends and agendas in topics that range from the environment and politics to technology and the entertainment industry [1]. Likewise, with the hoax news that also developed among social media users. Fake news or hoax news is a type of fabricated story that has no basis in fact, provably false having mass appeal and published under the guise having a legitimate look and feel [2]. Hoaxes (the term origins from hocus to trick) are more or less present throughout the entire history of humanity. The general intention of hoax creator is to persuade or manipulate other people to do or prevent pre-established actions, mostly by using a threat or deception [3]. One of the most used social media is Twitter.

Twitter is an extremely popular online microblogging based social media site that was launched on July 13, 2006. It has a substantial user base, consisting of several millions of users. It can be considered a directed social network, where each user has a set of subscribers known as followers. Each user submits periodic status updates, known as tweets, that consist of short messages of the maximum size of 140 characters [1]. Users can interact with friends all over the world through short messages or it called tweet. On Twitter, a variety of positive to negative news can be found, such as hoax, gossip, pornography, fraud, defamation, even self-harming. Deception is considered to disturb the public with information that cannot be fully trusted. According to [4], Twitter is the most popular social media platform in hoax news deployment.

Accepted by editor: 16-06-2020 | Final Revision: 19-07-2020 | Online Publication : 20-08-2020

Based on [5], they divided methods on how to detect hoax news. The first type is fake news identification using content-based methods that classify news based on the content of the information to be verified. The second type is identification using feedback-based methods that classify news based on the user responses it receives on social media. Lastly, the third type is intervention based solutions that provide computational solutions for actively identifying and containing the spread of false information, and methods to mitigate the impact from exposures to false information. The method used to detect hoaxes can be done manually or with the help of machine learning and artificial intelligence algorithm.

An automatic hoax detection using artificial intelligence can be more helpful. It collects content using a crawler engine to gain a dataset that can be labeled manually. Fake news detection is one of the emerging topics that has caught the attention of researchers across the world in the field of artificial intelligence. Despite receiving significant attention in the research community, fake news detection accuracy has not improve significantly due to insufficient context-specific news data. In contrast to the classical feature-based model, deep learning is advantageous because it doesn't require handcraft-based features, rather it recognizes the best feature set on its own for a specific issue or problem for classification [6]. There is a reason that recently artificial intelligence algorithms have started to work much better on lots of classification problems (text classification, image recognition, voice detection and so on) because hardware is cheaper and bigger datasets are available [7].

There are various literatures found in the field of text classification algorithms like decision tree, K-Nearest Neighbor, Naïve Bayes, and Artificial Neural Network [8]. Various research has been developed to classify hoax texts. Text classification research begins with an email classification that contains a hoax [9]. Based on [6], the fake news detection using deep Convolutional Neural Network classification has 98.36% accuracy. Convolutional Neural networks also have accomplished excellent performance in text classification. Based on [10] the fake news detection using Deep Neural Network classification has 96.77% accuracy. On the other hand, deep neural networks have demonstrated clear advantages for many machine learning problems [Sutskever et al., 2011; 2014; Devlin et al., 2014; Kombrink et al., 2011; Cho et al., 2014]. Based on [11], the fake news detection using Hybrid CNN and RNN models has 82% accuracy. Ma et al. [12] have employed a recurrent neural network (RNN) to capture contextual features from articles in contrast to machine learning algorithms that require hand-crafted features. The authors conclude that the implementation of a deep learning framework helped achieve better results as compared to existing techniques. Ruchansky et al. [13] have proposed a hybrid neural network model that combines the text, response, and source characteristics of a news article. Neural networks are a form of machine learning method that has been found to exhibit high accuracy and precision in clustering and classification of text [12]. They also prove effective in the prompt detection of spatio-temporal trends in content propagation on social media [11]. One of the researches, Alvaro et.al. [15] have used three different neural network architectures for fake news classification with the dataset: Getting Real about fake news and Fake News Corpus in their research. They have just considered the text of news articles in their research. They have achieved better classification results as compared to existing models using context-related fake news dataset.

In this study, the authors proposed feed-forward and back-propagation neural networks as the classification method. We decided FF & BP models that inspired by how the human brain works in processing information to detect hoax words. The initial stage of text classification is pre-processing. Pre-processing is the process of removing words that are commonly used and convert it into a basic word. The words will be tokenized into several n-grams in the form of unigram, bigram, and trigram. Then the words will be vectorized by two different methods that are Word2Vec and TF-IDF. We compared both as a vector representation to know which one performs well to increase neural network performance. Feature of text representation models used in this study such as count vectorizer and chi2-square. Precision, recall, F1-Score, and accuracy are used to validate the system performance. The goal of the research is to examine how this classification method works for a hoax detection problem that given a manually labeled dataset. The difference between this research and research on similar classification methods is that in this paper we proposed to detect hoax news on Twitter according to our topic which is Indonesian politics and compare the two different vectorization methods.

2. Research Method

2.1. Dataset

The dataset used in this study is tweet. To collect tweets, we do crawling process. Crawling is the process of taking data that is large or small inside a web page that can be stored in local storage, and the data is taken based on several keywords that are searched [16]. The crawling process was conducted in March 13 – 16, 2020. The crawling used Twitter as an object to retrieve data with the help of the API (Application Program Interface) that has been provided by Twitter developers as a link from the system to Twitter so that data can be retrieved and processed [17]. Data is retrieved using consumer_key, consumer_secret, access_token, and access_secret obtained from Twitter Developer. Users must register an account that can be connected to Twitter Developer in advance to be able to access the data we want to retrieve from Twitter by using a crawling tool. The crawling tool uses the Python programming language. There is no limit to the amount of data in crawling. We only use a time limit. It means that the tweet taken within a period of one week in accordance with the selected keywords and hashtags shown in Table 1. We get 50.646 tweets containing 25.021 with hoax class and 25.624 with non-hoax class.

Table 1 Hashtag/ Keyword

Hashtag/ Keyword	Amount
Jokowi	5.987
#BanjirJakarta2020	3.448
#CoronaVirusIndonesia	10.352
#PecatAniesBaswedan	10.285
#GubernurTerbodoh	10.223
#GantiGubernurDKI	10.351
Total	50.646

The keywords and hashtags were chosen are being widely discussed by the public and trending topic in Indonesia. From the topics raised, many politicians who participated expressed their opinions on Twitter, and citizens also participated in voicing their opinions in the reply column. Not infrequently, citizens use impolite sentences and contain provocation. Such moments are used by certain groups/individuals to divide the unity of Indonesia, where they spread the hoax news to bring someone down. Therefore, the authors raise the topics to be classified and analyzed. Data that has been collected to be processed into a classification system needs to have classes or labels. The class labels used are hoax to represent hoax class and valid as non-hoax class. The labeling process was done manually and analyzed by the authors refers to hoax features shown in Table 2.

Table 2 Description of Hoax Identification Feature

No.	Feature	Description
1.	Tweet	The contents of the tweet whether they contain hoax definitions.
2.	URL	The URL used is from a trusted source.
3.	Location	The user's location when sharing a tweet.
4.	Retweet	Number of retweets.
5.	Mention	The contents of the tweet whether mentioning someone.
6.	Username	The name used is real or a pseudonym.
7.	Hashtag	Hashtag that used.
8.	Verified Account	User accounts have been verified.
9.	Following > Followers	The following number is more than the number of followers.

2.3. Pre-processing

Twitter data contains unstructured and complicated words. Thus, the preprocessing of data is necessary to eliminate all characters that do not have a significant influence and leave important words [18]. Pre-processing data can directly improve system performance significantly by reducing the number of features. Figure 2 shows the step by step of pre-processing.

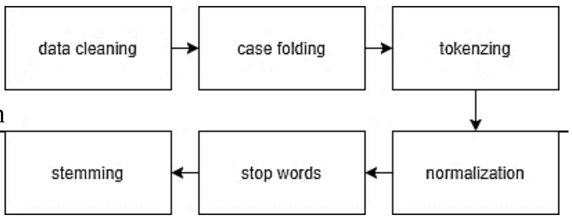

Figure 1 Pre-processing

a. Data cleaning

Data cleaning is a method to remove noise, inconsistent data, and errors in the training data. Noises such as punctuation marks (points, commas, question marks, exclamation marks, etc.), numeric numbers, and other characters ($, %, ^, &, #, *<, etc.). This should enable the use of better and representative data set to develop a reliable classification model [19].

b. Case folding

Case folding is the process of converting capital letters from input data to lowercase all (lowercase). It is done so that all letters in the input data become uniform [20].

c. Normalization

Normalization is particularly useful for classification algorithms or distance measurements. If using the neural network back-propagation algorithm for classification mining, normalizing the input values for each attribute measured in the training samples will help speed up the learning phase [21]. Normalization helps prevent attributes with initially large ranges from outweighing attributes with initially smaller ranges [22].

d. Stop words

The tweets still contain words that are considered to have no influence in classifications, such as conjunctions. These words are entered to stop word list. If it is included in the stop word, the word will be deleted or removed from the data [20].

e. Stemming

Stemming is the process of eliminating the word that has become a necessary word. This can be done by removing the prefix or suffix from a word [20].

f. Tokenizing

Tokenizing is a process to separate words separated by spaces. This is done to facilitate the next preprocessing stage [20]. The classification process is undertaken using the tokenizing n-gram technique like unigram, bigram, and trigram. N-gram models operate by tokenizing documents (breaking these into words) and calculating the number of times every sequence of words appears in a given document corpus [23].

2.5. TF-IDF

TF-IDF is a method used to weigh the position of words in a document. TF-IDF calculated values for each word in a document through an inverse proportion of the frequency of the word in a particular document to the percentage of documents the word appears in. TF-IDF works by determining the relative frequency of words in a specific document compared to the inverse proportion of that word over the entire document corpus [24]. TF states the number of words that appear in the document, while IDF shows how often the word appears in the document. The TF-IDF algorithm formula is [20]:

$$W_{ij} = tf_{ij} * Idf_j$$

$$Idf_j = (\log(\frac{N}{df})) \qquad (1)$$

With W_{ij} being the weight of the i document to the j word, tf_{ij} is the number of occurrences of the intended keywords in a document, Idf_j indicates the number of keywords in the aggregate data, N is the total data, and df is many documents containing keywords.

2.6. Word2Vec

Word2Vec is a method for representing each word as a vector using the concept of neural networks. Word2vec has two layers of neural networks that process text. Input is a corpus, and the output is a set of vectors. Word2Vec implements neural networks to calculate the contextual and semantic similarity of each word input. Word2vec is a group of models used to produce word embedding. Vector representations of a word using the word2vec model can be useful in a variety of natural language processing tasks [25].

The purpose and usefulness of Word2vec are to group the vectors of similar words together in vector-space. That is, it detects similarities mathematically. Word2vec creates vectors that are distributed numerical representations of word features, features such as the context of individual words [26]. Word2vec has two different techniques, namely, Continuous Bag of Words (CBOW) and Skip-gram. The Word2Vec technique used in this study is CBOW. Skip-gram works well with a small data train, and CBOW works faster than skip-gram with better accuracy for frequent words. So, this study, with 50.646 amount of data and many of frequent words, this study used the CBOW method to make the system more efficient. The CBOW goal is to predict the words given by the words around it. During training, the learning algorithm optimizes vector for each word using the CBOW. The architecture of CBOW can be seen in Figure 3.

2.4. Feed-Forward and Back-Propagation Neural Networks

Feed-forward and back-propagation are part of Artificial Neural Network (ANN). ANN is a machine learning technique for classification and has a self-adaptive nature that is driven by data so that it can adjust to data. It has advantages, such as tolerance of data that contains much noise and learning from data. However, besides these advantages, it also has many disadvantages. First, the training of the neural network is costly and time-consuming. Training of the network plays an essential role in classification accuracy [27].

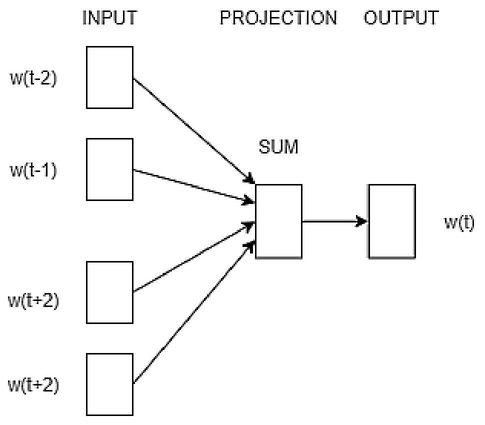

Figure 2 CBOW [21]

ANN can be divided into three parts of the layer, such as the input layer, hidden layers, and the output layer. The input layer is responsible for receiving information (data), signals, features, or measurements from the external environment. Hidden layers are composed of neurons that are responsible for extracting patterns associated with the process or system being analyzed. Hidden layers perform most of the internal processing from a network. The hidden layer can be more than one called multi-layer. The multi-layer feed-forward and back-propagation architecture used in this study are shown in Figure 4. Last, the output layer is also composed of neurons and thus is responsible for producing and presenting the final network outputs, which result from the processing performed by the neurons in the previous layers [28].

Each layer has an activation function whose purpose is to limit the output of neurons within a reasonable range of values. This study used ReLU in the input layer and hidden layers and sigmoid functions used in the output layer. As emphasized by Krizhevsky [29], the advantages of using the ReLU function include faster training speed, decreased saturation problems, a smaller number of epochs, and usually fewer samples. However, the ReLU activation function has the disadvantages of potentially causing a neural network to explode (retain too much information) or die (retain too little information) during learning calculations [29]. The sigmoid function exists between (0 – 1); therefore, it is mainly used for models that have to predict the probability as an output (hoax and non-hoax). ANN learns from data by optimizing objective functions. Gradient descent is one of the most popular algorithms for optimizing but tends to be slow in calculating big data. Therefore SGD with momentum is chosen because it can help accelerate SGD in the relevant direction and reduce oscillation [30] [31].

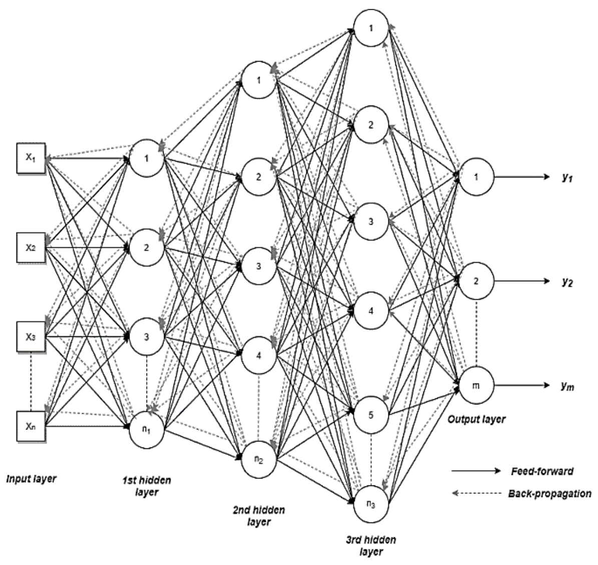

Figure 3 Multi-Layer Feed-Forward and Back-Propagation Architecture

Feed-forward algorithm is trained with back-propagation algorithm. The steps of the feed-forward algorithm are as follows:

a. Initialize all weights (w) with random numbers, specify epoch, learning rate (α), and the number of neurons in the hidden layer.
b. Each *input layer* ($x1,\ i = 1, 2, 3, \dots, n$) receives the xi signal and forwards the signal to all units in the hidden layer.
c. Each *hidden layer* ($z1,\ j = 1, 2, 3, \dots, p$) adds up the weight of the input signal. As in the following equation:

$$z^l = (w^l a^{l-1}) + b^l$$

The hidden layer used ReLU activation function shown as follows:

$$f(x) = \max(0, x) = \begin{cases} x_i \ if\ x_i \ \geq 0 \\ 0, if\ x_i < 0 \end{cases} \tag{4}$$

d. Each unit *output* ($yk,\ k = 1, 2, 3, \dots, m$) adding up the weight of the input signal on the hidden layer, shown by the equation:

$$y^l = (w^l a^{l-1}) + b^l \tag{5}$$

Where *w* is the weight of the neuron, *a* is the input layer, and *l* is the current layer. The output layer used sigmoid activation function shown as follows:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{6}$$

$$z = \sum_{i=0}^{n} XiWi + b \tag{7}$$

Where σ is sigmoid activation function, *z* is the computational result of input with weight and bias, xi is the input of the i neuron, wi is the weight of i neuron, and *b* is the bias.

The next step in the derivation of the backpropagation algorithm consists of defining a function that represents the approximation error, whose purpose is to measure the deviation of the responses produced by the output neurons of the network with respect to the corresponding desired values [28].

2.8 Validation Parameter

The performance of classification algorithms is pace out by F1-score, recall, precision, and accuracy. These measures consider the value of the confusion matrix. The confusion matrix contains information that compares the results of classification carried out by the system with the results of classification that should be [32]. The confusion matrix table is shown as follows:

Table 3 Confusion Matrix Table

Actual	**Predicted**	
	Positive	Negative
Positive	TP (True Positive)	FN (False Negative)
Negative	FP (False Positive)	TN (True Negative)

From table 3, accuracy values can be obtained. Accuracy is the proportion of the total number of correct predictions (8). F-1 score (9) is the average of recall (10) and precision (11). The accuracy value describes how accurately the system can classify data correctly. The equations are determined as follows:

$$\text{AC} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}}\ x\ 100\% \tag{8}$$

$$\text{F1} - \text{Score} = 2\ x\ \frac{\text{precision*recall}}{\text{precision+recall}} \tag{9}$$

$$\text{RE} = \frac{\text{TP}}{\text{TP+FN}} \tag{10}$$

$$\text{PRE} = \frac{\text{TP}}{\text{TP+FP}} \quad (3) \tag{11}$$

3. Result and Discussion

We have done define the architecture model and determine the hyperparameter used in hoax detection using feed-forward and back-propagation neural networks. Since the neural network has no definitive method to determine the model and hyperparameter, so it ultimately will come to trial and error. The parameters that define the model of neural network architecture are referred to as hyperparameters and the process of searching for the ideal model architecture is referred to as hyperparameter tuning. Therefore in this study, hyperparameters tuning have been carried out to get the best parameters. Choosing the best hyperparameters are challenging that must be solved for improvements in predictions. Hyperparameters are tuned by choosing the optimal parameter values for better accuracy. This process can be difficult and time-consuming. We used grid search hyperparameters tuning to improve the system performance by choosing the best parameter values that can be seen in Table 4. Grid search is used to find the optimal hyperparameters which result in the most accurate predictions.

Table 4 Tuning hyperparameter

Hyperparameter	Value
Learning rate	0.01
Momentum	0.9
Threshold	0.5
Dense layer	4
Number of epochs	100 - 200
Dropout	0.5
Hidden layer activation function	ReLU
Output layer activation function	Sigmoid
Optimizer	SGD with momentum

Dropout used to avoid over-fitting. It happens because the model is made too focused on training data, so it cannot make predictions correctly if given another similar dataset. This study also used weight and bias initializers that can be shown on table 5. Initializers define the way to set the initial random weights and bias of Keras layers. The aim of initialization is to prevent layer activation outputs from exploding or vanishing during the course of a forward pass through a neural network. Also, we used the random seed to initialize the pseudo-random number generator. The seed will always produce the same random tensor for a given shape and dtype.

Table 5 Keras Initializer

Dense Layer	Weight Initializer	Bias Initializer
1	Truncated Normal	He Normal
2	Truncated Normal	He Normal
3	Truncated Normal	He Normal
4	Glorot Normal	Glorot Normal

Since we used ReLU activation function, truncated normal and he normal are recommended initializer for neural network weights, bias, and filters to avoid dead neurons. Also, the glorot normal is recommended especially for output unit. It tries to keep the variance of the output gradient the same by initializing the weights to numbers that are not too small nor too big.

Table 6 Neurons Architecture

N-gram	Number of Input Neurons	Number of Hidden Neurons		
		1st	2nd	3rd
Unigram	23.991	500	250	100
Bigram	115.085	700	650	300
Trigram	110.601	700	650	400

Based on table 6, the number of hidden layers and the number of neurons in each of hidden layers must be carefully considered. Using too few neurons in the hidden layers will result in underfitting. But using too many neurons in the hidden layers may result in overfitting. There are some empirically-derived rules-of-thumb, the most commonly relied on is the optimal size of the hidden layer, such as:

a) The number of hidden neurons should be between the size of the input layer and the size of the output layer.
b) The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer.
c) The number of hidden neurons should be less than twice the size of the input layer.

But, there is also one additional rule of thumb that helps for supervised learning problems by the equation:

$$N_h = \frac{N_s}{(\alpha \; x \; (N_i + N_o)} \qquad (12)$$

Where:
N_h = Number of hidden neurons.
N_s = Number of training dataset.
N_i = Number of input neurons.
N_o = Number of output neurons.
α = Number of nonzero weights for each neuron (usually 2 – 10)

So that we considered the number of neurons in hidden layers in each unigram, bigram, and trigram with the rule-of-thumb. We found that in this study, in range 100-700 are the exact number of neurons in each hidden layer that does not produce overfitting or underfitting. And the output layer has 2 neurons based on its classes, which are hoax and non-hoax.

Our hoax detection using feed-forward and back-propagation neural networks will be compared with two vectorization methods that are TF-IDF and Word2Vec. We found that training using TF-IDF vectorization requires relatively consuming a long time reaching 6-12 hours and the Word2Vec faster by reaching 2 hours. The dataset is divided into 80% train data and 20% test data. We also used 20% of train data to be validation data to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters. The validation data also compared to train data that can be seen in the graphics on Figures 4, 5, 6 using TF-IDF and Figures 7, 8, 9 using Word2Vec. Classification performance results can be seen in tables 7 and 8. Table 7 shows the classification using TF-IDF vectorization.

Table 7 Classification using TF-IDF vectorization

N-gram	AC	PRE	RE	F1-Score
Unigram	78.76%	82.43%	75.49%	0.7880
Bigram	61.36%	66.47%	59.12%	0.6257
Trigram	61.32%	61.41%	63.82%	0.6259

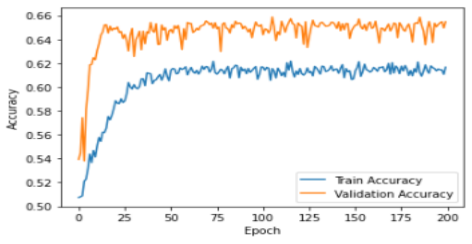

Figure 4 Validation Data vs Train Data Unigram (TF-IDF)

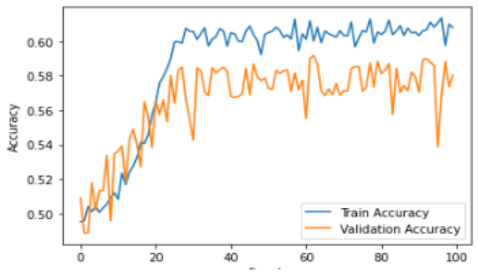

Figure 5 Validation Data vs Train Data Bigram (TF-IDF)

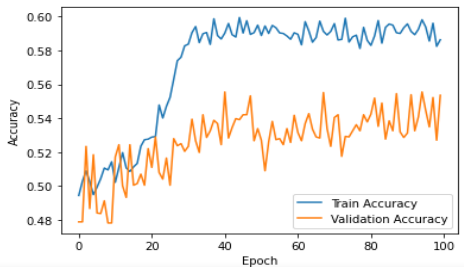

Figure 6 Validation Data vs Train Data Trigram (TF-IDF)

From table 7, we know that feed-forward and back-propagation using the TF-IDF vectorization method shows the highest accuracy is on the unigram model with 78.76%. This happens because the combination of words on bigram and trigram produces noises in dataset and it made unstable learning shown in graphics of bigram (Figure 5) and trigram (Figure 6). Meanwhile, table 8 shows the classification results using Word2Vec vectorization.

Table 8 Classification using Word2Vec vectorization

N-gram	AC	PRE	RE	F1-Score
Unigram	67.38%	70.24%	62.14%	0.6594
Bigram	66.16%	62.34%	57.53%	0.5983
Trigram	44.46%	44.45%	37.22%	0.4051

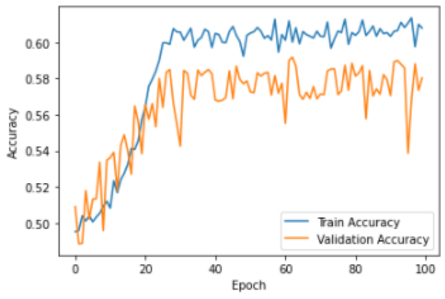

Figure 7 Validation Data vs Train Data Unigram (Word2Vec)

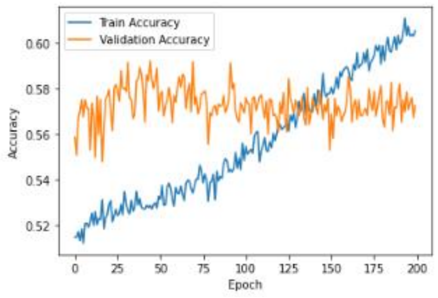

Figure 8 Validation Data vs Train Data Bigram (Word2Vec)

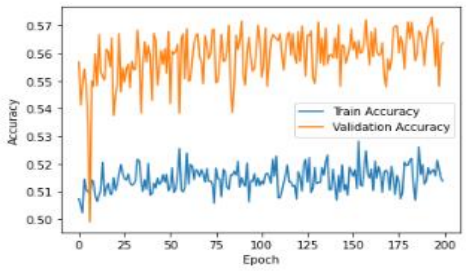

Figure 9 Validation Data vs Train Data Trigram (Word2Vec)

From table 8, feed-forward and back-propagation using Word2Vec vectorization also can be implemented and works much faster in learning than TF-IDF. However, the results are lower compared to TF-IDF. Word2Vec is about proportions of word occurrences in relations holding in the large corpus. It is demonstrated how to surface seemingly semantic relationships. The highest accuracy is on the unigram model with 67.38%. The graphics of bigram (Figure 8) and trigram (Figure 9) provide unstable learning because of too many words combine and the Word2Vec vectorization cannot look for the semantic relationships properly.

In this study, we found that feed-forward and back-propagation neural networks classification method can be used for hoax classification. To produce better performance of feed-forward and back-propagation, we also prove that using TF-IDF vectorization produces highest accuracy than Word2Vec. Word2Vec helps in going deeper into the document, measure semantic similarities between sentences, and helps to derive relations between a word and its contextual words. But TF-IDF helps in visualizing important words in document and topic modeling by using the importance score of words. Before they are into the neural network, each vector is normalized such that all elements of the vector add up to one. So, the frequency of each word is effectively converted to represent the probabilities of those words' occurrences in the document. Probabilities will activate nodes in the network and influence the document's classification. The effect of the combination of words on the bigram and trigram model also makes the system raises many noise words and causes unstable learning shown in bigram and trigram graphic figures. Precision and recall on unigram shows better results with the accuracy of the information generated compared to bigram and trigram. Based on analysis using CountVectorizer, table 9 shows the words that appear most frequently in the spread of hoaxes. We assume that these words is the most often used in hoax distribution with the aim of bringing down someone or making fake news to attract other's attention.

Figure 10 shows the top 20 of the most useful features selected by Chi2 feature selection along with its frequency. Chi2 statistics measure the lack of independence between classes, whether the tweet is hoax or non-hoax.

Table 9 Most Frequently Words

Unigram	Bigram	Trigram
Bakar	Bakar masjid	Kapolri tunda kiamat
Gubernur	Gubernur bodoh	Gubernur sekelas presiden
Presiden	Presiden gagal	Pemerintah gagal tangani
Bodoh	Pemerintah bodoh	Pemerintah modal bacot
Pemerintah	Pemerintah rahasiakan	Pemerintah modal bacot
Bacot	Jokowi bacot	Makan bacot luhut
Turunkan	Turunkan Jokowi	Turunakan gubernur Anies

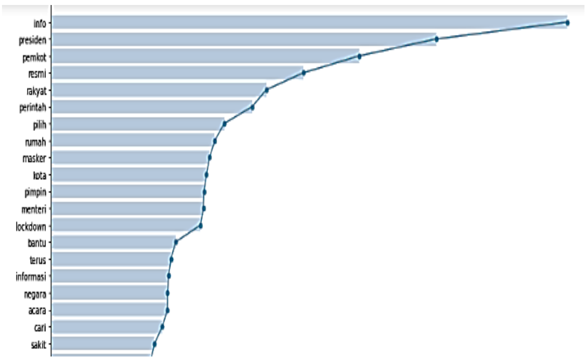

Figure 10 Top 20 Most Useful Feature Selected by Chi2 Statistics

In the first word, we assume that the word "info" is the most widely used to write non-hoax news. The word provides information to other users. The second word, we assume the word "presiden" is the most commonly used to write hoax news among the Indonesian political issues in order to bring its name reputation down or write news to protest against the its personal actions.

4. Conclusion

In this research, the classification system is built with Python programming language. This research has proposed to implement feed-forward and back-propagation neural networks to detect hoax on Twitter based on learning according to Indonesian political topics. Two vectorization methods were chosen to compare both algorithms to gain the best result in making the classification more effective. Based on results, using feed-forward and back-propagation neural networks as a classification method can be implemented in hoax detection. The feed-forward and back-propagation using TF-IDF vectorization improves highest performance than Word2Vec with 78.76% accuracy. TF-IDF works longer than Word2Vec, but the performance result shows that TF-IDF provides the highest accuracy. The choice of vectorization method also depends on the context of the dataset. However, classification using neural networks consumes time and memory.

Suggestions that can be considered for further research is improving the pre-processing and reducing noises to produce higher data quality that affects to system performance. Future researchers may be able to define the right neural network model and tuning hyperparameter that also affects accuracy.

References

[1] S. Asur and B. A. Huberman, “Predicting the future with social media,” *Proc. - 2010 IEEE/WIC/ACM Int. Conf. Web Intell. WI 2010*, vol. 1, pp. 492–499, 2010.

[2] B. Narwal, “Fake News in Digital Media,” *Proc. - IEEE 2018 Int. Conf. Adv. Comput. Commun. Control Networking, ICACCCN 2018*, pp. 977–981, 2018.

[3] J. C. Hernández, C. J. Herńndez, J. M. Sierra, and A. Ribagorda, “A first step towards automatic hoax detection,” *IEEE Annu. Int. Carnahan Conf. Secur. Technol. Proc.*, pp. 102–114, 2002.

[4] P. Assiroj, Meyliana, A. N. Hidayanto, H. Prabowo, and H. L. H. S. Warnars, “Hoax News Detection on Social Media: A Survey,” *1st 2018 Indones. Assoc. Pattern Recognit. Int. Conf. Ina. 2018 - Proc.*, pp. 186–191, 2019.

[5] K. Sharma, F. Qian, H. Jiang, N. Ruchansky, M. Zhang, and Y. Liu, “Combating fake news: A survey on identification and mitigation techniques,” *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 3, 2019.

[6] R. K. Kaliyar, A. Goswami, P. Narang, and S. Sinha, “FNDNet – A deep convolutional neural network for fake news detection,” *Cogn. Syst. Res.*, vol. 61, pp. 32–44, 2020.

[7] M. Granik and V. Mesyura, “Fake news detection using naive Bayes classifier,” *2017 IEEE 1st Ukr. Conf. Electr. Comput. Eng. UKRCON 2017 - Proc.*, pp. 900–903, 2017.

[8] R. Kumar and D. Verma, “Classification Algorithms for Data Mining: A Survey,” *Int. J. Innov. Eng. …*, vol. 1, no. 2, pp. 7–14, 2012.

[9] Y. Y. Chen, S.-P. Yong, and A. Ishak, “Email Hoax Detection System Using Levenshtein Distance Method,” *J. Comput.*, vol. 9, no. 2, pp. 441–446, 2014.

[10] S. Sneha, N. Fernandez, and S. Rao, “3HAN: A Deep Neural Network for Fake News Detection,” *Conf. Pap.*, no. October, pp. 118–125, 2017.

[11] O. Ajao, D. Bhowmik, and S. Zargari, “Fake news identification on Twitter with hybrid CNN and RNN models,” *ACM Int. Conf. Proceeding Ser.*, no. July, pp. 226–230, 2018.

[12] J. Ma *et al.*, “Detecting rumors from microblogs with recurrent neural networks,” *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2016-Janua, pp. 3818–3824, 2016.

[13] N. Ruchansky, S. Seo, and Y. Liu, “CSI: A hybrid deep model for fake news detection,” *Int. Conf. Inf. Knowl. Manag. Proc.*, vol. Part F1318, pp. 797–806, 2017.

[14] X. Zheng, Z. Zeng, Z. Chen, Y. Yu, and C. Rong, “Detecting spammers on social networks,” *Neurocomputing*, vol. 159, no. 1, pp. 27–34, 2015.

[15] Á. I. Rodríguez and L. L. Iglesias, “Fake News Detection Using Deep Learning Techniques,” *1st IEEE Int. Conf. Adv. Inf. Technol. ICAIT 2019 - Proc.*, pp. 411–415, 2019.

[16] S. Chakrabarti, “Crawling the Web,” *Min. Web*, pp. 17–43, 2003.

[17] J. Eka Sembodo, E. Budi Setiawan, and Z. Abdurahman Baizal, “Data Crawling Otomatis pada Twitter,” no. October 2018, pp. 11–16, 2016.

[18] S. N. Kane, A. Mishra, and A. K. Dutta, “Preface: International Conference on Recent Trends in Physics (ICRTP 2016),” *J. Phys. Conf. Ser.*, vol. 755, no. 1, 2016.

[19] P. Jeatrakul, K. W. Wong, and C. C. Fung, “Using misclassification analysis for data cleaning,” *IWACIII 2009 - Int. Work. Adv. Comput. Intell. Intell. Informatics*, no. January, 2009.

[20] A. Fauzi, E. B. Setiawan, and Z. K. A. Baizal, “Hoax News Detection on Twitter using Term Frequency Inverse Document Frequency and Support Vector Machine Method,” *J. Phys. Conf. Ser.*, vol. 1192, no. 1, 2019.

[21] L. Al Shalabi and Z. Shaaban, “Normalization as a Preprocessing Engine for Data Mining and the Approach of Preference Matrix,” *Proc. Int. Conf. Dependability Comput.*

Syst. DepCoS-RELCOMEX 2006, pp. 207–214, 2006.

[22] M. K. and J. P. J. Han, *Data Mining Concepts and Techniques Third Edition*. .

[23] M. Jimenez, C. Maxime, Y. Le Traon, and M. Papadakis, "On the impact of tokenizer and parameters on n-gram based code analysis," *Proc. - 2018 IEEE Int. Conf. Softw. Maint. Evol. ICSME 2018*, pp. 437–448, 2018.

[24] C. P. Medina and M. R. R. Ramon, "Using TF-IDF to Determine Word Relevance in Document Queries Juan," *New Educ. Rev.*, vol. 42, no. 4, pp. 40–51, 2015.

[25] X. Rong, "word2vec Parameter Learning Explained," pp. 1–21, 2014.

[26] E. B. Setiawan, D. H. Widyantoro, and K. Surendro, "Feature expansion using word embedding for tweet topic classification," *Proceeding 2016 10th Int. Conf. Telecommun. Syst. Serv. Appl. TSSA 2016 Spec. Issue Radar Technol.*, 2017.

[27] R. Bala and D. Kumar, "Classification Using ANN: A Review," *Int. J. Comput. Intell. Res.*, vol. 13, no. 7, pp. 1811–1820, 2017.

[28] I. N. da Silva, R. A. Flauzino, L. H. B. Liboni, S. F. dos R. A. Alves, and D. H. Spatti, *Artificial Neural Networks A Practical Course*, vol. 50, no. 2. 1954.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *ImageNet Classif. with Deep Convolutional Neural Networks*, pp. 1–1432, 2007.

[30] S. Ruder, "An overview of gradient descent optimization algorithms," pp. 1–14, 2016.

[31] N. Qian, "On the Momentum Term in Gradient Descent Learning Algorithms Acknowledgments," *Learning*, vol. 5213.

[32] A. K. Santra and C. J. Christy, "Genetic Algorithm and Confusion Matrix for Document Clustering," *Int. J. Comput. Sci. Issues*, vol. 9, no. 1, pp. 322–328, 2012.