MATEMATIKA: MJIAM, 2020, Volume 36, Number 2, 99–111

Deep Learning for Social Media Sentiment Analysis

Kartika Fithriasari*, Saidah Zahrotul Jannah and Zakya Reyhana
Department of Statistics, Faculty of Mathematics, Computing, and Data Science
Institut Teknologi Sepuluh Nopember
Kampus ITS Sukolilo, Surabaya 60111, Indonesia
*Corresponding author: kartika_f@statistika.its.ac.id

Article history
Received: 17 July 2019
Received in revised form: 3 March 2020
Accepted: 28 March 2020
Published online: 1 August 2020

Abstract Social media is used as a tool by many people to express their opinions. Sentiment analysis for social media is very important, as it allows information to be obtained about public opinion on government performance. The goal of this research is to learn about the opinions of Surabaya citizens, using deep learning methods. The data are extracted from the official Twitter accounts of the Surabaya government and a private radio station in Surabaya. The data are grouped into two categories: positive and negative sentiments. This research is conducted in three steps: data pre-processing, sentiment classification, and visualization. Data pre-processing is required before modelling approaches are applied. It is used to transform the unstructured text data into structured data. The data pre-processing consists of case folding, tokenizing, and the removal of stop words. Deep learning methods are then applied to the data. A Backpropagation Neural Network (BNN) and a Convolutional Neural Network (CNN) are used to perform the sentiment classification. The BNN and CNN are compared using various metrics, such as precision, sensitivity, and area under the receiver operating characteristic curve (AUC). A word cloud is then used to visualize the data and find the most frequent words in each class. The results show that the sentiment classification with CNN is better than that with the BNN because the values for the precision, sensitivity, and AUC are higher.

Keywords Back propagation Neural Network; Convolutional Neural Network; Sentiment Analysis; Social Media; Text Mining.

Mathematics Subject Classification 62P99, 68T50

1 Introduction

The Surabaya city government needs advice and suggestions from society in order to make service improvements. By utilizing social media, the process of delivering this aspiration will be faster. Sentiment analysis or opinion mining is a computational process to identify and classify opinions expressed in text [1]. The analysis is used to find the tendencies in sentiment or opinion.

These tendencies are classified into two categories: positive and negative. A classification method or algorithm that can analyse public opinion is needed so that the government can easily obtain information about public opinion.

Several classification methods have been developed, in various fields [2, 3]. In text mining, classification methods have also been developed [4, 5, 6]. An artificial neural network with a backpropagation algorithm (BNN) is one of the possible methods for text classification [7]. A convolutional neural network (CNN) has also been developed for sentence classification [8, 9]. CNN is a deep learning method that is a development of a multilayer perceptron (MLP). In this study, BNN and CNN methods were applied to Twitter data to perform a sentiment analysis.

The data were taken from the official Twitter accounts of the Surabaya city government and a private radio station in Surabaya. The sentiment of each sentence was classified into one of two categories: positive or negative. Before applying the two methods, the Twitter text was pre-processed, using case folding, tokenizing, and the removal of stop words. Case folding is a pre-process that converts all text into lowercase letters. Tokenizing is the process of splitting the text into individual words. Stop words are words that are not the unique or characteristic words of a document and so need to be removed [10]. In a neural network (NN), the input variables are the features or terms obtained after the pre-processing. To discover the topics in each sentiment group, a word cloud can be used. A word cloud is a visual method that displays how often words appear in a text dataset.

2 Data Source

The data used in this research were derived from two Twitter accounts, @sapawargaSby and @e100ss. @sapawargaSby is an official Twitter account of the Surabaya government which is used by citizens to deliver complaints or appreciation to the Surabaya government. @e100ss is an official account of *Radio Suara Surabaya*, which is a private radio station in Surabaya. Surabaya citizens often speak about their complaints, on subjects such as blackouts, potholes, traffic jams, and flooding. From these accounts, 1500 tweets were classified by the researchers into two categories: positive and negative sentiments. Those tweets were taken from 28 August until 15 October 2017. There were 299 tweets with a positive sentiment and 1201 with a negative sentiment. The data in this research were therefore imbalanced. The tweets were classified manually by observing the existence of positive or negative words in each tweet.

3 Backpropagation Neural Network (BNN)

A Backpropagation Neural Network (BNN) is a neural network that uses backpropagation as a supervised learning algorithm. It was developed as a mathematical and computational model for non-linear approximation functions [11]. The backpropagation algorithm looks for the minimum of the error function in the weight space using the method of gradient descent. The combination of weights that minimizes the error function is considered to be a solution of the learning program [12].

A neural network architecture consists of a combinational feedforward network. The type of architecture used in BNN is multi-layer perceptron. The back-propagation algorithm can also be described as the training of a multi-layer perceptron using gradient descent applied

to a sum-of-squares error function. The contribution of the backpropagation algorithm is its computational efficiency for evaluating such derivatives since, at this stage, the errors are propagated backwards through the network [13].

The architecture of a back-propagation neural network consists of different interconnected layers. It is shown in Figure 1.

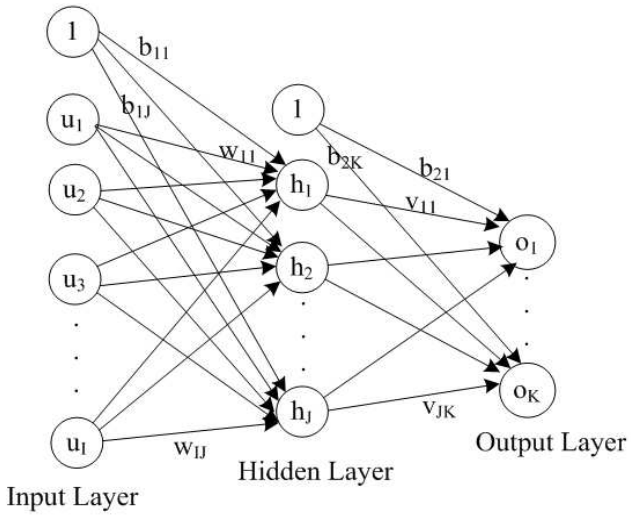


Figure 1: Architecture of a Backpropagation Neural Network

Figure 1 is an example of the architecture of a back-propagation neural network. It has three layers: an input layer, a hidden layer, and an output layer. Each layer has certain nodes. Before the network is trained, the weights of the network are randomly initialized. The back-propagation algorithm is used to compute the necessary corrections. There are four steps in the back-propagation neural network method [12].

i) Feed-forward computation

Suppose $(\boldsymbol{u}, \boldsymbol{o})$ is the input-output pair, then the model for the back-propagation neural network in Figure 1 is given as follows:

$$o_{lk} = b_{2k} + \left(\sum\nolimits_{j=1}^{J} v_{jk}.f\left(b_{1j} + \sum\nolimits_{i=1}^{I} w_{ij}u_{li}\right)\right) \tag{1}$$

where $f(.)$ is a logistic sigmoid function, w_{ij} and v_{jk} are the weights, $\boldsymbol{b_1}$ and $\boldsymbol{b_2}$ are the biases.

At the forward pass, input will be propagated to the output layer. The output prediction results will be compared with the target. The error function calculates the difference between the target and its prediction. Let $\mathbf{o}_{lk}$ and $\hat{\mathbf{o}}_{lk}$ is a vector $\mathbb{R}^n$ and n be the number of observations, then the function for a particular input l is

$$E_l = \frac{1}{2}\sum\nolimits_{k} \left(\mathbf{o}_{lk} - \hat{\mathbf{o}}_{lk}\right)^2 \tag{2}$$

If the value of the error function is still greater than the specified limit, then the weights are updated by following the back-propagation process. The weight and bias are adjusted based on the error obtained during the forward pass.

ii) Backpropagation to the output

The constant 1 is fed into the output and the network is run backwards. Incoming information to a node is added and the result is multiplied by the value stored in the left part of the unit. The result is transmitted to the left of the unit. The result collected by the input unit is the derivative of the network function with respect to x.

iii) Backpropagation to the hidden layer

The backpropagated error can be computed in the same way for any number of hidden layers, and the expression for the partial derivatives of E keeps the same analytic form.

iv) Weight updates

It is very important to make corrections to the weights only after the back-propagated error has been computed for all units in the network.

The algorithm is stopped when the value of the error function has become sufficiently small [12].

4 Convolutional Neural Network

A convolutional neural network (CNN) is one of the deep learning methods commonly applied to image data. This method is a development of MLP which is used to process one-dimensional data, while CNN is designed to process two-dimensional data. In complex data, MLP has weaknesses. If the number of hidden layers is greater than two, this often results in overfitting. To overcome this, a function is needed to transform the input data into a simpler form, making them easier to solve. In text mining, the input used is a sentence. In CNN, the sentence is represented as a matrix. Suppose there is a sentence with length n (the number of words in the sentence) and the dimension of the word vector is k, then $x_i \in \mathbb{R}^k$ is word vector i^{th} with k dimensions. The sentence can be represented as a combination of n words [8], as given below:

$$\boldsymbol{x_{1:n}} = \boldsymbol{x_1} \oplus \boldsymbol{x_2} \oplus \ldots \oplus \boldsymbol{x_n} \tag{3}$$

where $\oplus$ is the concatenation operator. An illustration for $\boldsymbol{x_{1:n}}$ is given in Figure 2.

The CNN architecture has three layers: the convolution layer, the pooling layer and the fully connected layer. The process carried out in those layers is a feature extraction and classification process [8].

- Feature Extraction Process

The feature extraction process happens in the convolution and pooling layers and transforms complex inputs into simpler inputs. This process makes the classification process faster. The convolution process occurs in the convolution layer. This is the main process of convolution using the output of the previous layer. This layer does not work in an analogous way to a neuron. The layer parameter is a set of learnable filters $\boldsymbol{w} \in \mathbb{R}^{\text{hk}}$, where h is the region size.

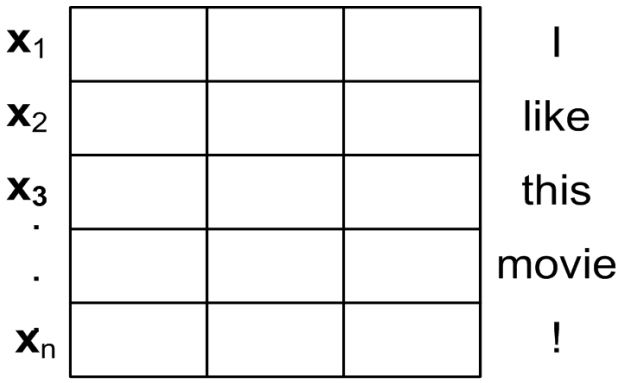


Figure 2: Representation of a Sentence with $k = 3$

The filter is shifted across all the words and calculates the dot products between the input and the filter values. The output of the convolution process is a new feature commonly called the activation map or feature folder $\boldsymbol{c} = [c_1 c_2 \dots c_{n-h+1}]$, where $c_i = f\left(b + \boldsymbol{w}.\boldsymbol{x}_{i:i+h-1}\right)$ and b are the biases and $f\left(.\right)$ is a nonlinear activation function.

The next process is the pooling process in the pooling layer. This reduces the spatial size and number of parameters in the network, speeds up computation, and controls for overfitting. The pooling layer works with spatial blocks that move along the size of the feature pattern. The dimension $\mathbf{c}$ depends on the values of n and h, so it will vary if the length of the sentence and the filter region size are different. To overcome this, the maximum pooling operation is used to get the maximum value of each $\mathbf{c}$ vector. This is meant to get the most important feature. One maximum value is obtained from one filter, $z = max(\mathbf{c})$. The pooling layer receives input directly and processes it to produce output in the form of vector features to be processed in the next layer. In order to obtain many features, many filters are needed with varying window sizes. These features are concatenated to form a single feature vector $\boldsymbol{z} = [z_1 z_2 \dots z_m]$. An example of the feature extraction process can be seen in Figure 3.

The activation function at each layer is applied with its alternating position. The activation function or transfer function is a non-linear function that allows a network to solve a non-trivial problem. Each activation function takes a value and performs mathematical operations. In the CNN architecture, the activation function lies in the calculation of the feature map at the output end or after the convolution or pooling calculation processes to produce a feature pattern. Some types of activation functions that are often used in research include sigmoid functions, the hyperbolic tangent (TanH), rectified linear units (ReLU), leaky ReLU (LReLU) and parametric ReLU.

- Classification Process

The classification process happens in fully connected layers. A neuron in this layer has full connections to all activations in the previous layers. Therefore, this layer accepts input from the output of the feature extraction layer in the form of a vector. The fully connected layer determines which of these features correlate best to a particular class. This layer outputs an N-dimensional vector representing the probability of each class, where N is the number of classes [14].

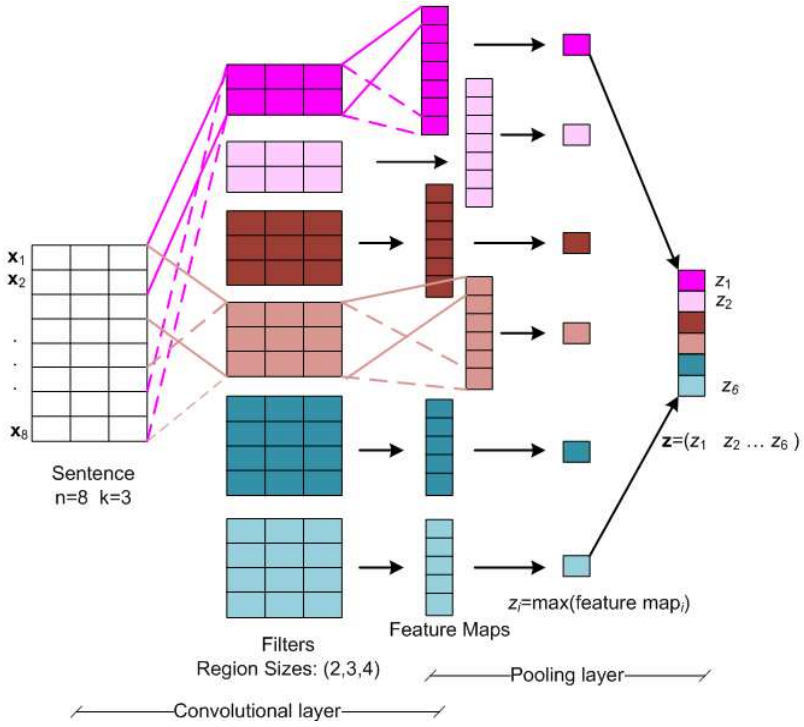


Figure 3: Feature Extraction Process

The convolution layer consists of neurons arranged in such a way as to form a filter. The filter is shifted throughout the data input section. The parameter that determines how many shifts of the filter are used is called the stride. If the value of the stride is 1, the convolution filter is shifted horizontally by 1 pixel, then is turned to be vertical. In this study, the stride used was 2. The padding, or zero padding, is the parameter that specifies the number of pixels (containing the value 0) that is added on each side of the input. In this case, the use of the padding "VALID" syntax means that the slides on the filter were above the sentence without padding the edges. The activation function in the convolution layer was the rectified linear unit (ReLU).

The next output from the convolution layer is changed into a one-dimensional feature vector. This process, called the flattening step, occurs in the pooling layer. The last process in this classification model happens in the dense layer. The training process is carried out by determining the batch size and epoch. Both parameters are determined by trial and error. The batch size is the number of observations used for parameter updates. The epoch is the number of iterations. In order to increase the generalization, dropout was used in this study in the fully connected (FC) layer. The activation function in the dense layer used in this research was logistic sigmoid.

5 Performance Metrics

Performance metrics in the case of classification are needed in order to choose the best model. In binary classification, performance metrics can be determined by a 2 x 2 confusion matrix. The matrix has four categories: True Positives (TP) are positive subjects correctly labelled

as positives, False Positives (FP) are negative subjects incorrectly labelled as positive, True Negative (TN) refers to negative subjects correctly labelled as negative, and False Negative (FN) corresponds to positive subjects incorrectly labelled as negative. The confusion matrix is shown in Table 1.

Table 1: Confusion Matrix

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

One of the widely used metrics for measuring the performance of the classifier is accuracy. This is the percentage of correctly classified positive and negative subjects. However, when the data are imbalanced or the prior probabilities of the classes are very different, the value of the accuracy can be misleading [15].

It is stated in [15] that the AUC, which is the area under the receiver operating characteristic (ROC) curve, is unaffected by skew. Precision-recall curves suggest that ROC may mask poor performance. This means that the AUC and the precision-recall are suitable for imbalanced data. Therefore, this research used these metrics.

The ROC curve is based on the False Positive Rate (FPR) and the True Positive Rate (TPR). An illustration of the ROC and the AUC is in Figure 4.

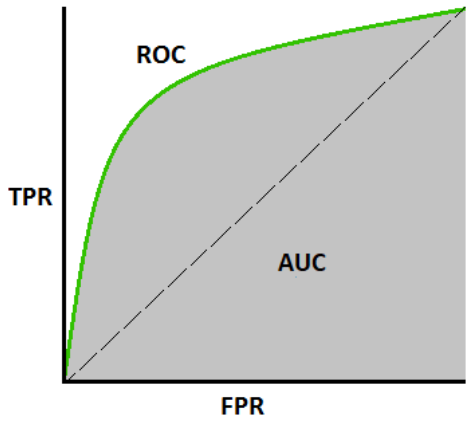


Figure 4: ROC and AUC

The FPR and TPR can be determined by:

$$FPR = \frac{FP}{FP+TN} \quad ; \quad TPR = \frac{TP}{TP+FN} \tag{4}$$

The AUC can be calculated from the equation in [16]:

$$AUC = \frac{TPR + FPR - 1}{2} \tag{5}$$

The value of the AUC ranges between 0 and 1. If the prediction results are all wrong, then the AUC value is 0. Otherwise, if the prediction values are all correct, then the AUC value is 1. Therefore, the method that has the highest value for the AUC is the best method.

Other performance metrics used in this research are precision and recall. In the sentiment analysis, there were two sentiments, positive and negative. Therefore, the precision and recall (sensitivity) are calculated for each sentiment. The best sensitivity value is 1 and the worst is 0. The positive and negative sensitivity can be determined by the formulae in [16]:

$$\text{Positive Sensitivity} = \frac{TP}{TP+FN} \quad ; \quad \text{Negative Sensitivity} = \frac{TN}{TN+FP} \tag{6}$$

Positive precision is calculated as the number of inputs correctly predicted as positive, divided by the total number of positive predictions. Positive precision calculates how many of the positively classified inputs were correct. Below are the formulae for positive and negative precision [16]:

$$\text{Positive Precision} = \frac{TP}{TP+FP} \quad ; \quad \text{Negative Precision} = \frac{TN}{TN+FN} \tag{7}$$

6 Results and Discussion

In this study, BNN and CNN were used to classify the sentiment of Twitter data obtained from the official Twitter accounts of the Surabaya government and a private radio station in Surabaya. The Twitter data were collected for ten days in August 2018. The number of tweets collected was 1500, and the tweets consisted of 1201 negative tweets and 299 positive tweets. Therefore, 80% of the data had a negative sentiment. The ratio between negative and positive sentiment is too wide, so the data were imbalanced.

Twitter data are unstructured data, so the data must be converted into structured data. The first step was to pre-process the data. The steps followed in the data pre-processing were removing the URL-link, removing retweets, removing the username, removing numbers, removing punctuation, case folding, removing stop words and tokenizing. The pre-processing and classification processes were done using Python.

In the removal process, unnecessary characters were removed because those were considered as noise. Next, the data were converted into the same letter form, which is lower case. This process is called case folding. Then, stop words were removed, to remove insignificant words and reduce the number of features or variables. This process aims to remove common words (for example, '*dan*', '*saya*', '*di*', '*pada*' and so on) and to leave terms that have more meaning. The list of stop words was based on a list of Indonesian stop words [17]. There were 816 words that were considered as stop words. The last step was then tokenizing. This process separates the words in each sentence into single words or terms.

The number of features (terms) before pre-processing was 4192. After pre-processing, the number of terms was reduced to 2975 terms. There were several terms that had a high frequency. Figure 5 shows the ten terms with the highest frequency after pre-processing.

The 2975 terms obtained after pre-processing were used as the input variables for the BNN. The data were divided into testing and training data by using K-fold with $K = 10$. The BNN architecture had 2975 input nodes. The number of hidden layers was one, and the number of hidden nodes that were tried ranged from 5 to 75.

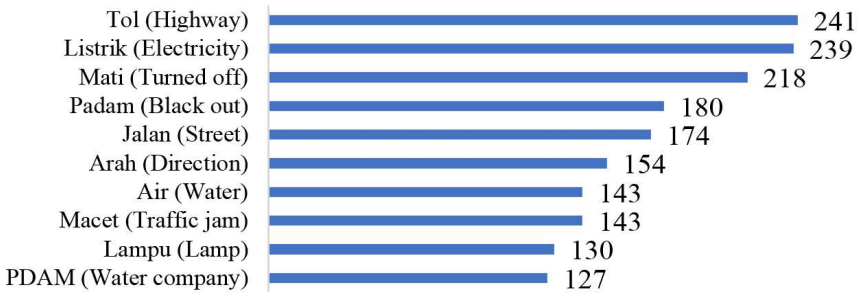


Figure 5: Ten Terms with The Highest Frequency

The performance metric used to evaluate the method was the average of the AUC, the precision, and the sensitivity of the testing data. The result is given in Figure 6. The figure showed that the BNN (297,70,1) was the best BNN model, since the AUC average with 70 hidden nodes was the highest. The average AUC was 68.46%. The positive precision and positive sensitivity for BNN(297,70,1) were, respectively, 29% and 33%, while the negative precision and negative sensitivity were 83% and 80%. The BNN (297,70,1) is a BNN that has 297 input nodes, 70 hidden nodes, and 1 output node.

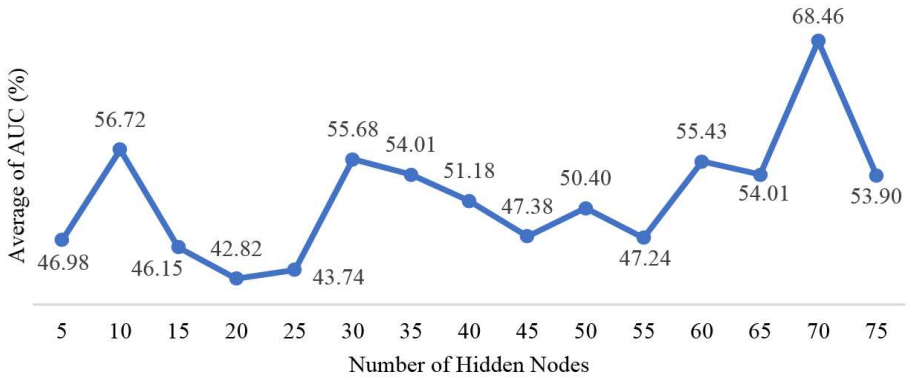


Figure 6: Average AUC for the Testing Data

The BNN result was compared with the CNN result. Before modelling with the CNN, each word was changed into a word vector using the *word2vect* process. The dimensions of the word vectors (size) were 100. This should be considered as a feature extraction process, and it occurs in the convolution and pooling layers. The convolution layer used was three layers, with filter sizes of 300, 150 and 75, respectively. The filter sizes are the number of output filters in the convolution layer. The length of the convolution window was specified to be three. The layer structure of the CNN is given in Table 2.

The average of the AUC value of the CNN for 10-fold testing data was 81%, as shown in Figure 7. This was higher than the average AUC value for the BNN.

The positive precision and positive sensitivity for CNN were, respectively, 68% and 72%, while the negative precision and negative sensitivity were 93% and 91%. The comparison is shown in Table 3. Based on the table, the values of the AUC, precision and sensitivity of the CNN were higher than the values of the BNN. This means that the CNN performed better at classifying the data. In other words, the CNN was able to classify the positive and negative

Table 2: Model Summary of CNN

Layer (type)	Output Shape	Param #
embedding_9 (Embedding)	(None, 2975, 100)	306100
dropout_621 (Dropout)	(None, 2975, 100)	0
conv1d_621 (Conv1D)	(None, 1487, 100)	90300
conv1d_622 (Conv1D)	(None, 743, 100)	135150
conv1d_623 (Conv1D)	(None, 371, 100)	33825
flatten_207 (Flatten)	(None, 27825)	0
dropout_622 (Dropout)	(None, 27825)	0
dense_417 (Dense)	(None, 150)	4173900
dropout_623 (Dropout)	(None, 150)	0
dense_418 (Dense)	(None, 2)	302
Total params : 4,739,577		
Trainable params : 4,433,477		
Non-trainable params : 306,1		

sentiments in this data, although the precision and sensitivity for the negative sentiment were higher than those for the positive sentiment, since the negative sentiment had more data.

Table 3: Performance Metrics of BNN and CNN

Performance Metrics	BNN	CNN
AUC	68.46%	81%
Positive Precision	29%	68%
Positive Sensitivity	33%	72%
Negative Precision	83%	93%
Negative Sensitivity	80%	91%

After classifying the data, the next step was visualization using a word cloud. This was used to understand the content of each sentiment after classification. The results are shown in Figure 8. The size of a word displayed in the word cloud describes the frequency at which that word appeared in the data. In other words, if the word is the most frequently used word in the data, then its size is the biggest.

Figure 8 shows that the most frequent word in the tweets with a negative sentiment was *listrik*(electricity). This was followed by *mati* (turned off), *padam* (black out), *tol* (highway), *air* (water), and *macet* (jammed). On the other hand, the most frequent words in the tweets with a positive sentiment were *jalan* (street), *tol* (highway), *arah* (direction), and *lancar* (swift).

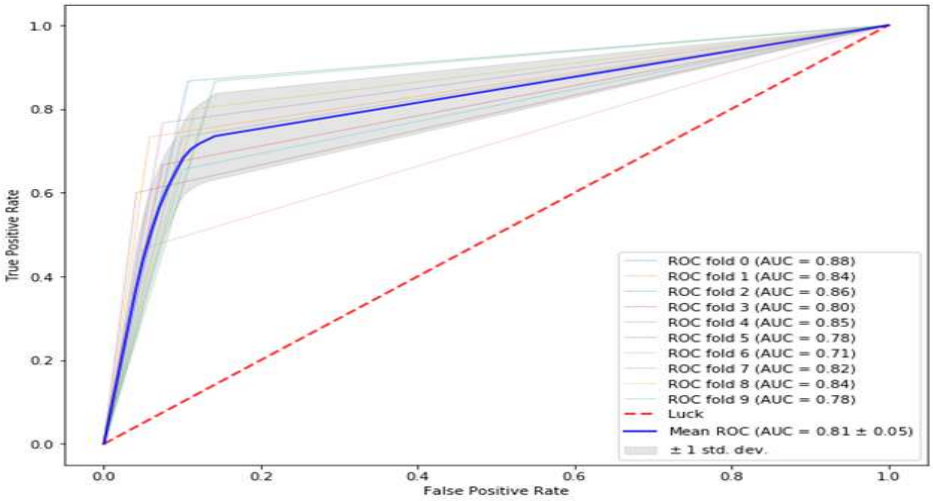
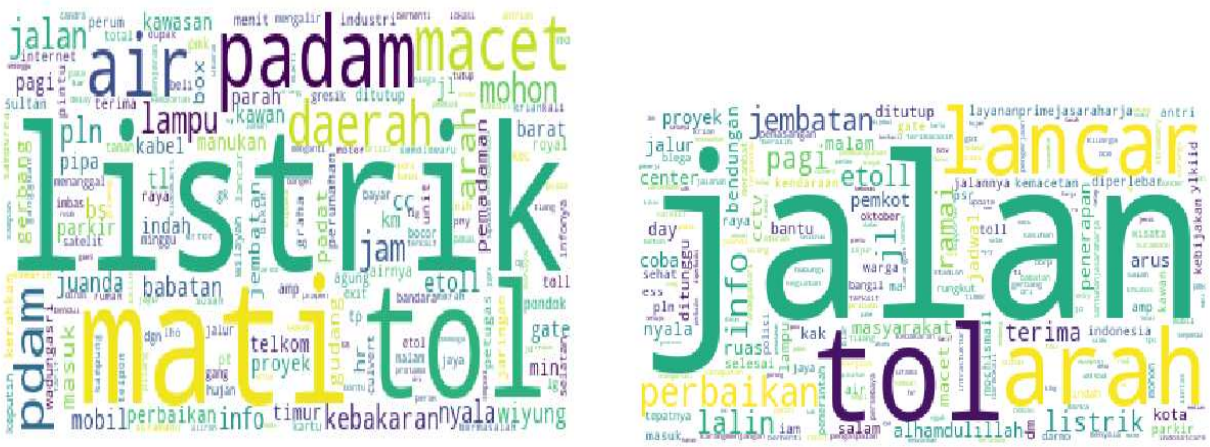


Figure 7: AUC of CNN



(a) Negative Sentiment (b) Positive Sentiment

Figure 8: Visualization of The Data for Each Sentiment

7 Conclusion

In this study, sentiment analysis was performed on Twitter data from the official Twitter accounts of the Surabaya government and a private radio station in Surabaya. Unstructured Twitter data was converted into structured data through pre-processing. The pre-processing steps were removing the URL-link, removing retweets, removing the username, removing numbers, removing punctuation, case folding, removing stop words and tokenizing. Two methods were used to model the sentiment, BNN and CNN. These methods were compared based on the average of the AUC value, precision and sensitivity of 10-fold cross-validation. The result was that the CNN outperformed the BNN since the CNN had the highest values for AUC, precision and sensitivity.

Acknowledgments

This study was funded by a research grant under the scheme of Penelitian Dasar Unggulan Perguruan Tinggi (PDUPT), project No. 907/PKS/ITS/2018. The authors thank *Direktorat Jenderal Penguatan Riset dan Pengembangan, Kementerian, Teknologi, dan Pendidikan Tinggi, Indonesia*, who supported the research.

References

[1] Liu, B. *Sentiment Analysis and Subjectivity in Handbook of Natural Language Processing*. 2nd ed. Boca Raton: CRC Press. 2010. 627–666.

[2] Iriawan, N., K. Fithriasari, B. S. S. Ulama, W. Suryaningtyas, I. Susanto and A. A. Pravitasari. Bayesian Bernoulli Mixture Regression Model for Bidikmisi Scholarship Classification. *Jurnal Ilmu Komputer dan Informasi*. 2018. 11(2): 67–76.

[3] Oktaviana, P. P. and K. Fithriasari. 2018. Analysis of Salmonella sp Bacterial Contamination on Vannamei Shrimp using Binary Logit Model Approach. *Journal of Physics: Conference Series*. 2018. 1008: 012024.1–012024.7.

[4] Manochandar, S. and M. Punniyamoorthy. Scaling Feature Selection Method for Enhancing the Classification Performance of Support Vector Machines in Text Mining. *Computers & Industrial Engineering*. 2018. 124: 139–156.

[5] Lucini, F. R., F. S. Fogliatto, G. J. da Silveira, J. L. Neyeloff, M. J. Anzanello, R. D. S. Kuchenbecker and B. D. Schaan. Text Mining Approach to Predict Hospital Admissions using Early Medical Records from the Emergency Department. *International Journal of Medical Informatics*. 2017. 100: 1–8.

[6] Mohammad, A. H., T. Alwada'n and O. Al-Momani. Arabic Text Categorization Using Support Vector Machine, Naïve Bayes and Neural Network. *Journal on Computing*. 2016. 5(1): 108–115.

[7] Manikandan, R. and R. Sivakumar. Machine Learning Algorithms for Text-Documents Classification: A Review. *International Journal of Development Research*. 2018. 3(2): 384–389.

[8] Kim, Y. Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar. 2014.

[9] Collobert, R. and J. Weston. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. *ICML '08 Proceedings of the 25th International Conference on Machine learning*. Helsinki, Finland. 2008.

[10] Hemalatha, I., G. S. Varma and A. Govardhan. Preprocessing the Informal Text for Efficient Sentiment Analysis. *International Journal of Emerging Trends & Technology in Computer Science*. 2012. 1(2): 58–61.

[11] Rumelhart, D. E., G. E. Hinton and R. J. Williams. Learning Representations by Back-propagating Errors. *Nature*. 1986. 323: 533–536.

[12] Rojas, R. *Neural Network: A Systematic Introduction*. Berlin: Springer. 1996.

[13] Bishop, C. M. *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press. 1994.

[14] Patil, S., A. Gune and M. Nene. Convolutional Neural Networks for Text Categorization with Latent Semantic Analysis. *International Conference on Energy, Communication, Data Analytics, and Soft Computing (ICECDS)*. Chennai. 2017.

[15] Jeni, L. A., J. F. Cohn and F. D. L. Torre. Facing Imbalanced Data Recommendations for the Use of Performance Metrics. *International Conference on Affective Computing and Intelligent Interaction Workshops*. 2013.

[16] Powers, D. M. W. *Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness& Correlation*. Technical Report. School of Informatics and Engineering. Flinders University, Adelaide, South Australia. 2007.

[17] Tala, F. Z. *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*. M.Sc. Thesis. Universiteit van Amsterdam, Amsterdam. 2003.