Opinion Mining on Book Review Using Convolutional Neural Network Algorithm — Long Short Term Memory

Imam Mukhlash, Anshar Zamrudillah Arham, Fakhrur Rozi, Masaomi Kimura, and Dieky Adzkiya

***Abstract*—The quality of a product can be determined by consumer reviews of the product because consumer opinion is in general honest and sentimental to the product. The process of opinion extraction is called opinion mining. This process is done to determine the tendency of the reviewer to the reviewed object. Deep Learning is a recently developed opinion extraction model. This model is widely used for achieving performance in Natural Language Processing. This work classifies the book review on Amazon.com into positive reviews or negative reviews using a combination of Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) algorithms. Simulation results show that the performance of this algorithm is approximately 65.03% for testing data and 99.55% for training data.**

***Index Terms*—Book review, deep learning, long short term memory, opinion mining.**

I. Introduction

The rapid increase of internet users in the world has an impact on the rise of online trading activities that are also popularly called e-commerce. In its use, there are many developments and problems, for example the distance between the seller and the buyer makes the importance of previous consumer testimony to convince potential buyers. Reviews or consumer testimony on a product is a representation of the quality of the product. These reviews are consumer opinions on the quality of the products they buy. However, in practice, it is difficult to digitally distinguish the intentions or trends of opinion. Therefore, it is necessary to do the extraction of the review. The process for extracting useful information from user reviews is called opinion mining. This process is done by using Natural Language Processing method and text analysis method [1]. Opinion is processed to accurately measure the emotions conveyed by consumers in the review, so that information about the quality of the product will be classified as good, neutral and bad.

Recently, there is a rapid development of methods in opinion mining. A recently developed model is emerged from the Naive Bayes method and the Support Vector Machine. Pang and Lee [2] have applied the Deep Learning (DL) model to movie review object. One of the DL model is the Convolutional Neural Network (CNN). In general, CNN is applied to digital image processing for classification and clustering. Jin Wang *et al.* [3] proposed a combination of Regional CNN and LSTM tested on dimensional sentiment analysis. While Long Chen, Yuhang He and Lei Fan work on the description of car images using the LSTM algorithm for classification [4]. Thereafter, in 2014 Kim conveyed his innovation in the application of the CNN model to NLP, especially in the classification of sentences [5]. Kim proposed a new concept in the use of CNN on text processing which shows that CNN is an excellent method of processing text.

Therefore, in this paper, a combination of CNN and LSTM methods is used in the classification of an opinion. This method produces the class type of an opinion into two main classes: positive and negative. The CNN method is the same as Kim's research [5] but the activation function is the same as in Long Chen [6] and Jin Wang [4]. This method is applied on book review data obtained from Amazon.com's online site. Fakhrur Rozi *et al.* [6] discussed opinion mining using CNN-L2-SVM algorithm. Taspinar described about sentiment analysis with bag-of-words [7].

II. Models and Preliminaries

In this section, we discuss the convolutional neural network and long short term memory.

A. Convolutional Neural Network

Convolutional Neural Network (CNN) is the development of Multilayer Perceptron (MLP) designed to process two-dimensional data. CNN is included in the type of Deep Neural Network because of the high depth of the network and applied to many image data. The CNN algorithm has some similarities to MLP, but in CNN each neuron is presented in a two-dimensional form, unlike the MLP that each neuron is of one dimension.

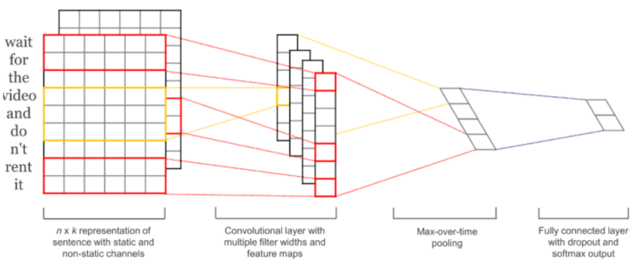

Fig. 1. Architecture of the model containing two channels for a word [5].

Manuscript received July 19, 2018; revised August 27, 2018. This work was supported in part by Program Bantuan Seminar Luar Negeri Ditjen Penguatan Riset dan Pengembangan, Kemenristekdikti.

I. Mukhlash, A. Z. Arham, F. Rozi and D. Adzkiya are with the Department of Mathematics, Institut Teknologi Sepuluh Nopember, Kampus ITS Sukolilo-Surabaya 60111, Surabaya, Indonesia (e-mail: imamm@matematika.its.ac.id).

M. Kimura is with Department of Information Science and Engineering, School of Engineering, Shibaura Institute of Technology, 3-7-5, Toyosu, Koto City, Tokyo 135-8548, Japan (e-mail: masaomi@shibaura-it.ac.jp).

doi: 10.18178/ijmlc.2018.8.5.725

There are two main layers in the structure of the CNN algorithm: convolution layer and pooling layer. In the sentiment analysis, we simply use the two main layers illustrated in Fig. 1.

1) Convolution layer

The first layer in the CNN-LSTM network architecture performs a convolution process on all vectors of words present in the review. A convolution process is performed to get a feature map of the initial input data. Convolution layer consists of neurons which is arranged in such a way that it forms a filter of a certain length and height. For example, we use the corpus "He is not lazy. He is intelligent. He is smart" that can be seen in Fig. 2. The input data is 6×6 and has 4 filters. These four filters will be shifted to all parts of the input data and then dot product operation between input and filter values at each shift so that the result of the operation will be operated using activation function. The output of the activation function is a feature map. There are no specific rules in filter and bias value selection. These values used in the example are commonly used in previous research.

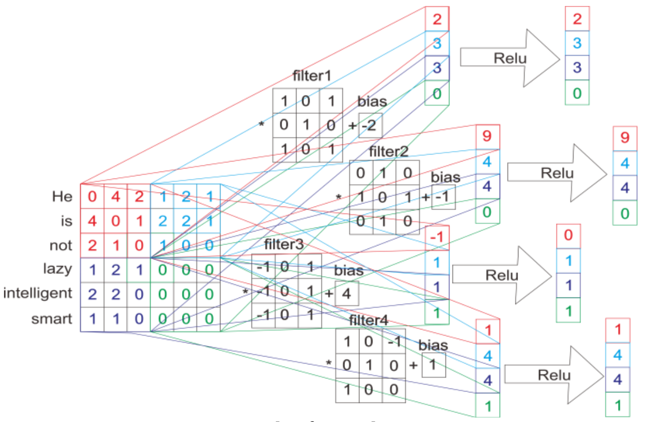

Fig. 2. Example of convolution process.

Assume a review phrase has a length n of word vectors having a dimension of 50 as the input of the model. For example, let $x_i \in \mathbb{R}^{50}$ be the word vector of the i-th index. When word vectors are combined, then reviews can be represented as follows:

$$x_{1:n} = x_1 \oplus x_2 \oplus x_3 \oplus \dots \oplus x_n \quad (1)$$

where $\oplus$ is the join operator. In general, $x_{i:i+j}$ represents the combined result of the word vectors from index *i* to index *i*+*j* as in the following equation:

$$x_i, x_{i+1}, x_{i+2}, \dots, x_{i+j} \quad (2)$$

Convolution operation uses a filter or parameter $w \in \mathbb{R}^{50h}$ where *h* represents the window size. Then, we use the following equation to obtain the feature value:

$$c_i = f(net) \quad (3)$$

$$net = w \cdot x_{i:i+h-1} + b \quad (4)$$

where $b \in \mathbb{R}$ is the bias parameter and $c_i \in \mathbb{R}$ is the feature value at index *i* and *f* is a nonlinear function. In this work, we use the Rectified Linear Unit (ReLu) as the nonlinear function. This function generates a positive upper bound, that can be written as follows:

$$ReLu(x) = \max(0, x) \quad (5)$$

Thus the equation becomes

$$c_i = ReLu(net) \quad (6)$$

$$net = w \cdot x_{i:i+h-1} + b \quad (7)$$

Filter *w* is applied to every window of possible words in reviewed sentences $\{x_{1:h}, x_{2:h}, x_{3:h}, \dots, x_{n-h+1:n}\}$ until we obtain a feature map as follows:

$$c = [c_1, c_2, c_3, \dots, x_{n-h+1}] \quad (8)$$

where $c \in \mathbb{R}^{n-h+1}$. In order to obtain a maximal result, in this work, we use more than one filter and more than one window word.

2) Pooling layer

In this layer, we perform pooling operations on each feature map that has been generated from the convolution layer. It will capture the maximum value of each feature map to select the values that are important to the feature map. Here is the pooling operation formula:

$$\hat{c} = \max\{c\} \quad (9)$$

where $\hat{c}$ is the maximal value of feature map *c* corresponding to a filter. Because there are *m* filters, the result of pooling layer is a vector of maximal values for each feature map. The vector contains *m* entries. Thus, we obtain:

$$z = [\hat{c}1, \hat{c}2, \hat{c}3, \hat{c}m] \quad (10)$$

where *z* is the resulting vector from pooling layer that will be used in the subsequent step.

B. Long Short Term Memory

Long Short Term Memory (LSTM) is a special type of Recurrent Neural Network (RNN) algorithm that is capable of studying long-term dependency. Simply put, remembering long-term information is LSTM's innate behavior. LSTM also has a chain structure like the RNN structure, the difference lies in the structure of the repetition module [8].

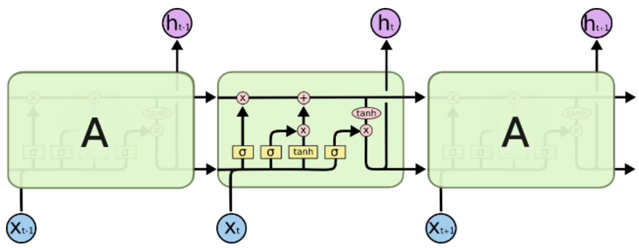

Fig. 3. LSTM Repetition Module [8].

In Fig. 3, each line carries all the vectors from the output of a node to the other input of a node. The key of LSTM is its cell state. The horizontal line is passing through the top of the diagram. The analogy of cell state is like a conveyor belt. It runs across the chain with just a little linear interaction. It is easy for information to flow without change.

We will describe the procedure in LSTM.

1) Determine the information removed from cell state

The focus is on the sigmoid layer or also known as forget gate. This layer selects the information that will be removed. Forget gate produces 0 and 1 for cell state C_{t-1}. Number 1 represents keeping the memory, whereas number 0 represents forgetting the memory:

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \quad (11)$$

2) Determine the information stored in the cell state

In this step, there are two layers: sigmoid layer and tanh layer. Sigmoid layer is an input layer which decides the value that will be updated. Then tanh layer constructs vector of values in the new candidate $\tilde{C}_t$ which will be added to the cell state. Then the old cell state C_{t-1} is updated into the new cell state C_t.

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \quad (12)$$

$$\tilde{C}_t = tanh\,(W_C.[h_{t-1}, x_t] + b_C) \quad (13)$$

3) Add new information

Then we execute the decision that has been taken in the previous step. Equation (14) represents the mechanism of updating the information or new memory:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (14)$$

4) Compute the output

We will execute the sigmoid layer that decides which part of the cell state becomes the output. Then fed the cell state through tanh (to force the value to be between -1 and 1) and multiply it with the result from sigmoid layer so that we just produce the needed part:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (15)$$

$$h_t = o_t * \tanh\,(C_t) \quad (16)$$

C. The Proposed Algorithm

The algorithm for opinion mining in this work is a combination of CNN and LSTM. The input of this algorithm is a vector of reviews. Then we compute the feature map. After that, we compute the maximal value of feature map. Finally, we use the LSTM algorithm described in the previous section. In CNN-LSTM, there are three main steps, namely preprocessing, feature extraction and data mining process. The preprocessing step consists of two parts, i.e. tokenizing and filtering. These processes are described in Fig. 4.

Next the preprocessing stage is described in detail. Tokenization is a process to separate words in every sentence of a review. Separated words are called tokens. Tokenization is done to facilitate the observation of the meaning of each word that influences in determining whether the review is positive or negative. The stage after tokenization is filtering. At this stage, we filter important words from the review data, in other words discarding words that have no effect on the main meaning of the sentence or paragraph of the review. Deleted words include, for example, hyphens, affixes and others. In addition, case folding is also done or the conversion of capital letters into small letters for the system to process data more efficiently and effectively.

In the second step, we execute the process of feature extraction. The process is as follows: when we already have a stopwords-free array of words, then we will change it into a vector form so that it can be digested by a computer. The word vector is a representation of the closeness between words. The vector to be used is word2vec which is a Glove vector. For words that are not contained in the Glove vector, the word vector will be randomly generated. Through feature extraction we obtain word matrix representation of the review sentence from the stopwords-free array of word reviews. A $n \times 50$ word representation matrices will be constructed where n is the word length in the review and 50 is the word vector length.

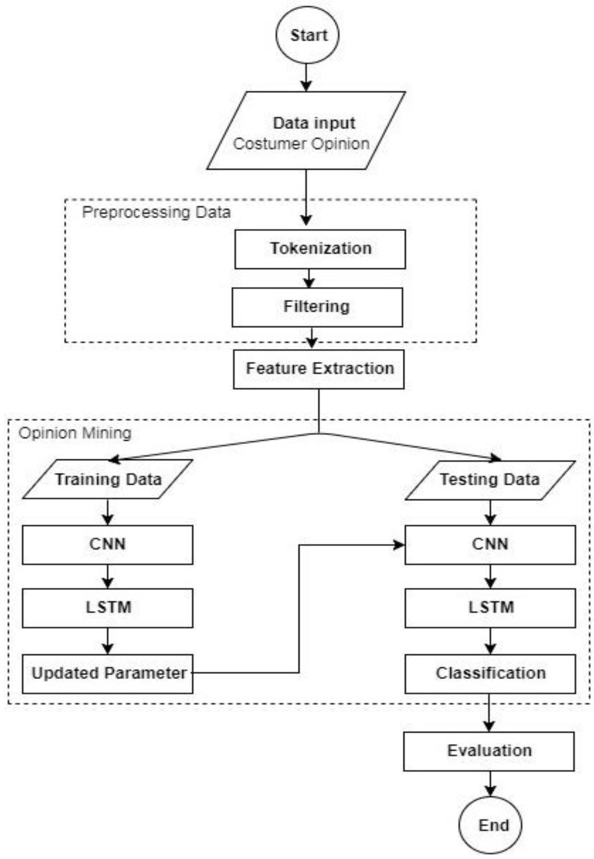

Fig. 4. Flowchart of the opinion mining process.

The architecture of CNN consists of four main layers: input layer, convolutional layer, pooling layer and output layer. The input is a review matrix and two-dimensional class vector. The first layer is the input layer that contains a matrix which is a combination of word vector representation in one review. The dimension of word vector used in this work is 50 so that the size of matrix is $k \times 50$ where k is the number of words in one review. In the convolution layer, we construct feature map vector. The number of feature map vectors equals the number of filters used in the convolution process. Then in the pooling layer, we take the best feature map value from each layer until we obtain the most important feature value from the review. The vector will be used in the output layer with a fully connected network. *Long Short Term Memory* (LSTM) is used to obtain the score of each class so that each class can be classified into positive or negative category.

III. Results and Discussions

The learning process produces a model that adjusts the input data. We need to check the accuracy of the model generated from the learning process.

The testing method in this work is using cross validation approach. This method is done by dividing the data into 10 equally sized parts. In the first iteration, the model will be tested against the first part out of 10 parts on the data and the remaining data are used in the learning process.

In the second iteration, the test data is the second part of the 10 parts of data. The third part of the data is test data for the third iteration, and so on. The accuracy of the test data from each iteration will be averaged to determine the performance of the model.

The model that has been created is necessary to evaluate in order to be able to see its performance in performing the tasks ordered. There are several ways of evaluation used in the model. In this paper, we use accuracy, precision, recall and F1 to evaluate the performance of the classifier.

Prior to the evaluation phase of the system, the results of the classification process are categorized into four types, namely true positive (TP), true negative (TN), false positive (FP), false negative (FN). TP or true positive is the amount of positive data that is classified into positive class. TN or true negative is the amount of negative data that is classified into negative class. FP or false positive is the amount of positive data classified to the negative class. FN or false negative is the amount of negative data classified to the positive class. The accuracy of classification is calculated by the following formula:

$$\text{accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{17}$$

Precision is a measure of the accuracy between the information requested by the user and the answers given by the system. The difference between accuracy and precision is, the accuracy shows the proximity of the measured result to the true value, precision indicating how close the difference in value when the repetition of the measurement takes place.

$$\text{precision} = \frac{TP}{TP+FP} \tag{18}$$

Recall is the success rate of the classifier in rediscovering an information, that formulated by

$$\text{recall} = \frac{TP}{TP+FN} \tag{19}$$

While $F1$ can be interpreted as an average of precision and recall, that is formulated

$$F1 = 2 \times \frac{(\text{precision} \ \times \text{recall})}{(\text{precision} \ + \text{recall})} \tag{20}$$

The learning process produces a model or classifier that has adjusted the input data. The model generated from the learning process needs to be tested for its accuracy in order to know whether the resulting model is the right model. The first test was conducted on training data. The model produced by the train data in this study has a very high accuracy, as shown in Fig. 5. The accuracy of the training data is 99.55% after 20 iterations and the graph is always increasing for each iteration.

There are not so many high changes in the results of the testing process. When processing the test data, the model produces a final accuracy of 64.01%, which can be seen on the accuracy of the test graph that the accuracy for the test data reached the highest accuracy of 66.98% and lowest accuracy 63.88% and ended in 64.01%. In other word, the average of accuration is 65.03. The graph of the movement of model accuracy to the testing data is quite stable. The graph in Fig. 6 shows the model's accuracy movement for the testing data.

The difference of accuracy graph between the training data and the testing data is due to the supervision on the accuracy of classification on each testing using the training data, the accuracy of training data will always increase or remain on each iteration. In contrast to testing data, model testing using testing data is not supervised.

The comparison between the movement of the learning result and the test results looks very much different. This shows the model is overfitting. Overfitting is an event where the model is too good at classifying training data but very bad in the classification of test data. This is a common phenomenon in classification using artificial neural networks.

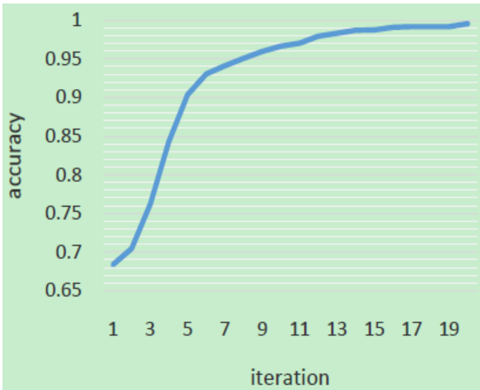

Fig. 5. Accuracy graph of training data.

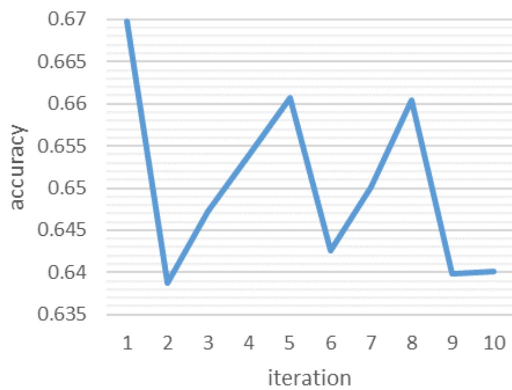

Fig. 6. Accuracy graph of testing data.

In the research of Rozi *et al.* [6] that use a combination of methods of Convolutional Neural Network with Support Vector Machine to classify the same data. The result obtained a smaller error than those who do not use Support Vector Machine with a result of 64.6% for testing data and 83.23% for training data.

In the related research, Kim's study [5] used the Convolutional Neural Network model for the classification of sentence-shaped opinion. Kim did some ways to evaluate model, the results were different based on the data used. The result from SST-1 data only got 48% result while for others reach more than 80%.

In the research of Jin Wang *et al.* [3] used a combination of CNN-LSTM model for dimensional sentiment analysis. The

Root Mean Square Error (RMS) value is 1.341, Mean Absolute Error (MAE) value is 0.987, Pearson Correlation Coefficient (r) value is 0.778.

There is also research that has been done on Amazon.com's book review data by Taspinar. Researchers performed sentiment analysis using SVM with Bag-of-Word (BOW) as word vector constructs. The result obtained an accuracy of 60%. So, our method outperforms the methods are used by Rozi *et al*. [6] and Taspinar [7].

There are several possible ways to improve the performance of this algorithm: adding the amount of input data to improve performance and avoid overfitting, exploring the depth of the network and type of classification, and using better word representation vectors.

IV. Conclusions

In this paper, we have explored opinion mining using CNN-LSTM algorithm. The input of this algorithm is a vector of reviews. Then we compute the feature map. After that, we compute the maximal value of feature map. Finally, we use the LSTM algorithm described in the previous section. According to our experimental results, the CNN-LSTM model can decide the sentiment of book review. Furthermore, the performance of the algorithm is 65.03% for the testing data and 99.55% for training data, which are quite satisfactory. Opinion mining on book review using CNN-LSTM algorithm resulted in good performance. It takes a larger amount of learning data to achieve better performance. The word representation vector is helpful in determining the sentiments of the review.

References

[1] F. Ali, K. S. Kwak, and Y. G. Kim, "Opinion mining based on fuzzy domain ontology and support vector machine: A proposal to automate online review classification," *Applied Soft Computing*, vol. 47, pp. 235-250, 2016.

[2] Singh and S. K. Dubey, "Opinion mining and analysis: A literature review," in *Proc. 5th International Conference on the Confluence the Next Generation Information Technology (Confluence)*, 2014, pp. 232-239.

[3] J. Wang, L. C. Yu, K. R. Lai, and X. Zhang, "Dimensional sentiment analysis using a regional CNN-LSTM model," in *Proc. the 54th Annual Meeting of the Association for Computational Linguistics*, 2016, vol. 2, pp. 225-230.

[4] L. Chen, Y. He, and L. Fan, "Let the robot tell: Describe car image with natural language via LSTM," *Pattern Recognition Letters*, vol. 98, pp. 75-82, 2017.

[5] Y. Kim, "Convolutional neural networks for sentence classification," arXiv:1408.5882, 2014.

[6] M. F. Rozi, I. Mukhlash, Soetrisno, and M. Kimura, "Opinion mining on book review using CNN-L2-SVM algorithm," *Journal of Physics: Conference Series*, vol. 974, no. 1, p. 012004, 2018.

[7] T. Ahmet. Sentiment analysis wih bag-of-words. [Online]. Available: http://ataspinar.com/2016/01/21/-sentment-analysis-with-bag-of-words

[8] O. Christopher. (2015). *Understanding LSTM Networks*. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs

Imam Mukhlash is a lecturer in the Department of Mathematics, Faculty of Mathematics, Computing, and Data Sciences, Institut Teknologi Sepuluh Nopember, Indonesia. He received his B.Sc. in mathematics from Institut Teknologi Sepuluh Nopember, a master degree and a doctoral degree both in informatics from Institut Teknologi Bandung, Indonesia. His research interests include data mining, fuzzy logic, and intelligent systems. He has been awarded some grant research projects such as from Institute of Research and Public Services, Institut Teknologi Sepuluh Nopember and National Research Program, the Ministry of Higher Education of Indonesia. He has also published many papers in international journal and conferences.

Fakhrur Rozi is a software engineer in a software company. He received his B.Sc. in mathematics from Institut Teknologi Sepuluh Nopember, Indonesia in 2017. In 2017, he participated in a student exchange program at the Department of Information Science and Engineering, Shibaura Institute of Technology, Japan.

Masaomi Kimura received a B.E. degree in precision mechanics in 1994, M.S. degree in 1996, and Ph.D. degree in 1999 in physics from The University of Tokyo. He is a professor in the Department of Information Science and Engineering, Shibaura Institute of Technology. His current interests include data engineering, artificial intelligence, and medical safety.

Dieky Adzkiya received the B.Sc. degree in September 2005 and the M.Sc. degree in October 2008, both in mathematics from the Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. He received the Ph.D. degree in systems and control in October 2014 and after that he continued as a postdoctoral researcher until June 2015, both at the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands. Currently, he is a lecturer in the Department of Mathematics at Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. His research interests are in the analysis and verification of max-plus-linear systems and in their applications.