computing@computingonline.net
www.computingonline.net

Print ISSN 1727-6209
On-line ISSN 2312-5381
International Journal of Computing

HUM-TO-CHORD CONVERSION USING CHROMA FEATURES AND HIDDEN MARKOV MODEL

Hariyanto [1], Suyanto [2]

[1] School of Computing, Telkom University, Bandung, Indonesia, e-mail: harydj@students.telkomuniversity.ac.id
[2] School of Computing, Telkom University, Jl. Telekomunikasi Terusan Buah Batu, Bandung, Indonesia,
e-mail: suyanto@telkomuniversity.ac.id

Paper history:
Received 23 February 2019
Received in revised form 9 May 2020
Accepted 17 October 2020
Available online 30 December 2020

Keywords:
chroma features;
Hidden Markov Model;
hum-to-chord conversion;
music.

Abstract: Music is basically a sound arranged in such a way to produce a harmonious and rhythmic sound. The basis of music is a tone, which is a natural sound and has different frequencies for each sound. Each constant sound represents a tone. The tones can also be represented in a chord. Humans are capable of creating a sound or imitating a tone from other human beings, but they are naturally unable to represent them into musical notation without musical instruments. This research addresses a model of Hum-to-Chord (H2C) conversion using a Chroma Feature (CF) to extract the characteristics and a Hidden Markov Model (HMM) to classify them. A 10-fold cross-validating shows that the best model is represented by the chroma coefficients of 55 and HMM with a codebook of 16, which gives an average accuracy of 94.83%. Examining on a 30% testing set proves that the best model has a high accuracy of up to 97.78%. Most errors come from the chords with both high and low octaves since they are unstable. Compared to a similar model called musical note classification (MNC), the proposed H2C model performs better in terms of both accuracy and complexity.

1. INTRODUCTION

Many musicians can make a song from a hum, but they have a difficulty to directly translate the tone into the chord because of lack of experience in playing musical instruments. A novice musician needs a musical instrument to match a hum with that tone or with the help of other musicians who have cognitive knowledge of music. To address this problem, many experts do researches on the hum, such as in [1, 2]. Since 2010, the query-by-humming has been a part of the area of research called Singing Information Processing [3-5].

This paper focuses on developing a model to recognize a hum and convert it into a chord automatically. This model is built using the chroma features (CF) method to extract some features of the input data in the form of a hum that produces 12-dimensional vectors representing diatonic tones. The chroma used in this research is a chroma discrete cosine transform-reduced log pitch (CRP), which is a variant of CF that is capable of handling the timbre changes [6-8]. The CF method is used as a feature extraction since it produces the form of pitch information needed to recognize a chord [9].

The extracted features are then classified using a classification model. Many classifiers can be used for this task, such as Artificial Neural Networks (ANN) [10], Convolutional Neural Networks (CNN) [11], Recurrent Neural Networks (RNN) [12], Deep Learning [9], and HMM [13].

In this paper, HMM is selected since it is well known as an excellent classifier for recognizing a data sequence without being affected by the variances of the sequence length [13]. It is also capable of predicting the next frame of data based on the previously recognized frames. HMM is commonly used for many applications with an input of data sequences, such as automatic speech recognition [14-16], video processing [17, 18], gene sequence classification [19], chord recognition [20], and music analysis [21].

2. HUM-TO-CHORD CONVERSION

The model of Hum-to-Chord Conversion is illustrated in Fig. 1. It consists of three sub-processes, i.e., Preprocessing, Feature Extraction, and HMM-Based Classifier.

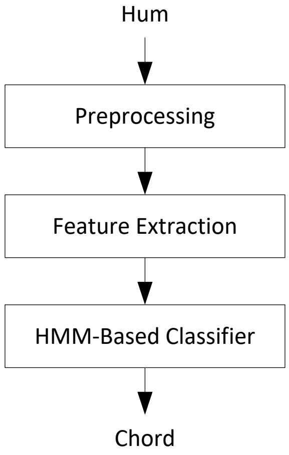

Figure 1 – Block diagram of H2C

2.1 DATASET

The dataset used here is a set of audio .wav files containing hums of male and female with a distance of E2 to C6. It is collected by recording the hum using a smartphone. It consists of four types of soprano, alto, tenor, and bass from three singers each. All singers have cognitive knowledge of music. The process of recording is accompanied by a qualified choir trainer so that the accuracy of the hum tone can be corrected directly.

A singer sings the tone with a hum that consists of twelve diatonic tones with octave distance according to the capacity of the singer, where each tone is repeated five times. Hence, each singer produces 12×2×5=120 hums. Those recorded hums are then manually selected by the choir trainer to get the 100 best ones. Thus, the total number of recorded hums is 4×3×100=1,200, as listed in Table 1.

Table 1. Dataset of hum

Type	#Singers	#Hums
Sopran	3	300
Alto	3	300
Tenor	3	300
Bass	3	300
Total		**1200**

2.2 PREPROCESSING

There are two steps in the preprocessing. Firstly, a recorded hum of .m4a format is converted into a .wav format. Secondly, the stereo .wav file is then converted into a mono channel as needed by the next process of feature extraction.

2.3 FEATURE EXTRACTION

Chroma features can be used to estimate a pitch of signal by framing the signal. A frame is then transformed using a Discrete Cosine Transform (DCT) to get its spectral energy distributed in 12 bins that represent 12 diatonic tones. Fig. 2 illustrates the block diagram of chroma feature extraction. The output of the feature extraction is a chromagram, which is a spectrogram that shows the intensity of each bin in the time domain. The values of each bin are then added up to get a chroma pitch [8]. In this paper, a chroma pitch is a column vector with a size of 12×1.

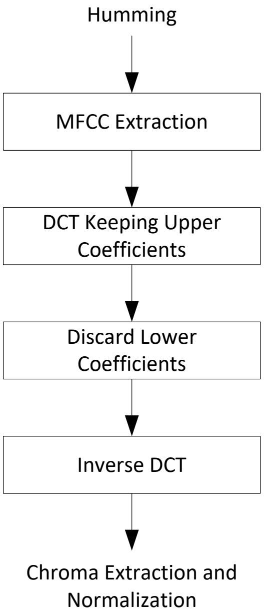
Humming
MFCC Extraction
DCT Keeping Upper Coefficients
Discard Lower Coefficients
Inverse DCT
Chroma Extraction and Normalization

Figure 2 – Chroma feature extraction

Each person has a unique timbre. Hence, the feature extraction method used here is the CRP that is capable of handling the varying timbres [7]. The CRP works by extracting the signal using a Mel Frequency Cepstral Coefficients (MFCC). The extracted signal is then transformed using a DCT to produce 120 pitch logarithmic vectors that produce 120 coefficients. The top 120 coefficients are then selected (*n*: 120), and the lowest coefficients are discarded to solve the problem of timbre variances [7] and [8]. An inverse DCT is then performed to return the signal into the time domain. Finally, the Chroma extraction process is carried out.

A chroma representation is illustrated in Fig. 3, which is adopted from a paper described in [6]. An example of a chromagram is illustrated in Fig. 4. It is generated by the stable energy of chroma bin.

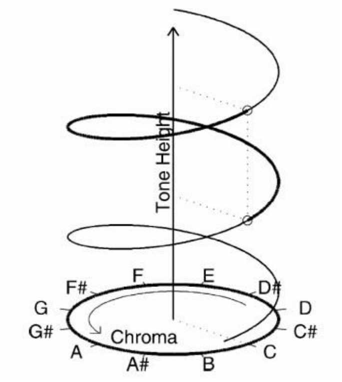

Figure 3 – Chroma representation, adopted from [6]

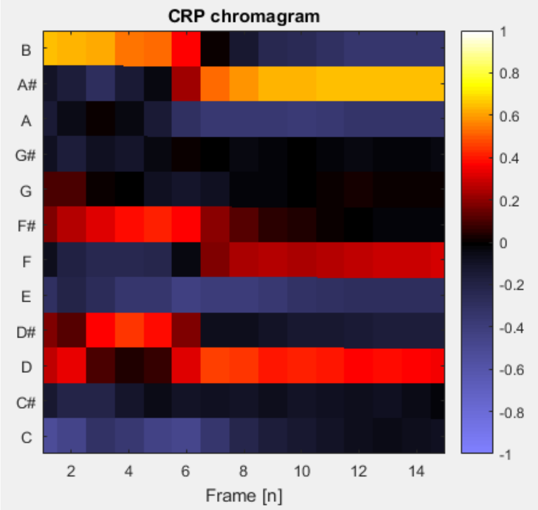

Figure 4 – Example of a chromagram

2.4 HIDDEN MARKOV MODEL

The HMM is designed to have twelve states that represent twelve tones in one octave, as in Fig. 5.

C	C#	D	D#	E	F	F#	G	G#	A	A#	B

Figure 5 – List of tones used as the HMM states

First, the extracted features should be discretized using a codebook with a particular size based on the length of the feature vector. The discretized features are then fed to the HMM. The classification is performed by calculating the probability of the chroma vector O to the tones to get 12 values of probabilities, where the normalized values must be 1, which is formulated as

$$\sum_{i=1}^{12} P(O\,|\,C_i)=1\cdot \qquad (1)$$

The trained HMM models are used to calculate the probabilities of converting a tone into a chord. The testing data is classified by maximizing the likelihood between the model and the input tone.

3. RESULT AND DISCUSSION

In order to get the highest accuracy, the proposed model is evaluated using two scenarios: 1) observation of the effects of chroma parameters to the model accuracy; and 2) observation of model performance using 10-fold cross-validation. The CRP coefficients used here are 55 that refers to the result in [6], which proves that those CRP coefficients reach the highest accuracy. Those CRP coefficients of 55 can also be used for hum since it has similar characteristics with music and other instruments for chord extraction and analysis.

3.1 SCENARIO 1

The dataset used in this scenario is a train-set of 840 (70%) hums, and a test-set of 360 (30%) hums with a codebook size of 16. The experimental result shows that the CRP parameters give a much higher accuracy (up to 95.83%) than the default parameters (90.00%). It is achieved using the CRP coefficient of 55. This result proves that the log compression in CRP makes the signal stable, and the DCT is capable of distinguishing the unique timbres.

3.2 SCENARIO 2

The H2C model is then evaluated using *k*-fold cross-validation. In this research, *k* = 10 since it is recommended for many model-validation methods [22]. The dataset of 1,200 hums is divided into ten folds, where each fold consists of 120 hums. Hence, there are ten experiments conducted. In each experiment, nine folds are fed to train a model, and one other fold is used to validate the model.

The experimental results of 10-fold cross-validation using the CRP coefficient of 55 and varying codebook sizes are illustrated in Fig. 6. The codebook size of 256 gives the lowest averaged accuracy of 23.91%, and the codebook size of 16 gives the highest one up to 94.83%. A bigger codebook does not always give higher accuracy. A too-large codebook makes optimizing cluster does not converge to an optimum point as there are so many centroids. Hence, the codebook size of 16 is the optimum parameter that gives the stable accuracies for the ten folds.

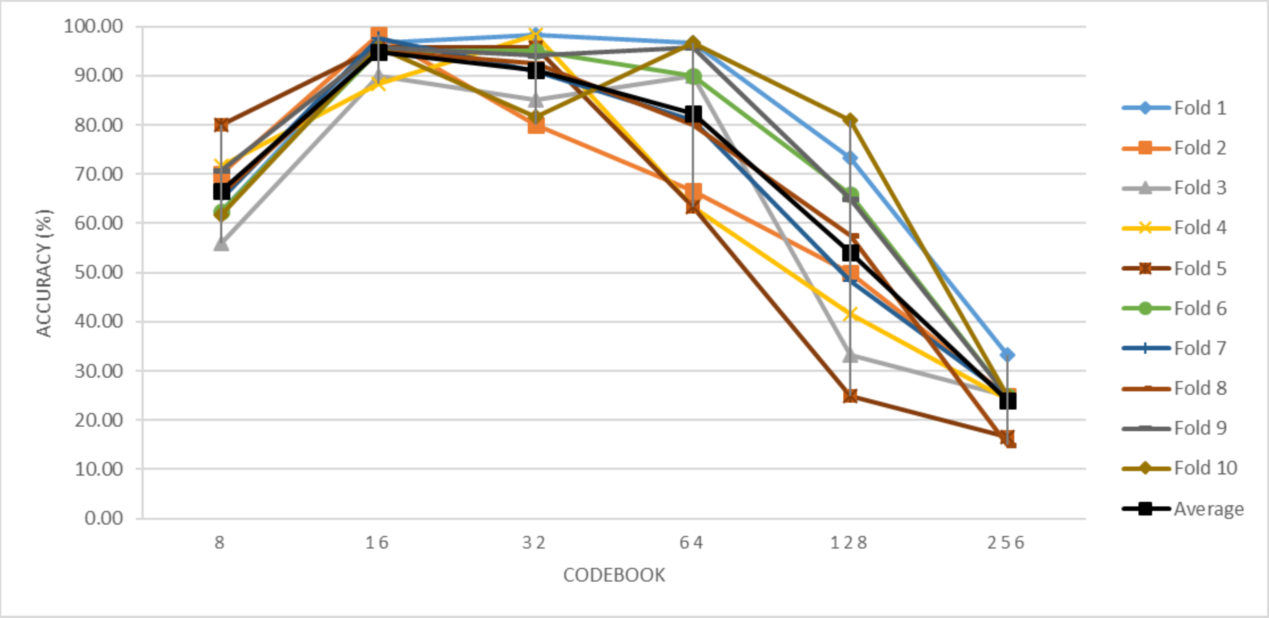

Figure 6 – Accuracy of H2C using 10-fold cross-validation

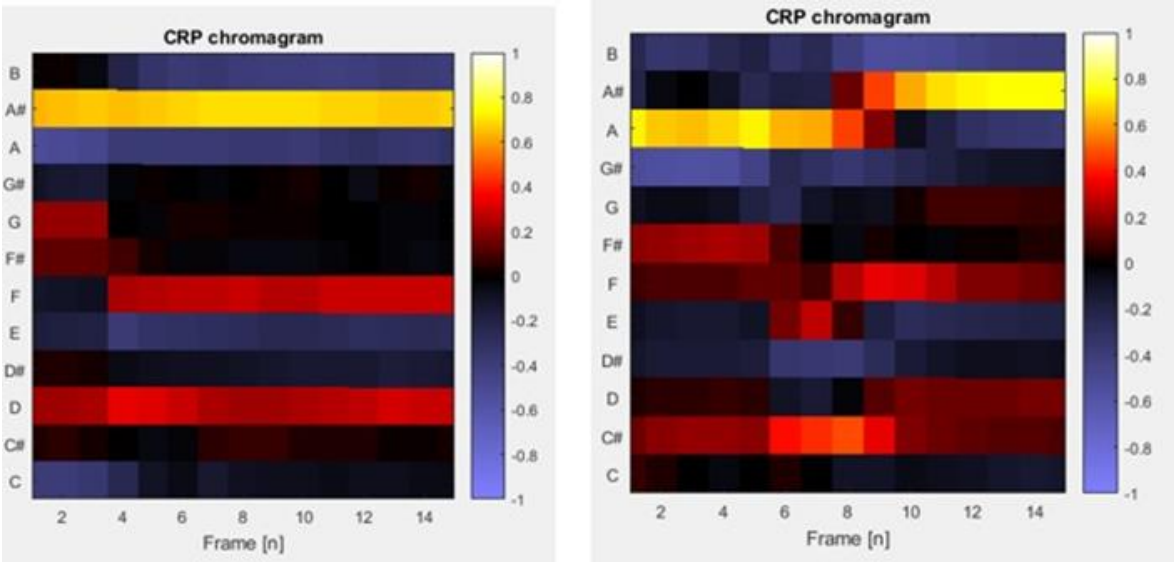

Figure 7 – Chromagrams of chord A generated by hums with stable energy (left) and unstable energy (right)

Furthermore, the proposed H2C model is then compared to the musical note classification (MNC) model described in [10], which uses a harmonic product spectrum (HPS) as the feature extraction and ANN as the classifier. Both H2C and MNC have the same task, which is to classify the sounds in the dataset into 12 chords. However, there are two differences between H2C and MNC models related to the input and the number of datasets. H2C receives a hum as the input while MNC accepts the sound of electric guitar and other instruments. H2C is trained using 840 data (70 data in each chord) to generalize 360 unseen data into 12 chords (30 data each) in the testing set. Meanwhile, MNC is trained on 2,400 data (200 data in each chord) to classify 2,400 unseen data into 12 chords (200 data each) in the testing set. It seems that the testing set for MNC is more challenging than that in H2C.

The results are illustrated in Table 2. H2C model generally performs better than MNC. It gives perfect accuracies of 100% for ten chords and suffers for two chords: F and A# with an accuracy of 83.33% and 90%, respectively. The errors occur since those chords are unstable that have both high and low octaves for all singers, as illustrated in Fig. 7 that shows the chromagrams of chord A generated from two different hums. The figure on the left shows a chromagram with stable energy of chroma bin that gives a true conversion. In contrast, the figure on the right shows a chromagram with unstable energy of chroma bin and, as a consequence, gives a false conversion.

Table 2. Classification accuracy (%) produced by both H2C and MNC models

Music Key	H2C	MNC [10]	
	Hum	Electric Guitar	Other Instruments
C	100.00	93.60	96.00
C#	100.00	95.80	92.00
D	100.00	97.90	92.00
D#	100.00	100.00	92.00
E	100.00	97.90	96.00
F	83.33	100.00	96.00
F#	100.00	100.00	96.00
G	100.00	100.00	100.00
G#	100.00	91.30	92.00
A	100.00	97.90	84.00
A#	90.00	97.90	96.00
B	100.00	97.90	92.00
Average	**97.78**	97.50	93.60

In contrast, MNC that receives electric guitar produces the perfect accuracies of 100% only for four chords: D#, F, F#, and G. It gives low accuracies of 91.30% and 93.60% for both chord G# and C, respectively. Meanwhile, MNC that accepts other instruments gives the perfect accuracies of 100% only for one chord (G) and obtains low accuracies ranging from 84% to 96% for the other 11 chords.

Overall, H2C produces an averaged accuracy of 97.78% that is slightly higher than MNC with the electric guitar that gives a mean accuracy of 97.50%. This result seems to indicate that the performance of H2C is similar to MNC. However, in practice, the electric guitar sounds generated by some guitarists are commonly more stable than the hums produced by several singers. It means that the conversion of electric guitar to chord is easier than hum. It is proved by the result of MNC for other instruments that yields a much lower averaged accuracy of 93.60%.

Hence, it can be implicitly said that the combined CRP and HMM is better than both HPS and ANN in converting a hum or music into a chord. Firstly, CRP offers much lower complexity (55 feature elements) than HPS (181 feature elements). Secondly, HMM is easier trained using some codebook sizes to get the optimum structure than ANN that should be trained on a manually-designed structure, which can be not optimum.

4. CONCLUSION

The 10-fold cross-validating on the developed H2C model shows that the best model is achieved using a chroma coefficient of 55 and an HMM with a codebook of 16, which gives an average accuracy of 94.83%. Testing on a more challenging dataset (where the testing set is 30% of the total data) proves that the best model is capable of converting hums into chords with high accuracy of 97.78%. The feature extraction using CRP parameters gives significantly higher accuracy than the default parameters. The proposed H2C also performs better than MNC in terms of accuracy as well as complexity. Most errors come from the chords that have both high and low octaves because of their inflections or instability of sounds.

ACKNOWLEDGMENT

We would like to thank all students who are singers of Telkom University Choir for contributing the dataset of hums.

5. REFERENCES

[1] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith, “Query by Humming: Musical information retrieval in an audio database,” *Proceedings of the Third ACM International Conference on Multimedia*, 1995, pp. 231–236.

[2] M. Ryynänen, A. Klapuri, “Query by humming of MIDI and audio using locality sensitive hashing,” *Proceedings of the IEEE International Conference on Audio, Speech and Signal Processing*, Las Vegas, USA, 2008, pp. 2249–2252.

[3] M. Goto, T. Saitou, T. Nakano, and H. Fujihara, “Singing information processing based on singing voice modeling,” *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 5506–5509.

[4] M. Goto, “Singing Information Processing,” *Proceedings of the 12th International Conference on Signal Processing*, 2014, pp. 2431–2438.

[5] E. M. Martínez, “Singing Information Processing: Techniques and Applications,” Universidad de Málaga, 2017.

[6] M. Muller, S. Ewert, and S. Kreuzer, “Making chroma features more robust to timbre changes,” *Proceedinsg of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 1877–1880.

[7] M. Muller and S. Ewert, “Towards timbre-invariant audio features for harmony-based music,” *IEEE Trans. Audio. Speech. Lang. Processing*, vol. 18, no. 3, pp. 649–662, Mar. 2010.

[8] M. Müller and S. Ewert, “Chroma toolbox: Matlab implementations for extracting variants of chroma-based audio features,” *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, 2011, pp. 215–220.

[9] F. Korzeniowski and G. Widmer, “Feature learning for chord recognition: The deep

chroma extractor," *Proceedings of the 17h International Society for Music Information Retrieval Conference (ISMIR)*, 2016, pp. 37–43.

[10] J. d. J. Guerrero-Turrubiates, S. E. Gonzalez-Reyna, S. E. Ledesma-Orozco, and J. G. Avina-Cervantes, "Pitch estimation for musical note recognition using artificial neural networks," *Proceedings of the 2014 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, 2014, pp. 53–58.

[11] E. J. Humphrey and J. P. Bello, "Rethinking automatic chord recognition with convolutional neural networks," *Proceedings of the 2012 11th International Conference on Machine Learning and Applications*, 2012, vol. 2, pp. 357–362.

[12] S. Sigtia, N. Boulanger-Lewandowski, and S. Dixon, "Audio chord recognition with a hybrid recurrent neural networks," *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 127–133.

[13] S. Blasiak, H. Rangwala, "A hidden Markov model variant for sequence classification," *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, 2006, pp. 1192–1197.

[14] M. Gales and S. Young, "The application of hidden Markov models in speech recognition," *Found. Trends Signal Process.*, vol. 1, no. 3, pp. 195–304, 2007.

[15] A. Amrouche, A. Taleb-ahmed, J. M. Rouvaen, and M. C. E. Yagoub, "A robust speech recognition system using a general regression neural network," *Int. J. Comput.*, vol. 6, no. 3, pp. 6–15, 2007.

[16] L. Cuiling, "English speech recognition method based on hidden Markov model," *Proceedings of the 2016 International Conference on Smart Grid and Electrical Automation (ICSGEA)*, 2016, pp. 94–97.

[17] H. Zhang, "Video action recognition based on hidden Markov model combined with particle swarm," *Int. J. Comput. Sci. Inf. Syst.*, vol. 7, no. 2, pp. 1–17, 2013.

[18] Q. Zhang, S. Member, B. Li, and S. Member, "Relative hidden Markov models for video-based evaluation of motion skills in surgical training," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 6, pp. 1206–1218, 2015.

[19] A. Mesa, S. Basterrech, G. Guerberoff, and F. Alvarez-Valin, "Hidden Markov models for gene sequence classification," *Pattern Anal. Appl.*, vol. 19, no. 3, pp. 793–805, Aug. 2016.

[20] K. Lee and M. Slaney, "Automatic chord recognition from audio using an HMM with supervised learning," *Proceedings of the 7th International Society for Music Information Retrieval Conference (ISMIR)*, 2006, pp. 133–137.

[21] Y. Qi, J. W. Paisley, and L. Carin, "Music analysis using hidden Markov mixture models," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5209–5224, Nov. 2007.

[22] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, vol. 2, 1995, pp. 1137–1143.

__Hariyanto__ was born in Ujung Pandang, South Sulawesi, Indonesia, in 1997. He received the B.Sc. on Informatics Engineering from Telkom University, Bandung, Indonesia, 2018. His research interests are speech processing, artificial intelligence, machine learning, and Bayesian network.

__Suyanto__ was born in Jombang, East Java, Indonesia, in 1974. He received the B.Sc. on Informatics Engineering from STT Telkom (now Telkom University), Bandung, Indonesia in 1998, the M.Sc. on Complex Adaptive Systems from Chalmers University of Technology, Goteborg, Sweden, in 2006, and the Ph.D. on Computer Science from Universitas Gadjah Mada in 2016. Since 2000, he joined STT Telkom as a lecturer in School of Computing. His research interests include artificial intelligence, machine learning, deep learning, swarm intelligence, speech processing, and computational linguistics.