Steganographic-Algorithm and Length Estimation Classification on MP3 Steganalysis with Convolutional Neural Network

Muhammad Rizki Duwinanto
School of Electrical Engineering and Informatics
Bandung Institute of Technology, Ganesha Street 10
Bandung 40132, Indonesia
rizkiduwinanto@gmail.com

Rinaldi Munir
School of Electrical Engineering and Informatics
Bandung Institute of Technology, Ganesha Street 10
Bandung 40132, Indonesia
rinaldi@informatika.org

***Abstract*— Steganography is a method of embedding secret messages into a cover file in the form of text, audio, picture or video, so that the message is not suspected by those who are not authorized to open the message. The technique to find out whether the cover media is a stego file or not is steganalysis. In this study, detection of hidden messages focused on MP3 files inserted by the MP3Stego algorithm and Equal Length Entropy Codes Substitution to classify based on algorithms and the estimated length of the message and detect cover files. In conducting this research, it is necessary to know the audio features of MP3, build suitable deep learning methods and the performance of the models that have been produced. The proposed solution for these two problems is to use the QMDCT audio feature and deep learning architecture with Convolutional Neural Network. The results of this study are the best algorithm classification model with an accuracy performance of 91.78% and F1-Score 92.22% and the best classification model for message length estimation has an accuracy performance of 24.16% and F1-Score 21.40%. Thus, the proposal of deep learning architecture is good in classifying algorithms and covers, but still poor in classifying the estimated length of the message.**

***Keywords*—Convolutional Neural Network, Steganalysis, MP3, Classification, Steganography**

I. INTRODUCTION

Steganography is a method of embedding secret messages into a cover file in the form of text, audio, picture or video. Steganography aims to be a secure method of message delivery so that no unprivileged third parties become suspicious of the hidden message inside the stego file [1]. Steganography hides the message into the file inside whereas cryptography creates an encrypted message that provides no information to anyone that cannot decrypt it.

An audio file is often used as a cover file for steganography. Audio media is often used in steganography because the media is hard to suspect in storing hidden messages in the form of soundwaves that are heard on the ear. MP3 files are often used as a format for audio steganography [2]. Steganography is used in MP3 files in order to protect MP3 files that contain music from piracy [3]. However, steganography in MP3 files is also used for deplorable motives such as computer virus infection and illegal communication [4]. Therefore, a technique is needed to find out which audio files contain hidden messages.

The technique to discover whether the cover media is a stego file or not is steganalysis. Steganalysis can analyze stego-files based on features that are owned or statistically. Steganalysis can also be classified according to its dependence on certain algorithms, namely targeted steganalysis and blind steganalysis [9]. Targeted steganalysis is more accurate in classifying media cover or not. However, targeted steganalysis is very limited compared to blind steganalysis.

The method commonly used in blind steganalysis is machine learning such as Support Vector Machine and Decision Tree Learning in stego file and cover file classification. The blind steganalysis method for audio can also be developed in deep learning, one of which is Convolutional Neural Network, Recurrent Neural Network, Time Delay Neural Network, etc [10]. However, blind steganalysis that was done in recent works can only classify MP3 audio files as stego-files or not.

Convolutional Neural Network was chosen as a steganalysis method in this classification because it was used in classifying images and audio [10]. Steganalysis, although different from conventional image/audio classifications, can also be used by taking artifacts produced from steganography rather than the media content [7]. CNN can also extract features from the media directly needed for steganalysis and increase accuracy of hidden message detection [11].

Therefore, research is needed whether blind steganalysis on audio MP3 cover media with Convolutional Neural Network can produce the classification results of the steganography algorithm used and the estimated message length.

II. PRELIMINARIES

A. MP3 Overview

MP3 or MPEG Layer III is a file encoding audio files in digital audio. MP3 was originally part of the Moving Picture Experts Group (MPEG) with MPEG-1 and MPEG-2. MP3 has good compression, although it is still lossy, so the file size becomes smaller. MP3 compression works by reducing the accuracy of sound components such as frequencies that cannot

978-1-7281-5118-2/19/$31.00 ©2019 IEEE

be heard by humans. This method is known as perceptual or psychoacoustic coding. Uncompressed digital audio files use the PCM format for digital audio storage and transmission. The file can be compressed and converted into an MP3 file.

B. MP3 Steganography

1) MP3Stego

MP3Stego is a steganographic algorithm that stores hidden messages in MP3 files during the process of compression from the original WAV audio file input [6]. The data is compressed first, then encrypted with the block cipher Data Encryption Standard (DES) algorithm and hidden in MP3 bitstream.

The process of concealment occurs in the Inner Loop. Inner Loop quantifies input data and increases the step size of the quantizer until quantized data can be encoded with available bits. The other loop will check the distortion provided from quantization not to exceed the psychoacoustic model.

The part2_3_length variable stores the number of data bits for scale factors and the Huffman code in MP3 bitstream. The bit will be encoded as parity by changing the final condition of repetition of the Inner Loop. Randomly, the selected part2_3_length value will be modified, the selection will use a pseudorandom generator based on SHA-1.

2) Huffmann Coding Substitution and Equal Length Entropy Codes Substitution

Huffman Code Substitution is a steganographic algorithm specifically for compressed bitstream MP3s that are Huffman codes. HCM can get more capacity, better security, and simpler computing.

There are several HCM algorithms found. According to Gao, the Huffman code will be extracted to be a substitution in accordance with the similarities of codewords [13]. The Huffman Code is sorted by lexicon with values from the QMDCT coefficient to minimize modification of the QMDCT coefficient.

Equal Length Entropy Codes Substitution (EECS) is an adaptive MP3 algorithm that is a development of the Huffman Code Substitution [8]. There is a content-aware distortion function that is designed to obtain the optimal masking effect with the psychoacoustic model of this algorithm, which makes this algorithm safer than the previous method on MP3.

C. Steganalysis

Steganalysis is a technique for determining whether a file is a stego file or not. Steganalysis detects whether there is hidden data in various cover media files whether it is steganography or not. Steganalysis is very closely related to steganography and is useful in security and forensics for detecting stego-files.

Steganalysis is grouped based on input to find out the closing of the media or not a feature-based steganalysis and statistical steganalysis. Steganalysis can also be grouped based on compatibility with certain steganography targeted at steganalysis and blind steganalysis.

III. Related Works

Chen, Luo, & Li proposed an architecture with convolutional layers with fixed kernel (-1, 2, -1) with 7 layers of groups to conduct audio steganalysis experiments with CNN. Each group has a 1 × 5 convolutional layer, 1 × 1 convolutional kernel and a subsampling (pooling layer) [6]. The highest accuracy achieved is 88.85% and has a stable performance in learning. This classification only classifies stego-files or not in the temporal domain and WAV format.

Wang, Yang, Yi, Zhao, & Xu proposed to detect audio steganalysis with CNN in the entropy code domain. CNN architecture is a high pass filter, a combination of six convolutional layer blocks with each block combination, namely 3 × 3 convolutional layers, 1 × 1 convolutional kernel, activation function tanh and max pooling layer [5]. Fully connected layers and batch normalization layers are then placed at the end. Then cross-entropy loss is updated on each network parameter. Input data entered the network is a QMDCT (Quantified Modified Discrete Cosine Transform). The highest accuracy of this experiment to classify HCM with embedding rate 0,1 from the cover is 75,92%. However, this classification only works in binary classification and cannot determined algorithm except EECS and HCM algorithm.

IV. Proposed Method

In conducting this research, it is necessary to know the audio features of MP3, build suitable deep learning methods and the performance of the models that have been produced.

A. QMDCT

MDCT that has gone through the quantization process is called QMDCT (Quantified Modified Discrete Cosine Transform). This QMDCT is compressed using the Huffman code. QMDCT consists of three parts, namely the big-value part, count1 part, zero part.

Large parts worth QMDCT encoded into one Huffman code. The big-value section can be subdivided into three subsections, namely region 1, region 2 and region 0. This section will be entered independently with the Hufmann table from table 0 to table 31. If the value of the QMDCT coefficient is less than 15 then the value is directly encoded. If not less than 15, then the excess value will be represented by linbits. If the coefficient value is not 0, then the bit sign is used. A value of 0 indicates a positive number and value 1 indicates a negative number.

The QMDCT coefficient in the count1 section is {-1, 0, 1} and each multiple of 4 coefficients will be coded with one Huffman code with tables 32 and 33. All coefficients in the zero section are zero and the coefficient will not be encoded.

Hc as the Huffman code in the QMDCT coefficient, linbits_x signifies the linbits of the first coefficient, sign indicates the sign of the first coefficient, linbits_y signifies linbits from the comparison, sign_y indicates the bit sign from the second coefficient. Sign_v, sign_w, sign_x and sign_y indicates the sign bits of the four QMDCT coefficients in the count1 section.

The input feature is the extraction of Markov features from rows and columns from the quantified Modified Cosine Transform (QMDCT) coefficient [12]. The QMDCT feature was extracted in the MP3 decoding process. The QMDCT feature with a size of 200 × 380 is defined by MQ in Equation 1. In this case the variable i represents the number of channels and the variable j represents the index of the QMDCT coefficient on the channel. The range of the variable i from selected frames N that satisfies i ∈ [0, 4N].

$$M_Q = \begin{bmatrix} Q_{11} & \cdots & Q_{1j} \\ \vdots & \ddots & \vdots \\ Q_{i1} & \cdots & Q_{ij} \end{bmatrix}; \; i \in [1,200]; \quad j \in [1,380] \qquad (1)$$

There are a few advantages of using QMDCT coefficient that is described by equation 1 as a feature. First, QMDCT is encoded in the Huffman code, which in this case Huffman Code recovery will result in changes in the QMDCT coefficient. Second, the modification of the QMDCT is greater than the modification of sampling dots in the temporal domain on the hiding method data. Finally, the statistical characteristics of the QMDCT coefficient matrix are more effective in detecting MP3 steganography algorithms.

B. Proposed CNN Architecture

Convolutional Neural Network Architecture that corresponds to this MP3 audio file will consist of a combination of six convolutional layer blocks with each block combination of 3 × 3 convolutional layers, 1 × 1 convolutional kernel, function activation of tanh and max pooling layer. The network is shown in Figure 1. Fully connected layers and batch normalization layers. This CNN architecture is the adoption of Wang, Yang, Yi, Zhao, & Xu (2018) which is modified according to this study to support multilabel classification [11]. Modification include omittance of High Pass Filter which is not compatible with this classification which includes class from MP3Stego. Figure 1 is an architectural proposal from CNN that will be used during the experiment.

The CNN architecture will receive input in the form of a QMDCT feature extracted from an Audio MP3 file. QMDCT will then enter convolutional layer (Conv) as the main component of CNN proposed in this network is a 3 × 3 convolutional layers to get features of data input and convolutional 1 × 1 to get interaction and information integration of each channel, reduction of parameters network, reduce overfitting.

Pooling layer is used to reduce the dimensions of features by maintaining data input and reducing parameters. Pooling used is max pooling. The output of max pooling is the maximum value of the filter that shifts, which maintains texture information, in contrast to average pooling, which maintains background information. From the steganographic characteristics and complexity of the model, only max pooling will be used in the architecture. Batch normalization layer (BN) serves to accelerate convergence and the addition of accuracy of Batch normalization detection also reduces overfitting. Batch normalization layer works by normalizing transformation by looking for variances of each batch value with errors. The

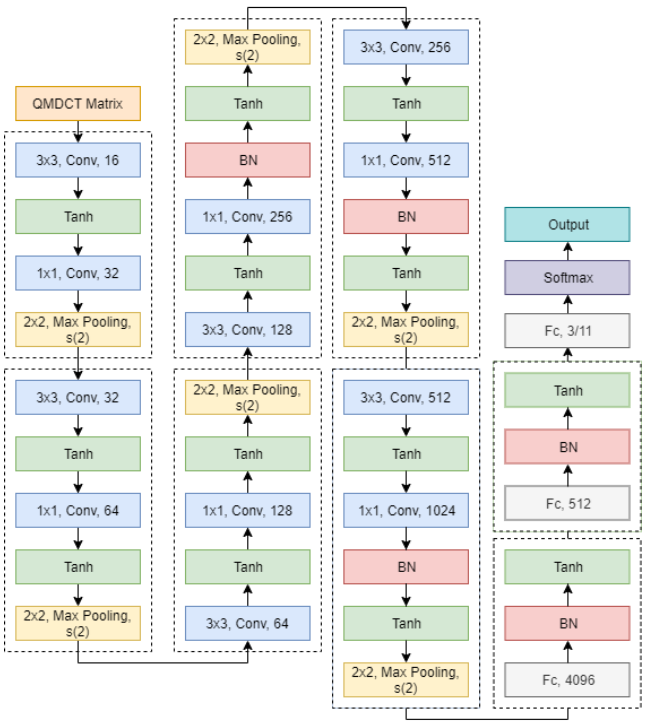

Figure 1 Proposed CNN Architecture

activation function used is a hyperbolic tangent or tanh rather than relu because of its limited range.

Finally, the fully connected layer (Fc) is used to classify existing stego-MP3-files into various classes, namely the steganographic algorithm used and the estimated length of hidden messages. Fully connected layers will have the number of output neurons as many as the number of classes used for classification will produce values that represent each class.

Before the outcome becomes the result of a classification, the output of the process from the fully connected layer is processed with the softmax function first. The softmax function is used to issue multi class classification results, namely the steganography algorithm in the form of the probability of each class. The number of probabilities of each class is one hundred percent.

Then the results of the classification of the softmax function will be calculated loss value with Cross Entropy Loss or log loss. The performance of the classification model that has binary output. Cross Entropy Loss value if the greater indicates a very far probability with the observation label.The resulting loss will be optimized with the adam function. The adam or adaptive moment estimation function is an optimization function which is the development of AdaGrad and RMSProp. Adam is an efficient stochastic optimization method that accepts first-order gradients with small memory. The adam function calculates the individual learning rate with parameters from the first and second moments. The advantage of adam is that the large update parameter is unchanged from changes in the learning gradient so that it is adaptive in determining the step size.

V. EXPERIMENT AND RESULTS

A. Experiment

Before conducting experiments, data in the form of MP3 audio files must be prepared for model training. Audio MP3 data that must be prepared starts from 17940 samples of WAV audio files that are inserted in rotation with the MP3Stego Algorithm, HCM and not inserted and the rotating class lengths are 1-10 bytes, 11-20 bytes, 21-30 bytes, 31-40 bytes , 41-50 bytes, 51-60 bytes, 61-70 bytes, 71-90 bytes, 81-90 bytes and 91-100 bytes. If it is not inserted it will immediately get a 0 bytes long class. 17940 WAV audio sample file is used in this experiment.

Next is the labeling of audio files with message length and inserted algorithms. Labels are stored in CSV form and sequential separation will be carried out before the training and testing stages. All data will be labeled according to the inserted algorithm and the estimated length of the message unless the data that is not inserted will be labeled with a negative algorithm and zero length.

After that, the features of the MP3 file are extracted in the form of QMDCT or Quantified Modified Cosine Transform. Then just before being used in the experiment, the data is divided into training data and test data with a proportion of 5: 1. The training data is then subdivided with the proportion of 5: 1 to become training data and validation data to support the hold-out scheme.

The experiment will be carried out using the convolutional neural network as the initial architecture and various factors. Training for models for classification of algorithms and covers will use batch values of ten sizes, while for models for classification of estimated length of messages using batch values of size four.

Testing the best model produced in the previous process will use the predetermined test data when preparing data. First, the test data is classified with the model that has been built in the previous stage according to the minibatch value. Testing for models for classification of algorithms and covers will use batch values of ten sizes, while for models for classification of estimated message lengths use batch values of size four.

The classification results will consist of the steganographic algorithm class used and the estimated length of the per-batch message class. After that, the classification results from the test data are measured for their performance against the label of the original test data with confusion matrix, accuracy, and F1-Score, so that an evaluation of the models that have been built in the previous stage is produced.

B. Results

The test results for message detection in MP3 audio files for algorithm classification models produce the best performance with an accuracy of 91.78% and F1-Score of 92.22% in the sixth iteration. The test results for the detection of messages in MP3 audio files for message length classification models

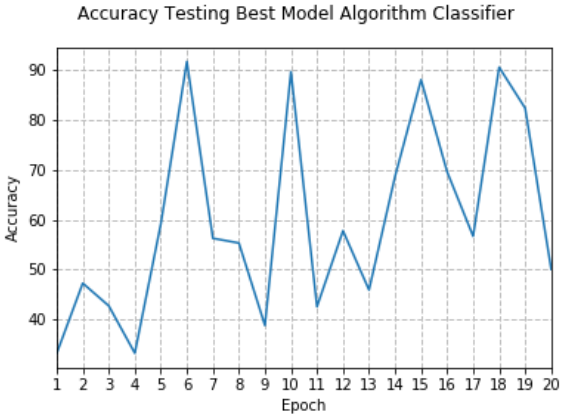

Figure 2 Accuracy of Testing of the Best Models of Algorithm Classification

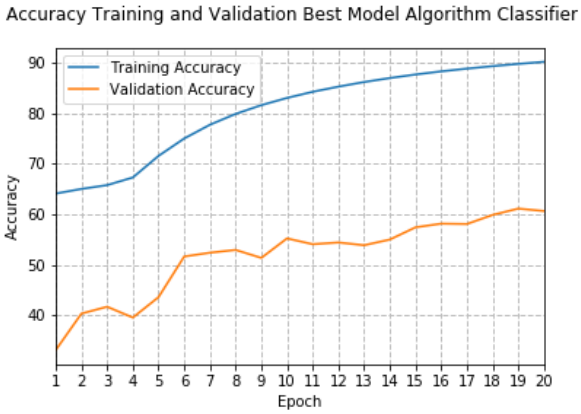

Figure 3 Accuracy of Validation and Training of the Best Models of Algorithm Classification

produced the best performance, namely accuracy of 24.16% and F1-Score of 21.40% in the fourteenth iteration.

Figure 2 shows the testing accuracy of algorithm classification of 20 epochs. It can be seen in Figure 2 that the best accuracy is found in the sixth iteration / epoch. Figure 3 shows the validation and training accuracy of algorithm classification of 20 epochs. If seen in the training and validation graph in Figure 3, it was found that the accuracy of the validation was still up and not quite convergent, but it was found that the accuracy of the next epoch dropped significantly, so that the overfit in the seventh iteration. This is also supported by data loss validation which illustrates that loss changes to rise after peak iteration. The accuracy graph can be made possible from the influence of the existing adam optimization function, due to the fitting process involving three classes namely cover, MP3Stego algorithm and HCM algorithm.

From the classifications of the three classes that existed during the testing process, namely the cover, the MP3Stego algorithm and the HCM algorithm. The fitting process is generally seen in the HCM algorithm and cover. This is because when the QMDCT results from the HCM encoding process hardly change significantly from the cover file.

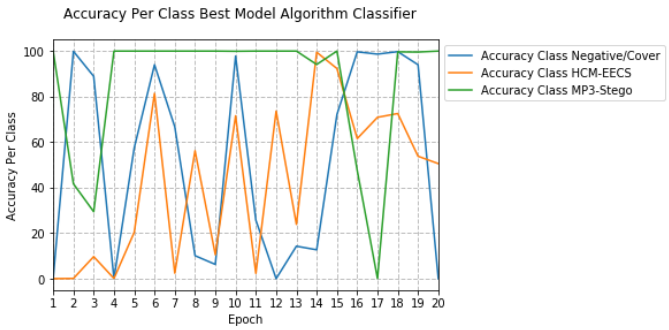

Figure 4 Per Class Accuracy from The Best Algorithm Model Classification

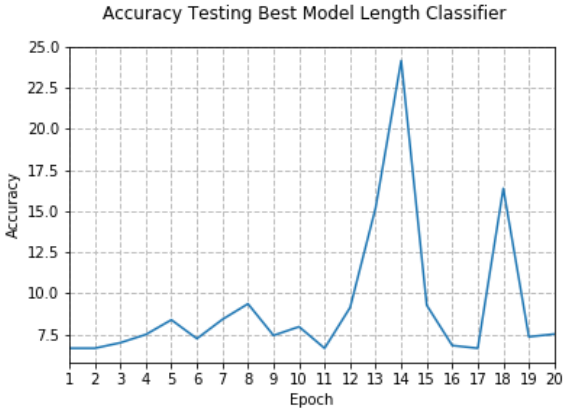

Figure 5 Accuracy of Testing of the Best Models of Length Estimation Classification

The accuracy of the algorithm classification by class is shown in Figure 4. In the epoch with the best performance, the accuracy of the HCM-EECS label was only 81% while the other two classes got good accuracy of 94% for the cover class and 100% for MP3Stego as in Figure 4. Accuracy of 91.78% and F1-Score of 92.22% indicates that the model has been able to classify stego-files based on algorithms and cover files properly, although there are some inaccuracies in the HCM-EECS class and cover class.

Figure 5 shows the testing accuracy of length estimation classification of 20 epochs, while Figure 6 shows the validation and training accuracy of length estimation classification of 20 epochs. The best accuracy for the model in determining the best message length is at the fourteenth epoch. If seen in the training and validation graph in Figure 6, it was found that the accuracy of validation began to show stagnation from the peak value of 11.39% in the fourteenth iteration and began to show a decrease in the validation accuracy value in the next iteration although the accuracy of the training value continued the iteration / epoch. It can be seen in the test graph in Figure 5 that the accuracy of the next epoch has dropped significantly, so that it can be analyzed that the fifteenth iteration to the next illustrates the occurrence of overfit in the fifteenth iteration. This is also supported by the data loss validation which illustrates that the loss that had always dropped, changes to rise after the fourteenth iteration.

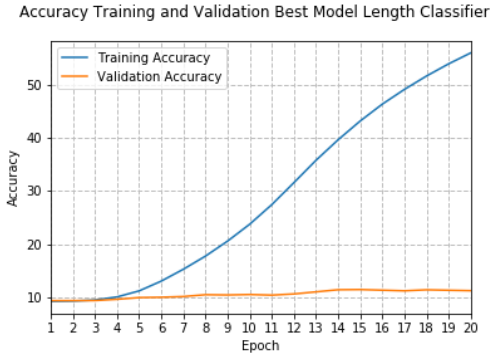

Figure 6 Accuracy of Validation and Training of the Best Models of Length Estimation Classification

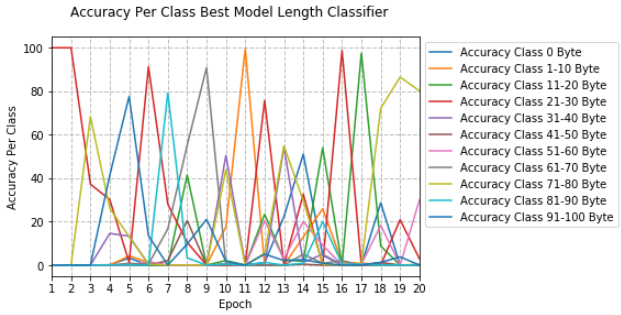

Figure 7 Per Class Accuracy from The Length Estimation Best Model Classification

From the classification of eleven classes that exist during the testing process, namely the message length 0 bytes, 1-10 bytes, 11-20 bytes, 21-30 bytes, 31-40 bytes, 41-50 bytes, 51-60 bytes, 61-70 bytes, 71-80 bytes, 81-90 bytes and 91-100 bytes, the accuracy of eleven classes per iteration is obtained in Figure 7. Fitting process is seen to occur in each iteration with each iteration there is always a majority class with a significant although the data used is balanced. The fourteenth iteration shows that all classes can be predicted with the highest-class accuracy is the 0-byte class with an accuracy of 51.04% and the lowest class accuracy is 31-40 bytes with an accuracy of 0.42%. Other iterations tend to set label predictions for a class. This shows that the model created cannot yet classify the length of the message in MP3 audio with a duration of ten seconds which has a maximum load of as many as one hundred bytes.

If further analyzed the estimated message length classification, it was found that the modification of QMDCT with the steganographic algorithm with MP3Stego and HCM does not reflect the estimated number of message lengths, therefore creates more inaccuracy in classification. There are two factors that occur during insertion that make it difficult to determine the estimated message length with the QMDCT coefficient. First, the change in the QMDCT coefficient value occurs with the formula for the coefficient changes on MP3Stego and HCM that correspond to the step-quantizer value, the bit of the message and the initial spectral value adjusted to the original MP3 file obtained before the inner-loop process during the MP3 compression process. Second, the

insertion of messages on MP3Stego also occurs randomly at the frame location of QMDCT, while the bit substitution on HCM also affects the location of the bits contained in the message. Also, it can also be influenced by the CNN architecture that is implemented. The proposed CNN architecture can only get patterns from the values of the steganographic artifacts contained in the QMDCT coefficient according to the algorithm but does not get the pattern of the estimated message length.

VI. Comparison with Wang

Previous studies can only classify stego or cover files. Tests conducted in previous studies only use data that is pasted with one algorithm in each test. In contrast, this study classifies MP3 audio files based on algorithm classes and estimated message lengths and more than one algorithm used for each test.

If the same data used in this research experiment is inserted in accordance with all algorithms on the limits, namely MP3Stego and HCM, an accuracy of 96.64% is obtained to obtain a binary classification in the form of a cover or stego file on the third epoch. While the results of the classification of the study if using a classification based on the algorithm to get the accuracy of the cover class of 93.99% obtained at the sixth accuracy, this accuracy is obtained by comparing the two algorithm classes with the cover class. When compared with the highest accuracy results of this study which classifies the algorithm, there is a decrease in accuracy of as much as two percent.

VII. Conclusion and Future Works

The audio MP3 feature that has been chosen, namely QMDCT, can be used to support MP3 audio steganalysis with the Convolutional Neural Network method to classify MP3 audio files based on algorithmic classes, but not so good to support classification based on estimated-length classes. The Convolutional Neural Network model that has been created can detect hidden messages in the stego-MP3-file by classifying MP3 audio files based on the algorithm class used and the cover with fairly good performance, However, the Convolutional Neural Network model that has been created has not been able to estimate message length that was embedded into stego-MP3-file with the same good performance.

Future works include classification that can classify algorithm not only to the MP3 compression domain, but also to the temporal domain and transformation. Also, classification with this Convolutional Neural Network should be implemented in order to receive data with different lengths of duration and can also detect the message length that is bigger and according to the length of the message.

Acknowledgment

I would like to express my deep gratitude to Mr. Rinaldi, my research supervisors, for their patient guidance, enthusiastic encouragement and useful critiques of this research work. I wish to thank my parents for their support and encouragement throughout my study. Finally, I would like to express grateful to Allah SWT who give me a chance to do this research.

References

[1] J. Antony, Sobin and Sherly, "Audio Steganography in Wavelet Domain – A Survey," *International Journal of Computer Applications,* pp. 33-37, 2012.

[2] W. Bender, D. Gruhl, N. Morimoto and A. Lu, "Techniques for data hiding," *IBM System,* pp. 313-316, 1996.

[3] N. Cvejic, Algorithms for Audio Watermarking and Steganography, Oulun: Unpublished, 2004.

[4] N. Meghanathan and L. Nayak, "Steganalysis Algorithms For Detecting The Hidden Information In Image, Audio And Video Cover Media," in *International Journal of Network Secuiry & Its Applicaiton*, St, Jackson, 2010.

[5] A. Nissar and A. Mir, "Classification of steganalysis techniques: A study," Elsevier, 2010.

[6] B. Chen, W. Luo and H. Li, "Audio Steganalysis With Convolutional Neural Network," in *IH&&MMSec'17*, Philadelpia, 2017.

[7] G. Xu and H.-Z. S. Y.-Q. Wu, "Structural Design of Convolutional Neural Networks," in *IEEE SIGNAL PROCESSING LETTERS*, 2016.

[8] M. Noto, "MP3Stego: Hiding Text in MP3 Files," SANS Institute, 2001.

[9] H. Gao, "The MP3 steganography algorithm based on Huffman coding.," *Acta Scientiarum Naturalium Universitatis Sunyatseni 4 (,* p. 009, 2007.

[10] K. Yang, X. Yi, X. Zhao and L. Zhou, "Adaptive MP3 Steganography Using Equal Length Entropy Codes Substitution.," in *Digital Forensics and Watermarking - 16th International Workshop, IWDW*, Magdeburg, 2017.

[11] Y. Wang, K. Yang, X. Yi, X. Zhao and Z. Xu, "CNN-based Steganalysis of MP3 Steganography in the Entropy Code Domain," in *IH&MMSec*, Inssbruck, 2018.

[12] C. Jin, R. Wang, D. Yan, P. Ma and K. Yan, "A NOVEL DETECTION SCHEME FOR MP3STEGO WITH LOW PAYLOAD," *IEEE,* pp. 602-606, 2014.

[13] X. Yu, R. Wang and D. Yan, "Detecting MP3Stego using Calibrated Side Information Features," *JOURNAL OF SOFTWARE, VOL. 8,* pp. 2628-2636, 2013.