ICIC Express Letters
Volume **18**, Number **9**, September **2024**
 ISSN 1881-803X
pp. **971–978**

AN MLP CLASSIFICATION METHOD FOR SOUND-BASED DRONE DETECTION SYSTEM

Risa Farrid Christianti[1,2], Azhari Azhari[1,*] and Andi Dharmawan[1]

[1]Department of Computer Science and Electronics
Faculty of Mathematics and Natural Sciences
Universitas Gadjah Mada
Building C, 4th Floor, North Sekip, Bulaksumur, Yogyakarta 55281, Indonesia
risachristianti78@mail.ugm.ac.id; andi_dharmawan@ugm.ac.id
*Corresponding author: arisn@ugm.ac.id

[2]Department of Electrical Engineering
Faculty of Telecommunication and Electrical Engineering
Institut Teknologi Telkom Purwokerto
Jl. D.I Panjaitan No. 128, Purwokerto 53147, Central Java, Indonesia

Received September 2023; accepted December 2023

Abstract. *With the increase in various types and production of drones for multiple purposes, there may be a danger of using drones illegally. This condition harms protected public areas, such as tourist attractions and places of worship, offices, etc. Therefore, it is crucial to detect events or conditions considered detrimental so that security operators can obtain this information and identify the presence of drones. This paper introduces a drone detection method based on sound data using a combination of Log-mel spectrogram and Mel-Frequency Cepstrum Coefficients (MFCC) features. Therefore, this paper's aim is to offer a deep learning methodology applied to tasks related to drone detection using voice data. Based on experimental results, this MLP-based model detected the presence of drones 549 times and caught non-drones 241 times from the data tested (20% of the dataset of 4,093 data). The accuracy percentage achieved is 96.46%; in the test data, the precision value is 96%. The highest recall percentage and f1-score were achieved in the Drone class (recall = 98%, f1-score = 97%). The confusion matrix from the model evaluation shows that the model made nine times the error of detecting objects as Non-Drone (FN = 1.1%).*
Keywords: Drone detection, Sound data, MFCC, Log-mel spectrogram, MLP

1. **Introduction.** The increasing use of drones threatens public safety and personal privacy. Therefore, it is necessary to have a system that can detect the presence of drones, which are considered detrimental, so that information and identification of drone conditions can be obtained. Several manufacturers who develop anti-drone systems can detect drones in protected air areas and effectively neutralize threats, so a system called counter-UAV technology (c-UAV) was created, with several features offered [1,2].

The drone detection system is a vital security function in many situations today, such as protecting critical infrastructure, mobile perimeter and border monitoring, and unmanned traffic management systems [3,4]. Since 2016, research has proposed a system for detecting drones. This system can perform functions such as motion detection, motion tracking, which will draw trajectories along the drone path, feature detection, and audio detection, which can identify the presence of a drone nearby by frequency analysis [5]. In order to implement this system, it had to use high technology with certain specifications that can detect drones. In order to simplify and optimize the drone detection system, array microphones fail to achieve high accuracy and operational range. Another disadvantage

DOI: 10.24507/icicel.18.09.971

is the unfeasible nature of the system in urban or noisy environments such as airports [6]. Subsequent studies revealed that radar technology is essential for detecting, tracking, classifying, and identifying airborne threats. Therefore, it was in the first phase of the entire c-UAV processing system. Due to its long-range, good positioning accuracy, and weather independence, the radar sensor fits into the critical sensors of the c-UAV system. However, this radar sensor must be combined with other sensors, such as visual, infrared, acoustic, or SIGINT (Signal Intelligence) based detectors, often used to detect, classify and identify drones [7-10]. The interesting point is that to overcome the problem of demanding high-tech computing performance with certain specifications, Wang et al. [10] developed a real-time drone detection and analysis system using sound data from DJI Phantom 1 and 2 drones and environmental noise from European football stadiums, using two different machine learning algorithms, namely the PIL (Plotted Image Machine Learning) technique and K-Nearest Neighbor (K-NN) [9].

This paper proposes a sound-based, deep-learning drone detection method that combines the Log-mel spectrogram and the MFCC feature of sound signals. The sound signal will be captured by an acoustic sensor (array microphone) and then processed by extracting the Log-mel spectrogram and MFCC features from the signal so that it can detect drones effectively. With this method, the proposed drone detection system is expected to be feasible in natural scenes, which can accurately detect whether there are drones in the surrounding environment with a low False Negative (FN). However, the new challenge of the proposed drone detection system is integrating different types of sensors to complete the function of the drone detection system so that the notification or information of this system can be more detailed. This paper contributes to finding an alternative way to classify drones using the flattening Log-mel spectogram and MFCC feature values of drone sounds.

The paper is structured as follows. Section 2 introduces the dataset, the structure of the proposed deep learning model for detecting drones, and the method of combining them. Section 3 discusses the process of the proposed detection method in detail. Section 4 presents and analyzes the experimental results that have been carried out. Moreover, finally, conclusions and future work are drawn in Section 5. This paper contributes to developing sound data classification methods by feature extraction of homogeneous multisensor data in drone sound detection systems. This method is developed using the MFCC algorithm and feature extraction learning based on deep learning with a multi-layer perceptron algorithm.

2. **Problem Statement and Preliminary.**

Drone detection systems typically utilize various features emitted from drones, such as heat, sound, and Radio Frequency (RF) signals, to communicate with remote operators. Drone detection systems collect sensor data to confirm drone presence in nearby areas. Because drones have distinctive sound characteristics, these distinctive characteristics are detected by the development of intelligent detection, recognition, and classification systems.

Regarding detecting drone sound, the first step is to pre-process the flying object sound signal dataset and then extract the sound signal to obtain two sound features: the Log-mel spectrogram and MFCC. Furthermore, these two features are combined (concatenated) with the splicing (flattened) method and trained with a Multi-Layer Perceptron (MLP) network model with two hidden layers. For obtaining the dataset, the secondary dataset is adapted to public environmental conditions, with limited scope and noise often occurs. The primary dataset is also added by recording the drone from several real-time viewpoints for a few seconds. Primary data is focused on drone sound with various types of drones and their recording distance (1-30 meters).

Figure 1 describes the proposed drone detection system. Acoustic data consists of two data classes in .WAV format. Two classes are taken as input for the acoustic data: drone

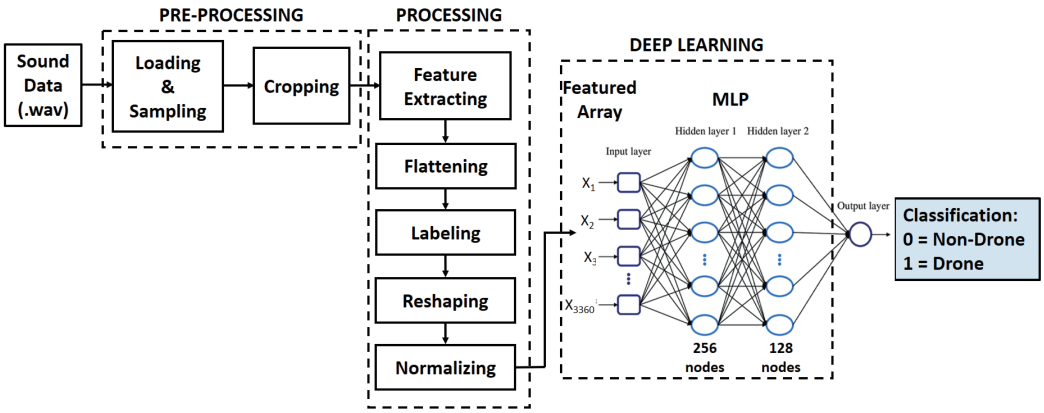


FIGURE 1. Block diagram of the proposed drone detection system

sound and environmental (non-drone) sound; in this case, the sound taken from the environment is the sound of helicopters, traffic, and thunder. Before processing, the data from each class equalizes the sample size by cutting the audio sample into the first 10,000 samples. This process must be done so that the resulting acoustic feature extraction is valid and error-free. Furthermore, this audio sample will be extracted by taking two sound feature parameters: the Log-mel spectrogram and Mel-Frequency Cepstrum Coefficiency (MFCC).

Secondary dataset collection is done by downloading links related to environmental sound included in the Non-Drone class. In contrast, the others are included in the Drone class. The primary data is taken from real drone sound recordings for a particular duration and converted into a .wav file format to be combined with the Drone class. Then, using Python, proceed to the secondary data pre-processing stage for feature extraction using the MFCC algorithm. Each trained voice data is converted into MFCC array data. Acoustic data is processed to obtain specific drone sound characteristic parameters. Learning patterns are feature extraction in analyzing and preparing audio files for deep learning. Python library Librosa will be used for audio feature processing [8]. This feature extraction process produces MFCC parameter values in the form of array data that shows the characteristics of each class of acoustic data. This feature value will be used as input for the deep learning method as a classifier.

The drone's acoustic pattern must be recognized to extract specific features from each frame. Features from raw acoustic data are extracted for model training to reduce data dimensionality, ignore redundant information, and simplify the training task. For traditional machine learning, an extensive list of features must be manually extracted and optimized to improve the algorithm. If the feature is not optimal for classification purposes, in this case, for a drone detection system, then the model's performance will be limited. Therefore, extracting features becomes an additional challenge, as there is no guarantee that they are optimized for classification purposes. Feature extraction with the MFCC algorithm is a good choice when working with deep learning models, which perform best with features close to the original audio signal, such as MFCC.

The development of this method is also applied to the data training process using the Multi-Layer Perceptron (MLP) algorithm. The MLP model structure is provided with hidden layers and defined activations. The Dense layer represents a fully connected layer, and the Dropout layer helps prevent overfitting. The output layer uses the sigmoid activation function because the expected output targets are classifications of two classes ("Drone" and "Non-Drone"). After compiling the model, the training uses the extracted sound features and the corresponding labels using the model match method. In addition, the model fit method (model.fit) is applied to determine the epoch, batsh_size, and validation split parameters according to the required design.

3. **System Design.** In this section, methods for detecting drones are introduced. Dataset generation, feature extraction methods, and the proposed network structure are explained in stages. Then, in the final step, this paper explained how the MLP algorithm is applied to train feature extraction to produce an appropriate and reliable model for classifying drones based on sound data.

In this research, the dataset is adapted to the conditions of the public environment, where the scope is limited and noise often occurs. Data collection is also added from accurate primary data by recording drones from several viewpoints in real time for a few seconds. The primary data is focused on the sound of drones with various types of drones and their recording distance (less than 30 meters).

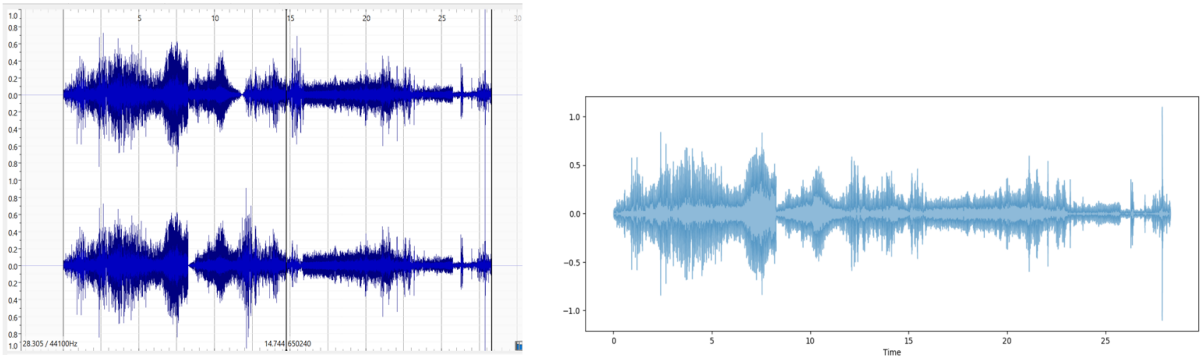
The collected dataset consists of primary and secondary datasets from GitHub and Kaggle. The secondary dataset was collected by downloading several dataset links related to the sound of helicopters and thunder, which were included in the Non-Drone class, while the others were in the Drone class. Then, using Python, proceed to the secondary data pre-processing stage for feature extraction using the Log-mel spectrogram and MFCC. From the twelve spectral feature extractions of sound data, we took the Log-mel spectrogram and MFCC feature extraction, based on previous research, which showed that these two features were the most effective and appropriate for classifying sound with deep learning models. The features of the voice data are stored in numpy array format, using the Librosa feature extraction method [7,10].

3.1. **Pre-processing.** The dataset includes 4,093 sound data files in .WAV format, divided into two categories: drone sounds (1,370 data) and non-drone sounds (2,723 data). The recorded drone sounds come from DJI Phantom 4 Pro and MJX Bugs 2 drones with a duration of 10-20 seconds.

Before pre-processing, the dataset is represented in the form of a spectrogram. Figure 2 illustrates the acoustic data representation of each data class. Figure 2(a) illustrates input acoustic data from the sound sensor in .WAV data format. The acoustic data input is mapped through pre-processing based on its spectrum and represented in a spectrogram, as shown in Figure 2(b). The results of this pre-processing will be used as a dataset ready to be extracted. The dataset in the form of data arrays is presented as shown below:

$$\begin{aligned}\text{array}([\ &8.2923099\text{e-}05,\quad 9.9526718\text{e-}05,\quad 1.0606833\text{e-}04,\quad \ldots,\\ &-3.0119958\text{e-}07,\quad 2.7582385\text{e-}07,\quad -2.4966062\text{e-}07],\ \text{dtype=float32})\end{aligned}$$

The initial stage of pre-processing is to load all the dataset files that have been prepared. Through Librosa, iteration is done through the class labels defined in the class list, which appear to contain the class names “Drone” and “Non-Drone”, to read and sample the data. In sending audio data, the sample rate parameter in the 22050 Hz range is recorded at a



(a) (b)

FIGURE 2. (a) Acoustic data visualization; (b) acoustic data input in spectrogram

specific sample rate. Then, the entire dataset that has taken the sample rate is cropped to the first 10,000 samples to be extracted on a Mel scale. This cropping prevents errors from occurring during processing to create the model. The Mel spectrogram captures the energy distribution of a signal in different frequency bands, where the frequencies are converted to the Mel scale, which is more representative of how humans perceive audio frequencies.

3.2. **Processing.** Each sound sample that has been loaded is sampled at a frequency of 22050 Hz, and the first 10,000 samples are taken. 22050 Hz is often determined by the audio data's characteristics and the pre-processing stage's specific requirements. The initial parameters that must be specified in this process include n_fft = 1,024, hop_length = 512, n_mels = 128, and n_mfcc = 40. From these definitions, it can be explained that each sound sample was divided into 128 frames, the number of FFT points was chosen to be 1,024, and the hop length was set to 512. The following details involved two different feature extractions from voice data into two parameters, namely Log-mel spectrogram and MFCC. These parameter choices can be fine-tuned based on experimentation and analysis of the features extracted from the data. Adjustments can be made to optimize performance for specific tasks or datasets.

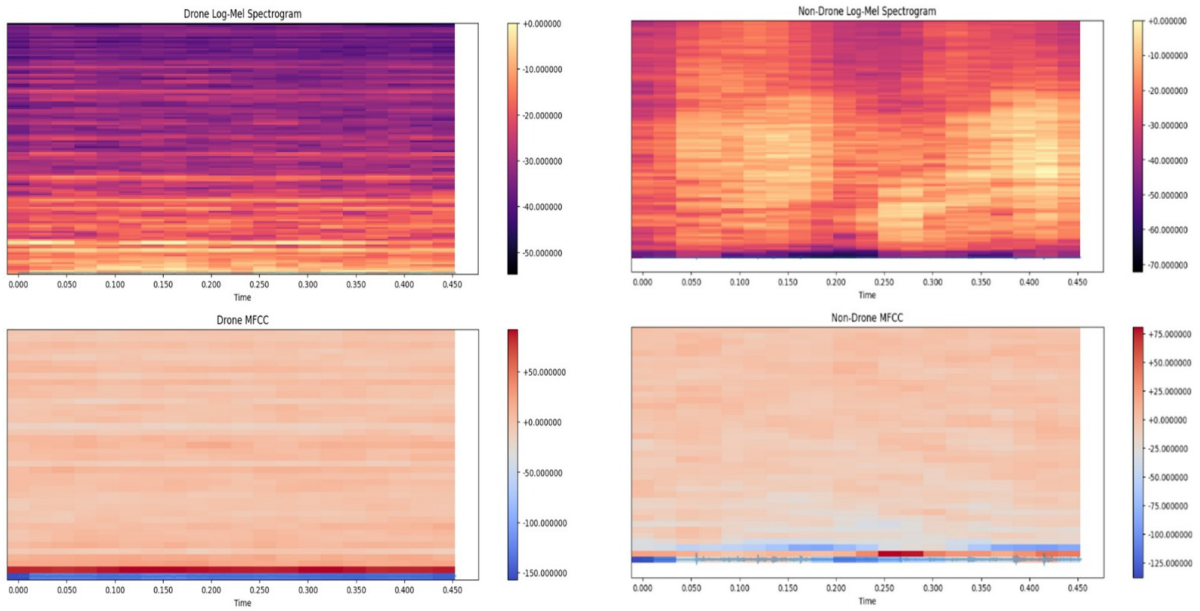
3.2.1. *Extracting feature.* The `librosa.feature.melspectrogram` function calculates the Mel spectrogram of the audio signal y. `n_mels` and `fmax` control the number of Mel bins and the maximum frequency. Then, the information obtained from the Mel spectrogram features is converted into a Log-mel spectrogram scale. The spectrogram logs are represented in decibels in the `librosa.power_to_db(S=S, ref=np.max)` function. Conversion to the Log-mel spectrogram results in Log-mel features, which are reshaped into a flattened 1D array using reshaping, suitable for coupling to MFCC features. Next, enter the MFCC feature extraction process using the `librosa.feature.mfcc` function. The `n_mfcc` parameter controls the number of coefficients to be calculated. As with the Log-mel spectrogram, the resulting MFCC is reshaped into a flattened 1D array. At this step, two features have been obtained from each voice data from the two classes.

Figure 3 visualizes the two features of the Drone and Non-Drone classes. The Log-mel spectrogram dimension is set to 128 to reduce the feature dimension while preserving the sound spectrum features of the audio signal. Librosa extracts the Mel spectrogram and then converts the Mel spectrogram to a logarithmic scale to get the Log-mel spectrogram. The final input size is 128 × 128. Each frame returns 40 features, so the MFCC feature size is 40 × 128.

3.2.2. *Labeling.* This labeling process is a common approach in machine learning, where categorical labels are converted to numeric values for model training. The function included in the program is `to_categorical` with labeling output, which changes the "Drone" and "Non-Drone" labels into numbers, namely [1. 0.] for Drone, while Non-Drone [0. 1.].

3.2.3. *Reshaping and normalizing.* The value of the feature array resulting from the feature extraction process needs to be normalized, considering that the range of values is too extensive. Therefore, the feature array values are normalized before training with the `MinMaxScaler()` function.

3.2.4. *Training datasets.* Before training on the feature array, the train test splitting boundaries are determined to divide the dataset into training and testing sets using the train_test_split function from the `sklearn.model_selection` module. X is the feature (input data) to be split, and y is the associated label variable to be split. test_size is set at 0.2 (20%), which means the dataset is split; 80% is training data, and 20% is test data. random_state is set to 50. The train_test_split function divides the dataset into two sets: one for training the models (X_train and y_train) and one for testing the models (X_test



(a) Drone (b) Non-Drone

FIGURE 3. (color online) Visualization of two feature extraction in (a) Drone and (b) Non-Drone

and y_test). The Multi-Layer Perceptron (MLP) model for binary classification can be seen in the following details.

TABLE 1. MLP model summary

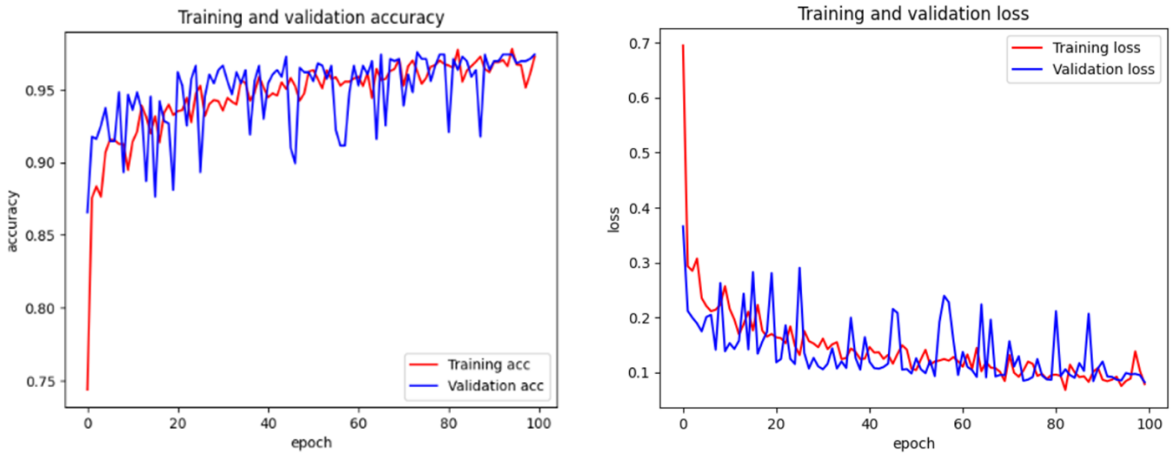
Model: "Sequential"

Layer (type)	Output shape	Param #
Dense (Dense)	(None, 3360)	11292960
dense_1 (Dense)	(None, 256)	860416
dense_2 (Dense)	(None, 128)	32896
dense_3 (Dense)	(None, 1)	129
Total Params: 12,186,401 Trainable Params: 12,186,401 Non Trainable Params: 0		

It can be explained that the neural network model is applied sequentially. Additionally, it is provided dense with 3,360 input nodes (3,360 features). The activation function used here is ReLU (Rectified Linear Unit). The hidden layer consists of two layers with 256 and 128 nodes, respectively, both of which use the ReLU activation function. Model compilation uses the Adam optimizer with a learning rate of 0.001.

4. **Result and Discussion.** The result of MLP classification method can be seen in the graph of Figure 4.

From the experimental results that have been carried out, the percentage of training and validation accuracy increases with increasing epochs (Figure 4). Conversely, the percentage of training and validation loss decreases with increasing epochs. From Table 2, the percentage of accuracy achieved in the training data is 97.92%, while in the test data, it is 96.46%. The results of the performance matrix (Figure 5) show that the precision values are the same for both classes at 96%. At the same time, the highest percentage of recall and f1-score was achieved in the Drone class (Recall = 98%, f1-score = 97%). The



(a) (b)

FIGURE 4. (a) Training and validation accuracy and (b) training and validation loss

TABLE 2. Model evaluation results

Percentage	Accuracy (%)	Loss (%)
X_train, y_train	97.92	6.76
X_test, y_test	96.46	12.63

	precision	recall	f1-score	support
Drone	0.96	0.98	0.97	558
Non-Drone	0.96	0.92	0.94	261
accuracy			0.96	819
macro avg	0.96	0.95	0.96	819
weighted avg	0.96	0.96	0.96	819

FIGURE 5. Performance matrix results

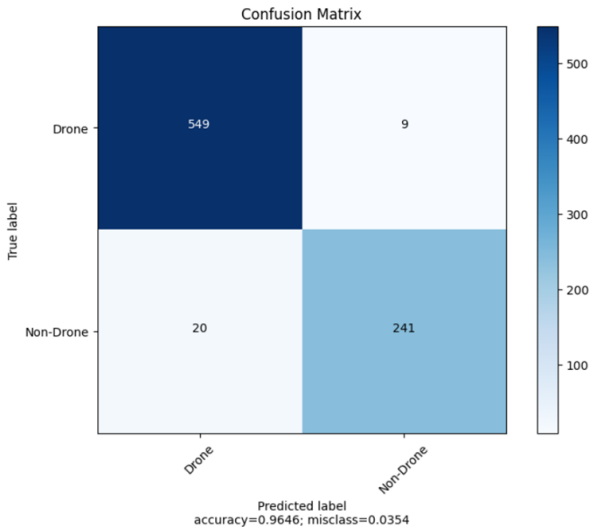


FIGURE 6. Confusion matrix of model evaluation

confusion matrix from the model evaluation (Figure 6) shows that the model made nine times the error of object detection as a non-drone and as a drone 20 times. Suppose this MLP model is compared with machine learning models (SVM, STFT). In that case, the MLP model is more accurate because the machine learning models cannot differentiate UAV sounds and sounds from other flying objects, causing False Positive (FP) and False Negative (FN) results to be dominant [11-13].

5. **Conclusion.** From the results of this experiment, the critical step in making a drone classification model is determining the suitable dataset pre-processing method to produce a reliable and accurate drone classification model. In implementing public area security models, False Negative (FN) values are kept to a minimum because flying objects that are supposed to be drones but are classified as non-drones will endanger the protected area. For future work, research will be continued by developing a model to classify and determine the distance of drones based on sound data with a deep learning model.

Acknowledgment. This work is financially fully supported by Institut Teknologi Telkom Purwokerto, from the Computer Science Postgraduate Program at the Department of Computer Science and Electronics, Gadjah Mada University.

REFERENCES

[1] M. F. Al-Sa'd, A. Al-Ali, A. Mohamed, T. Khattab and A. Erbad, RF-based drone detection and identification using deep learning approaches: An initiative towards a large open source drone database, *Futur. Gener. Comput. Syst.*, vol.100, pp.86-97, 2019.

[2] F. Svanström, C. Englund and F. Alonso-Fernandez, Real-time drone detection and tracking with visible, thermal and acoustic sensors, *Proc. of Int. Conf. Pattern Recognit.*, pp.7265-7272, 2020.

[3] D. A. Tyas, I. Candradewi, Baskara, N. P. Indarto, H. Abdurrahman, Y. Argha, B. A. A. Sumbodo and A. Dharmawan, Horizon detection for UAV attitude based on image processing approach, *ICIC Express Letters*, vol.16, no.12, pp.1249-1258, 2022.

[4] S. R. Ganti, Implementation of detection and tracking mechanism for small UAS, *International Conference on Unmanned Aircraft Systems (ICUAS)*, pp.1254-1260, 2016.

[5] A. Dharmawan, A. Ashari and A. E. Putra, Quadrotor flight stability system with Routh stability and Lyapunov analysis, *AIP Conf. Proc.*, vol.1755, 2016.

[6] S. Park, H. T. Kim, S. Lee, H. Joo and H. Kim, Survey on anti-drone systems: Components, designs, and challenges, *IEEE Access*, vol.9, pp.42635-42659, 2021.

[7] P. Wellig et al., Radar systems and challenges for C-UAV, *Proc. of Int. Radar Symp.*, vol.6, pp.1-8, 2018.

[8] Librosa Development Team, *Feature Extraction*, https://librosa.org/doc/main/feature.html, 2021.

[9] A. Vafeiadis, K. Votis, D. Giakoumis, D. Tzovaras, L. Chen and R. Hamzaoui, Audio content analysis for unobtrusive event detection in smart homes, *Eng. Appl. Artif. Intell.*, vol.89, 103226, 2020.

[10] Y. Wang, F. Fagiani, K. Ho and E. Matson, A feature engineering focused system for acoustic UAV payload detection, *Proc. of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022)*, vol.3, pp.470-475, 2022.

[11] E. Matson, B. Yang, A. Smith, E. Dietz and J. Gallagher, UAV detection system with multiple acoustic nodes using machine learning models, *Proc. of the 3rd IEEE Int. Conf. Robot. Comput. (IRC 2019)*, pp.493-498, 2019.

[12] M. Z. Anwar, Z. Kaleem and A. Jamalipour, Machine learning inspired sound-based amateur drone detection for public safety applications, *IEEE Trans. Veh. Technol.*, vol.68, no.3, pp.2526-2534, 2019.

[13] C. Dumitrescu, M. Minea, I. M. Costea, I. C. Chiva and A. Semenescu, Development of an acoustic system for UAV detection, *Sensors (Switzerland)*, vol.20, no.17, pp.1-27, 2020.