Identification of unmanned aerial vehicles using RF fingerprinting and deep learning networks

Yuriy Kondratenko[1,2,†], Ivan Sova[1,*,†], Oleksiy Kozlov[1,†] and Vitalii Kuzmenko[3,†]

[1] *Petro Mohyla Black Sea National University, 10 68th Desantnykiv st., Mykolaiv, 54003, Ukraine*
[2] *Institute of Artificial Intelligence Problems of MES and NAS of Ukraine, 11/5 Mala Zhytomyrska st., Kyiv, 01001, Ukraine*
[3] *Naval Institute of National University "Odessa Maritime Academy", 8 Diedrichson st., Odesa, 65029, Ukraine*

Abstract

This paper explores key challenges in machine learning classification for optimizing the identification of unmanned aerial vehicles (UAVs) using radio frequency (RF) features and introduces an improved approach based on specific fingerprinting techniques. The study begins by discussing essential data preprocessing steps and feature extraction techniques relevant to RF-based signal analysis for UAVs identification. Several RF feature representations—such as Power Spectral Density (PSD), Short-Time Fourier Transform (STFT), and wavelet-based methods—are tested and compared. The proposed strategy is evaluated on an open-source dataset using different machine learning classifiers. Results indicate that convolutional neural networks (CNNs), when paired with wavelet-based feature extraction, offer the highest classification accuracy, making it possible to differentiate UAV types more effectively. These findings underscore the growing role of deep learning in RF-based UAV identification, with important implications for security and spectrum monitoring.

Keywords

Radio-frequency machine learning, unmanned aerial vehicle, artificial neural network, digital signal processing, Fourier transform, spectral analysis, wavelet transform.

1. Introduction

The growing use of autonomous systems has led to the widespread adoption of unmanned aerial, ground, and marine vehicles in various industries. These technologies play a key role in areas such as surveillance, logistics, and industrial automation. To enhance their performance, researchers have explored AI-based control methods, including fuzzy logic and swarm optimization [1-3]. AI-driven techniques have proven particularly useful in improving drones' and robotic systems' decision-making and navigation capabilities [4-6]. At the same time, as UAVs become more common in both civilian and military settings, the need for effective detection systems has become increasingly important to ensure security and regulatory compliance.

Researchers have explored various machine learning approaches for detecting drones, using different sensing methods. One broad survey discusses a plethora of drone detection strategies [7], including those based on audio analysis [8, 9], computer vision [10-12], and data fusion [13, 14]. Audio-based techniques identify drones by analyzing their unique sound signatures, while computer vision methods rely on deep learning models like YOLO and Mask R-CNN to recognize drones in images or video. Meanwhile, fusion models combine multiple data sources to improve identification accuracy and reliability.

MoMLeT-2025: 7th International Workshop on Modern Machine Learning Technologies, June, 14, 2025, Lviv-Shatsk, Ukraine
* Corresponding author.
† These authors contributed equally.
y_kondrat2002@yahoo.com (Y. Kondratenko); owlvano@gmail.com (I. Sova); kozlov_ov@ukr.net (O. Kozlov); v86651984@gmail.com (V. Kuzmenko)
0000-0001-7736-883X (Y. Kondratenko); 0000-0002-8972-4069 (I. Sova); 0000-0003-2069-5578 (O. Kozlov); 0000-0001-8064-0726 (V. Kuzmenko)

As UAV identification becomes more complex, artificial intelligence methods—such as fuzzy logic—offer a promising way to enhance both accuracy and adaptability. Fuzzy logic has been successfully applied to a range of intelligent decision-making tasks, from optimizing rule-based systems to improving classification techniques [15, 16]. Studies have shown its effectiveness in refining detection models and making them more flexible in dynamic environments.

Radio frequency machine learning (RFML) has recently emerged as a powerful tool for UAV identification, thanks to its ability to recognize electromagnetic signal patterns. Researchers have focused on several key areas, including signal processing and feature extraction [17-19], modulation classification [20, 21], and specific emitter identification [22-24]. More recently, the use of generative adversarial networks (GANs) in RFML has helped strengthen models against adversarial attacks [25, 26]. While these advancements show promise, RFML is still a developing field, with ongoing research needed to refine optimization techniques and enhance real-world performance.

Recent research demonstrates increasing interest in multimodal fusion techniques, particularly those that integrate radio frequency signals with audio data to enhance identification accuracy [27]. It has been suggested that incorporating audio information may improve the robustness of RF datasets to noise and environmental variability. Nevertheless, the practical implementation of such approaches poses significant challenges, particularly in constructing well-balanced, synchronized datasets of radio and audio signals under real-world conditions. Moreover, additional constraints emerge when capturing UAV acoustic signatures in actual operating environments. Despite these difficulties, recent studies have proposed viable ways of integrating audio-based feature extraction techniques for raw RF signal data, which are extended and incorporated in the present work.

In parallel, efforts have emerged to create open-source datasets of drone RF signals, most notably the initiative described in [28], which outlines a systematic framework for the collection and preliminary analysis of raw RF emissions from unmanned aerial vehicles. While this work lays foundational groundwork, it provides only a cursory demonstration of the dataset's applicability to machine learning tasks.

In this study, we expand upon the abovementioned methodologies, and propose a method for constructing features using radio frequency fingerprinting techniques, namely power spectral density, short-time Fourier transform, and wavelets. The resulting feature vectors are applied to several machine learning methods for drone identification. By conducting extensive experiments, we assess the potential of these feature extraction techniques for practical applications in real-world environments.

Proposed approach

This section presents the approach used for the discussed problem. We follow the recommendations provided by [28], and extend the methodology with our proposed RF fingerprinting techniques and deep learning models.

This approach utilizes three RF fingerprinting techniques: power spectral density (PSD), short-time Fourier transform (STFT), and wavelets. These extraction methods rely on physical imperfections in analog components that arise from the device manufacturing process. The effectiveness of convolutional neural networks (CNN) and recurrent neural networks (RNN) is assessed for drone identification.

The proposed approach is shown on Figure 1.

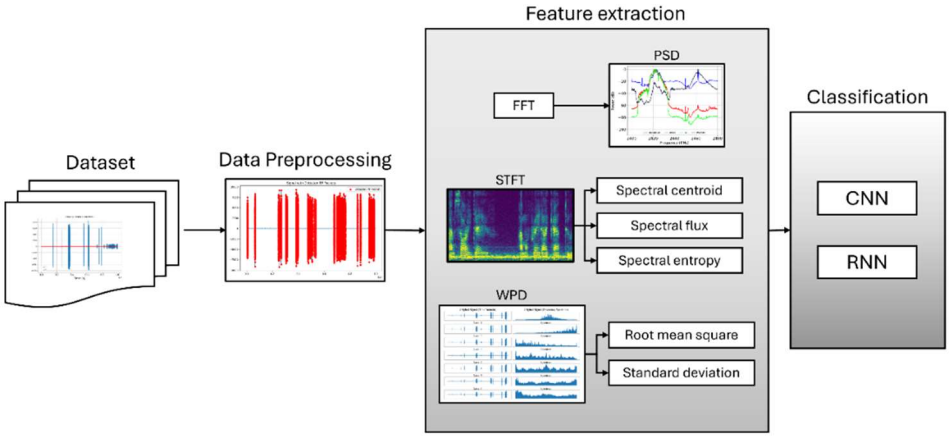

Figure 1: The proposed approach flow diagram.

It includes the following phases:

1. Dataset retrieval and extraction;
2. Data preprocessing;
3. Feature vector extraction;
4. Application of ML-based methods for drone identification.

Dataset

This section describes the RF dataset used to evaluate the effectiveness of the proposed approach — the DroneRF dataset. This is a public dataset provided by [28]. It contains raw RF signal data acquired from two 40MHz receivers that capture the 2.4-2.48 MHz range. The dataset contains background noise data when no drone is present, as well as data captured from three types of drones: Parrot Bebop, Parrot AR, and DJI Phantom. Since two receivers were used, every data sample consists of two signals: low-band and high-band.

Figure 2 depicts the raw sample of the DroneRF dataset. The data packets related to drone activity are clearly observable, as the signal's amplitude drastically increases.

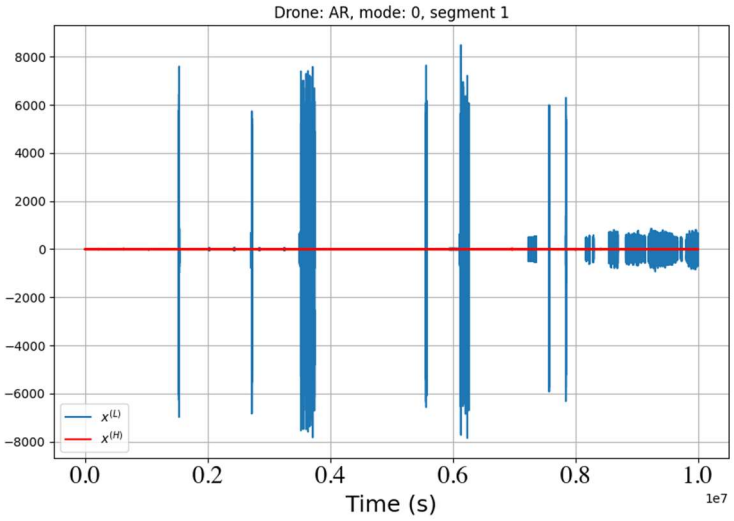

Figure 2: Raw RF signal from the DroneRF dataset.

Table 1 illustrates the summary of the DroneRF dataset. The dataset authors provide 3 experiment levels to work with: drone presence, drone type and drone flight mode. For this research, only the second level experiment level was chosen.

One can pinpoint the class imbalance of the dataset, which is a result of different sample sizes of different classes and of the varying numbers of flight mode classes for each drone. Therefore, data resampling techniques and stratified cross-validation will be used to address this issue.

Table 1
Details of the raw DroneRF dataset

Class	Segments	Samples	Ratio (%)
Background noise	41	820×10^6	18.06%
Bebop	84	1680×10^6	37.00%
AR	81	1620×10^6	35.68%
Phantom	21	420×10^6	9.25%

Data preprocessing

Since the dataset consists of long data packets that include silent intervals, only the RF data packets on drone activity have to be extracted. The main reason for this decision is to obtain meaningful temporal feature information.

In the preprocessing stage, a thresholding technique was used to segment signal data into RF packets. Considering that the background noise packets lack any RF information related to drone activity, thresholding is not applied to them. The resulting packets are subsequently split into equally-sized chunks that contain 4000 samples each. Finally, the dataset is balanced using an undersampling technique via generating centroids based on K-means clustering [29, 30]. This technique is applied only to the majority class (background noise data).

Feature extraction

To train various classifiers, several groups of features have been used: PSD, STFT-based features, and wavelet-based features. All of these feature types are extracted from the DroneRF dataset.

Analyzing UAV signals in the frequency domain is essential for detecting their unique spectral patterns. Methods like spectral analysis, wavelet transforms, and frequency-based filtering help improve classification accuracy by capturing these distinct features. Similar frequency-domain approaches have been successfully applied in other domains to optimize the performance of a complex system [31]. These insights suggest that frequency-domain analysis is a valuable tool for enhancing UAV identification capabilities.

Power Spectral Density

The power spectral density (PSD) describes how the power of a signal or process is distributed across different frequency components [32]. To calculate PSD, one must obtain a frequency-domain representation of a signal using the discrete Fourier transform (DFT) [33], which is calculated as follows:

$$X_f^{DFT}(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j2\pi\frac{k}{N}n}, \tag{1}$$

where $x(n)$ is the time signal at time index n, N denotes the total number of samples in the signal, k is the index of the frequency component ranging from 0 to $N-1$, and j is the imaginary unit.

Consequently, the formula to determine the PSD of the signal would be:

$$X_f^{PSD}(k) = \frac{2}{f_s N} \cdot \left| X_f^{DFT}(k) \right|_{k\geq 0}^2, \tag{2}$$

where f_s describes the sampling frequency of the signal.

According to the Nyquist-Shannon sampling theorem, the sample rate must be at least twice the bandwidth of the signal to avoid aliasing. Ergo, the sampling rate of 80MHz was chosen, due to the fact the signal bandwidth is 40MHz.

Because every signal packet is represented by a low- and high-band component, both of these components should be used for computing the PSD [28]. Therefore, after computing the DFT of both segments, we concatenate the resulting spectral information as such:

$$X_{DFT} = \left[X_{DFT}^{(L)},\ c \cdot X_{DFT}^{(H)} \right], \tag{3}$$

where $X_{DFT}^{(L)}$ and $X_{DFT}^{(H)}$ denote the DFT of the low- and high-band components respectively, and c is the normalization factor. It is calculated as follows:

$$c = \frac{\sum_{q=0}^{Q} X_{DFT}^{(L)} \cdot (M - q)}{\sum_{q=0}^{Q} X_{DFT}^{(H)} \cdot q}, \tag{4}$$

where Q is the number of samples to take from the end of the lower spectra $X_{DFT}^{(L)}$ and the star of the upper spectra $X_{DFT}^{(H)}$, and M is the total number of frequency bins in X_{DFT}. The normalization factor ensures spectral continuity between the two parts of the spectrum, while introducing spectral bias.

Figure 3 shows the average PSD for each of the classes present in the dataset. During the preprocessing stage, PSD vectors are calculated separately for each signal frame.

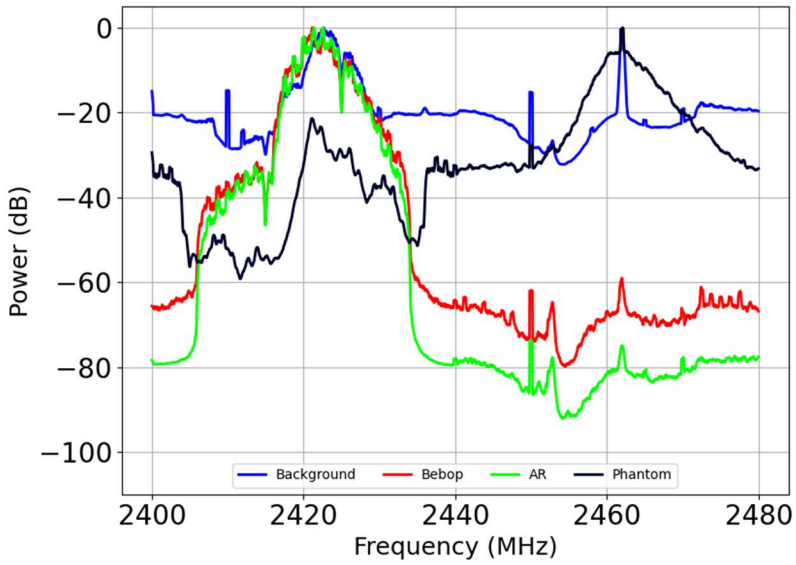

Figure 3: Average power-over-frequency distribution of the DroneRF dataset.

For the proposed approach, the following values have been chosen: $Q = 10, M = 2048$. This way, the value of Q is small enough to combine the two spectral vectors while being large enough to average out any random fluctuations.

Short-time Fourier Transform

While DFT outputs the frequency-domain representation of a signal, analysis of the temporal characteristics proves ineffective. Instead, one may use the short-time Fourier transform (STFT) [34, 35]. It splits the signal information into smaller overlapping segments and applies the Fourier

transform to each segment. The resulting output is the spectrogram: a two-dimensional array that describes the frequency content of a signal over time.

The discrete STFT is given by:

$$X(m,k) = \sum_{n=0}^{N} x(n) \cdot \omega(n-m) \cdot e^{-j2\pi\frac{k}{N}n}, \tag{5}$$

where $X(m,k)$ is the STFT at time index m and frequency index k, $x(n)$ is the original discrete time-signal, $\omega(n)$ is the window function applied to the signal, m is the index of the window and N is the length of the window.

For the purposes of this research, the following values have been chosen: N = 511, and $\omega(n)$ — Hanning window, a classic windowing function used in signal processing. Additionally, the signal is also zero-padded at the end to ensure the signal fits exactly into an integer number of window segments [36], so that all of the signal is included in the output. As a result, the STFT computed with these parameters has 17 timeframes and 256 frequency bins. An example of the generated STFT spectrogram is displayed on Figure 4.

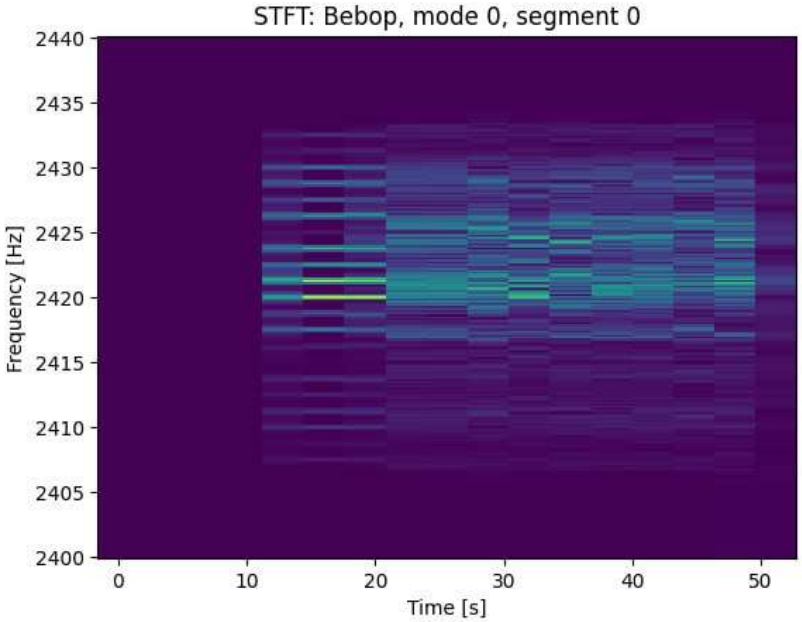

Figure 4: Visualization of the STFT spectrogram of a sample's low-band segment.

Following the STFT transformation of the signal data, instead of using the raw spectrogram itself, several other features are inferred from this representation: spectral centroid, spectral flux and spectral entropy. The following features are then concatenated in a single feature vector.

Spectral centroid is a measure of the "center of mass" of a spectrum and is often used in signal processing, particularly in the analysis of audio signals. It indicates the perceived brightness or timbre of a sound. In simple terms, it tells us where the "center" of the power distribution is in the frequency domain. For a discrete signal, the spectral centroid $C(t)$ at time t can be calculated as:

$$C(t) = \frac{\sum_{f=0}^{F-1} f \cdot |X(t,f)|}{\sum_{f=0}^{F-1} |X(t,f)|}, \tag{6}$$

where t is the time frame, f is the frequency bin index, $X(t,f)$ is the STFT of the signal at time t and frequency bin f, and F is the total number of frequency bins.

Spectral flux is a measure of how much the spectral content of a signal changes between consecutive frames or time windows. It is often used in audio signal processing to detect changes in

the sound over time, such as transitions between different musical notes, chords, or sounds in an audio signal. The spectral flux $S(t)$ at time t is given by:

$$S(t) = \sum_{f=0}^{F-1} (|X(t+1,f)| - |X(t,f)|)^2. \tag{7}$$

Spectral entropy is a measure of the disorder or unpredictability in a signal's frequency content. It quantifies the spread or concentration of power across the frequency spectrum, with higher entropy indicating a more complex or "noisy" spectrum, and lower entropy indicating a more predictable or "peaked" spectrum. The spectral entropy $H(t)$ at time t is calculated based on the equation:

$$H(t) = -\sum_{f=0}^{F-1} p(t,f) \cdot \log\big(p(t,f)\big), \tag{8}$$

$$p(t,f) = \frac{|X(t,f)|^2}{\sum_{f=0}^{F-1} |X(t,f)|^2}, \tag{9}$$

where $p(t,f)$ is the normalized power at frequency bin f and time t.

For the STFT spectrogram with 256 frequency bins and 17 timeframes, the number of time frames of the spectral centroid and spectral entropy is 17 as well. However, the spectral flux vector contains only 16 timeframes, because it calculates the difference between sequential time instants. Hence, the resulting feature vector contains $17 + 16 + 17 = 50$ features.

Wavelets

Wavelet transforms are a mathematical tool used in signal processing to analyze data at multiple scales or resolutions [37]. Unlike Fourier transforms, which decompose a signal into sine and cosine functions with fixed frequencies, wavelet transforms break the signal into components that capture both frequency and time information, allowing for the analysis of signals that are non-stationary (i.e., their frequency content changes over time).

In this work, wavelet packet decomposition (WPD), an extension of discrete wavelet transform (DWT), is used to generate a flexible and detailed multi-resolution analysis of a signal [38]. While the DWT only decomposes the signal into approximation and detail coefficients (low and high-frequency components), the WPD goes further by decomposing both the approximation and detail components at each level. The result is a binary tree structure of wavelet coefficients, where each node represents either approximation or detail coefficients. This binary tree structure allows for a more flexible and complete decomposition, as each node can be further decomposed into finer frequency bands.

Figure 5 shows the visual representation of the WPD: the time- and frequency-domain representations of each decomposition level.

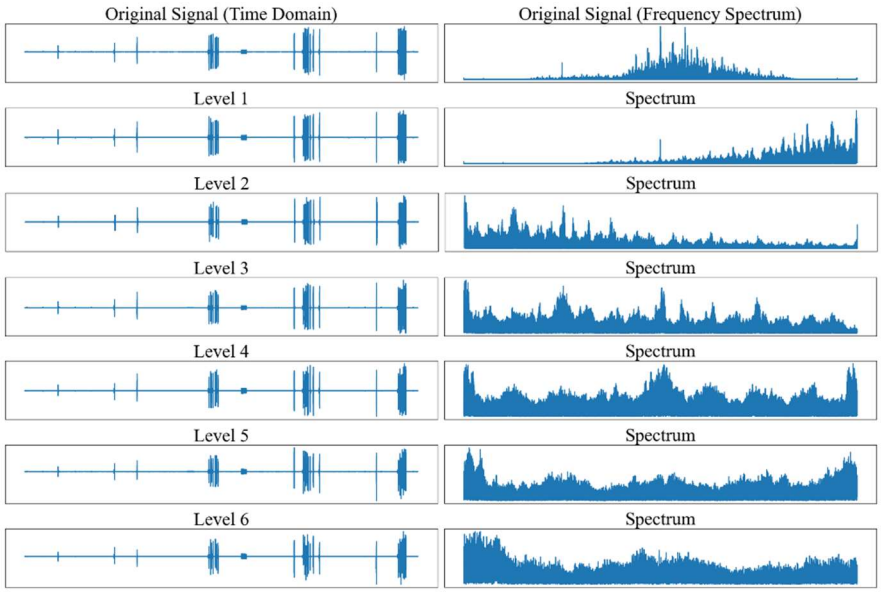

Figure 5: Visual representation of wavelet packet decomposition.

We use 6 levels of decomposition in WPD, resulting in 64 set of coefficients, and apply the FIR approximation of the Meyer wavelet. Finally, each vector is characterized with two features: the root mean squared error (RMS) and the standard deviation (STD) of each coefficient vector. As a result, each feature vector consists of $64 \cdot 2 = 128$ features.

Classification

We investigate two machine-learning-based classification methods used for the identification of UAVs: convolutional neural networks (CNN) and recurrent neural networks (RNN). The effectiveness of these methods is evaluated for the synthesized RF feature vectors.

CNN

Convolutional neural networks (CNNs) are deep learning models designed to process structured grid data, including images and spectrograms. Their ability to automatically extract features and recognize patterns makes them especially useful for tasks like radio frequency fingerprinting.

Figure 6 depicts the CNN used in this research. Its architecture was proposed in [28], which was in turn motivated by LeNet architecture [39]. This classifier is composed of three convolution modules, four dense modules and a SoftMax output layer.

Every convolution module consists of a one-dimensional convolution layer, batch normalization layer, rectified linear unit (ReLU) activation layer and a max pooling layer. Every subsequent layer decreases the dimension sizes: 64, 32, and 16 for the filters, and 11, 5 and 3 for kernels, respectively.

The dense modules receive the flattened input vector from the convolution modules. Every dense module consists of a fully connected (dense) layer, a batch normalization layer, and a parametric rectified linear (PReLU) unit activation layer. A default value of 0.25 was chosen for the PReLU's learnable parameter a. The four fully connected layers incorporate 512, 256, 128 and 4 neurons, respectively.

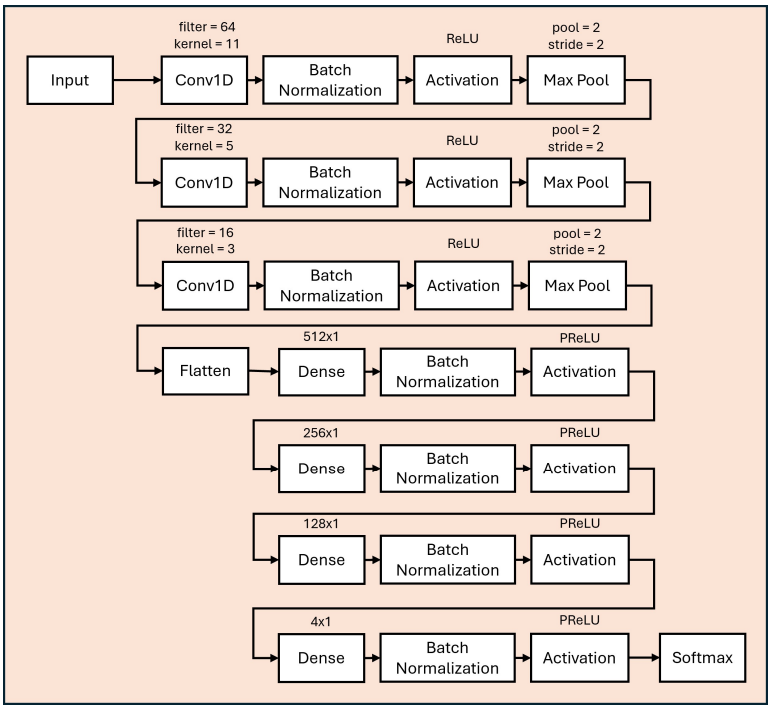

Figure 6: CNN model architecture.

RNN

Recurrent neural networks (RNNs) are specialized for handling sequential data, making them ideal for tasks like time-series analysis or recognizing patterns that change over time [40]. Unlike traditional feedforward networks, RNNs feature recurrent connections that allow information to be passed through from one time step to the next, which helps them capture temporal dependencies in data.

Figure 7 describes the RNN architecture. This model is composed of 2 long short-term memory (LSTM) modules, four dense modules and a SoftMax output layer.

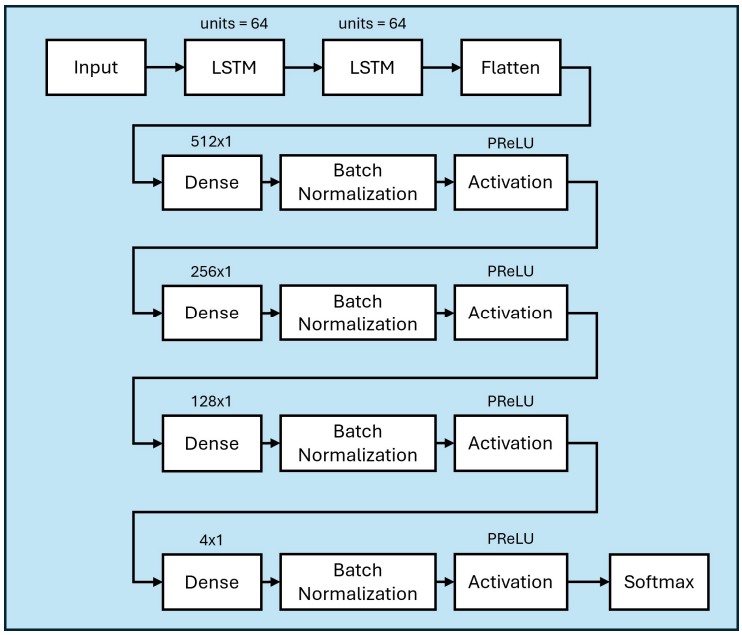

Figure 7: RNN model architecture.

The long short-term memory (LSTM) modules incorporate 64 units each, meaning a number of neurons that comprise the hidden state of the layer.

The dense module pipeline is identical to the CNN model architecture. The fully connected modules receive the flattened input vector from the convolution modules and are composed of a fully connected (dense) layer, a batch normalization layer, and a parametric rectified linear (PReLU) unit activation layer with a default value of 0.25. The four dense layers contain 512, 256, 128 and 4 neurons, respectively.

Model training parameters

Choosing the right training methods and parameters is paramount for achieving the best model performance. In this study, various hyperparameters, including learning rate, batch size, and the number of epochs, were fine-tuned to optimize classification accuracy.

To evaluate the models' performance as precisely as possible, a stratified K-fold cross-validation procedure [41] was integrated in the training pipeline, where $K = 5$. In summary, a provided dataset is split into K folds of training and testing data, which are fit and evaluated separately from each other. Every fold is made by preserving the percentage of samples in each data to address class imbalance.

The following training parameters have been chosen for both CNN and RNN:

- Optimizer: adaptive moment estimation (Adam) [42];
- Loss function: categorical cross entropy;
- Epochs: 50;
- Batch size: 32;
- Learning rate: 0.01.
- Performance metric: accuracy.

2. Results

This section describes the experimental results for the various feature vectors (PSD, STFT and wavelet) and the two ML-based classification methods (CNN, RNN).

As specified in section 2, the CNN classifier consists of three convolution modules, four dense modules (composed of 512, 256, 128 and 4 neurons, respectively), and SoftMax output layer. The RNN classifier, in turn, replaces the three convolution modules with two long short-term memory layers, each containing a state vector of size 64.

The classification performance is validated using a stratified K-fold cross-validation (with $K = 5$). The accuracy is evaluated by using the test subsets generated by each fold's split, and the resulting evaluations are used to generate confusion matrices. This allows us to visualize the average performance of the trained model.

Figures 8, 9 and 10 illustrate the confusion matrices for the PSD, STFT and wavelet datasets, respectively, using CNN(a) and RNN(b) model architectures.

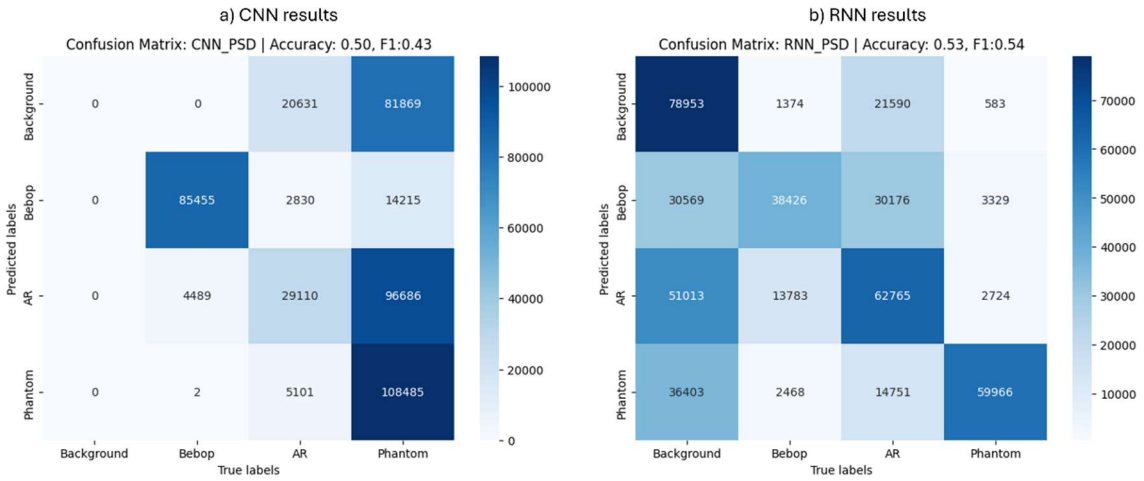

Figure 8: PSD: CNN (**a**) and RNN (**b**) results.

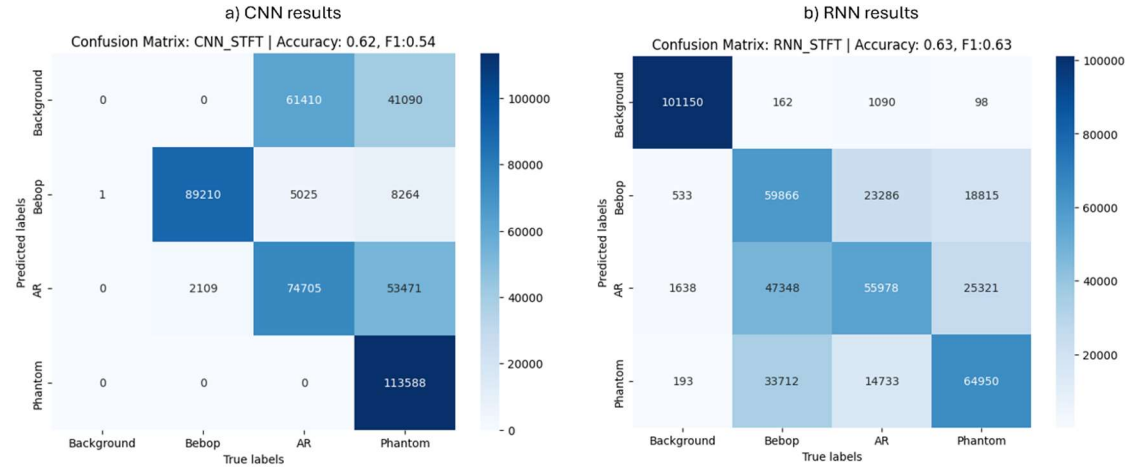

Figure 9: STFT: CNN (**a**) and RNN (**b**) results.

Additionally, Table 2 offers a comparative analysis of the performance metrics obtained during the training process. The accuracy and F1-score metrics are computed for each obtained test evaluation vector.

It is evident from Figures 8-10 and Table 2 that the CNN using the feature vector computed from WPD coefficients achieves the best results with 97% accuracy and an F1-score of 0.97. In comparison, the PSD dataset shows the worst results: the CNN is unable to correctly determine the background noise packets, while the RNN overfits on them. The STFT-based dataset exhibits slightly better results; however, in this case, the CNN still underfits on the background noise, while the RNN fails to ascertain enough features to distinguish different drone types from each other.

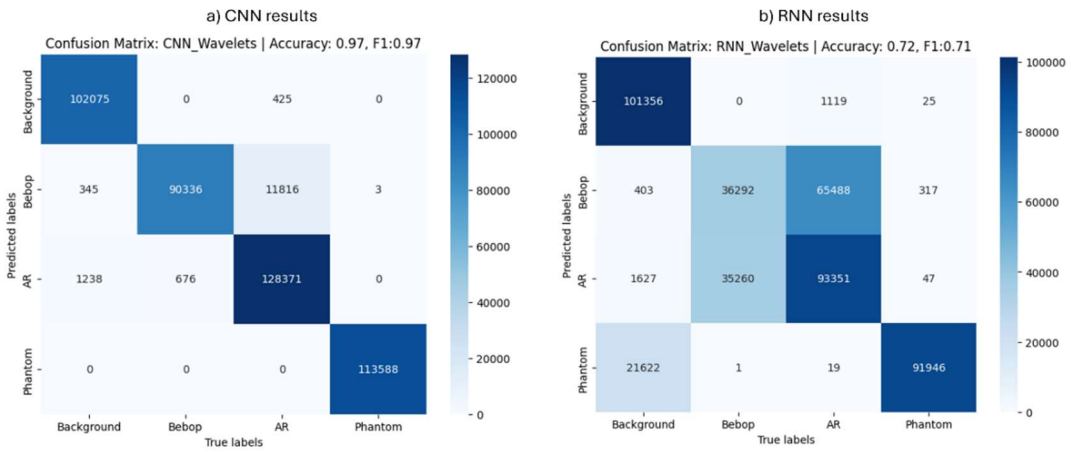

Figure 10: Wavelets: CNN (**a**) and RNN (**b**) results.

Table 2
Analysis of performance metrics of the ML-based classification methods

Dataset types	Performance metrics per model			
	CNN		RNN	
	Accuracy (%)	F1	Accuracy (%)	F1
PSD	50%	0.43	53%	0.54
STFT-based	62%	0.54	63%	0.63
Wavelet-based	**97%**	**0.97**	72%	0.71

3. Conclusion

This study conducts research and comparative analysis on machine learning classification methods for optimizing the identification of unmanned aerial vehicles using radio frequency fingerprinting techniques, namely power spectral density, short-time Fourier transform, and wavelets. Specifically, the proposed approach has been evaluated with an open-source dataset (DroneRF), and compared against different ML-based classification methods using various feature vectors (PSD, STFT-based and wavelet-based).

The results demonstrate the effectiveness of convolutional neural networks for RF-based machine learning using wavelet-based feature extraction. The proposed CNN shows a high classification accuracy of 97%, effectively determining the drone type. The classifier demonstrates a 7% improvement in classification accuracy over the model presented in [27], despite both approaches utilizing wavelet-based feature vectors. This improvement is primarily attributed to careful tuning of the training parameters, wavelet packet decomposition settings, and the CNN architecture, which collectively enhanced the model's ability to extract and learn discriminative features for UAV identification.

These findings highlight the potential of deep learning models, particularly CNNs, in enhancing the accuracy and reliability of UAV identification through RF fingerprinting. The ability to distinguish between different drone types with high precision is crucial for applications such as airspace security, unauthorized drone identification, and spectrum monitoring. Moreover, the use of wavelet-based feature extraction proves to be a significant factor in improving classification performance, reinforcing its viability as a preprocessing technique for RF-based machine learning tasks.

Future research could explore the integration of additional deep learning architectures, such as hybrid models, or augmentation of the data preprocessing stage to optimize the feature vectors, or further optimization of the RNNs in the context of radio frequency machine learning. Additionally, testing the proposed model on real-time or larger-scale datasets could provide insights into its robustness under varying environmental conditions and signal interference. Expanding this work could lead to the development of more efficient and scalable UAV identification systems, contributing to advancements in security and autonomous air traffic management.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT and Grammarly in order to: Paraphrase and reword, Improve writing style, Grammar and spelling check, Plagiarism detection. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] Y.P. Kondratenko, et al., Bio-inspired optimization of fuzzy control system for inspection robotic platform: comparative analysis of hybrid swarm methods, in: Modern Machine Learning Technologies Workshop, Proceedings of the Modern Machine Learning Technologies Workshop (MoMLeT 2024), Lviv-Shatsk, Ukraine, CEUR-WS, Vol-3711, 2024, pp. 109-123. https://ceur-ws.org/Vol-3711/paper7.pdf.

[2] O.V. Kozlov, Optimal Selection of Membership Functions Types for Fuzzy Control and Decision Making Systems, in: Proceedings of the 2nd International Workshop on Intelligent Information Technologies & Systems of Information Security with CEUR-WS, Khmelnytskyi, Ukraine, IntelITSIS 2021, CEUR-WS, Vol-2853, 2021, pp. 238-247. https://ceur-ws.org/Vol-2853/paper22.pdf.

[3] O. V. Kozlov, Y. P. Kondratenko, O. S. Skakodub, Information technology for parametric optimization of fuzzy systems based on hybrid grey wolf algorithms, SN Comput. Sci. 3.6 (2022). doi:10.1007/s42979-022-01333-4.

[4] O. Skakodub, et al., Optimization of Linguistic Terms' Shapes and Parameters: Fuzzy Control System of a Quadrotor Drone, in: 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2021, pp. 566-571, doi: 10.1109/IDAACS53288.2021.9660926.

[5] Kozlov O. et al. Swarm optimization of the drone's intelligent control system: comparative analysis of hybrid techniques, Proceedings of the 12th International Conference "Information Control Systems & Technologies 2024" (ICST 2024), Odesa, Ukraine, CEUR-WS, Vol-3790, 2024, 1-12. https://ceur-ws.org/Vol-3790/paper01.pdf.

[6] I. Sidenko, et al. Machine Learning for Unmanned Aerial Vehicle Routing on Rough Terrain. In: Hu, Z., Dychka, I., He, M. (eds) Advances in Computer Science for Engineering and Education VI. ICCSEEA 2023. Lecture Notes on Data Engineering and Communications Technologies, Vol. 181, Springer, Cham, 2023, pp. 626–635. DOI: 10.1007/978-3-031-36118-0_56.

[7] Z. Tan, M. Karakose, Survey and comparative study for drone detection using deep learning, in: 2022 international conference on data analytics for business and industry (ICDABI), IEEE, 2022. doi:10.1109/icdabi56818.2022.10041658.

[8] J. Fang, Y. Li, P. N. Ji, T. Wang, Drone detection and localization using enhanced fiber-optic acoustic sensor and distributed acoustic sensing technology, J. Light. Technol. (2022) 1–10. doi:10.1109/jlt.2022.3208451.

[9] C. A. Ahmed, F. Batool, W. Haider, M. Asad, S. H. Raza Hamdani, Acoustic based drone detection via machine learning, in: 2022 international conference on IT and industrial technologies (ICIT), IEEE, 2022. doi:10.1109/icit56493.2022.9989229.

[10] R. Valaboju, Vaishnavi, C. Harshitha, A. R. Kallam, B. S. Babu, Drone detection and classification using computer vision, in: 2023 7th international conference on trends in electronics and informatics (ICOEI), IEEE, 2023. doi:10.1109/icoei56765.2023.10125737.
[11] D. T. Wei Xun, Y. L. Lim, S. Srigrarom, Drone detection using YOLOv3 with transfer learning on NVIDIA Jetson TX2, in: 2021 second international symposium on instrumentation, control, artificial intelligence, and robotics (ICA-SYMP), IEEE, 2021. doi:10.1109/ica-symp50206.2021.9358449.
[12] A. S. Mubarak, M. Vubangsi, F. Al-Turjman, Z. S. Ameen, A. S. Mahfudh, S. Alturjman, Computer vision based drone detection using mask R-CNN, in: 2022 international conference on artificial intelligence in everything (AIE), IEEE, 2022. doi:10.1109/aie57029.2022.00108.
[13] T. R. Lenhard, A. Weinmann, S. Jäger, T. Koch, YOLO-Feder fusionnet: A novel deep learning architecture for drone detection, in: 2024 IEEE international conference on image processing (ICIP), IEEE, 2024, pp. 2299–2305. doi:10.1109/icip51287.2024.10647355.
[14] J. Kim, D. Lee, Y. Kim, H. Shin, Y. Heo, Y. Wang, E. T. Matson, Deep learning based malicious drone detection using acoustic and image data, in: 2022 sixth IEEE international conference on robotic computing (IRC), IEEE, 2022. doi:10.1109/irc55401.2022.00024.
[15] O.V. Kozlov, Information Technology for Designing Rule Bases of Fuzzy Systems using Ant Colony Optimization, International Journal of Computing 20(4) (2021) 471-486. https://www.computingonline.net/computing/article/view/2434.
[16] L. Congxiang, O. Kozlov, G. Kondratenko, A. Aleksieieva, Decision support system for maintenance planning of vortex electrostatic precipitators based on iot and AI techniques, in: Research tendencies and prospect domains for AI development and implementation, River Publishers, New York, 2024, pp. 87–105. doi:10.1201/9788770046947-5.
[17] N. Soltanieh, Y. Norouzi, Y. Yang, N. C. Karmakar, A review of radio frequency fingerprinting techniques, IEEE J. Radio Freq. Identif. 4.3 (2020) 222–233. doi:10.1109/jrfid.2020.2968369.
[18] X. Yu, H. Zeng, Y. Tian, L. Guo, P. Ye, M. Wang, C. Liu, Research on machine learning and signal processing of IQ signals, in: 2023 IEEE 16th international conference on electronic measurement & instruments (ICEMI), IEEE, 2023. doi:10.1109/icemi59194.2023.10270324.
[19] S. Aburakhia, A. Shami, G. K. Karagiannidis, On the intersection of signal processing and machine learning: A use case-driven analysis approach, Preprint, 2024. arXiv. doi:10.48550/arXiv.2403.17181.
[20] M. S. Ul Qamar, M. Awais Akhter, R. Nawaz, Automatic modulation recognition using convolutional neural networks, in: 2022 19th international bhurban conference on applied sciences and technology (IBCAST), IEEE, 2022. doi:10.1109/ibcast54850.2022.9990255.
[21] K. Jung, J. Woo, S. Mukhopadhyay, On-chip acceleration of RF signal modulation classification with short-time fourier transform and convolutional neural network, IEEE Access (2023) 1. doi:10.1109/access.2023.3344175.
[22] M. Sliti, M. Garai, Drone Detection and Classification approaches based on ML algorithms, in: 2023 28th Asia Pacific Conference on Communications (APCC), IEEE, 2023. doi:10.1109/apcc60132.2023.10460666.
[23] S. Al-Emadi, F. Al-Senaid, Drone detection approach based on radio-frequency using convolutional neural network, in: 2020 IEEE international conference on informatics, iot, and enabling technologies (iciot), IEEE, 2020. doi:10.1109/iciot48696.2020.9089489.
[24] S. Mandal, U. Satija, Time-Frequency multi-scale convolutional neural network for rf-based drone detection and identification, IEEE Sens. Lett. (2023) 1–4. doi:10.1109/lsens.2023.3289145.
[25] D. Roy, T. Mukherjee, M. Chatterjee, Machine learning in adversarial RF environments, IEEE Commun. Mag. 57.5 (2019) 82–87. doi:10.1109/mcom.2019.1900031.
[26] K. Merchant, B. Nousain, Securing IoT RF fingerprinting systems with generative adversarial networks, in: MILCOM 2019 - 2019 IEEE military communications conference (MILCOM), IEEE, 2019. doi:10.1109/milcom47813.2019.9020907.
[27] A. Frid, Y. Ben-Shimol, E. Manor, S. Greenberg, Drones detection using a fusion of RF and acoustic features and deep neural networks, Sensors 24.8 (2024) 2427. doi:10.3390/s24082427.

[28] M. F. Al-Sa'd, A. Al-Ali, A. Mohamed, T. Khattab, A. Erbad, RF-based drone detection and identification using deep learning approaches: An initiative towards a large open source drone database, Future Gener. Comput. Syst. 100 (2019) 86–97. doi:10.1016/j.future.2019.05.007.
[29] J. Han, M. Kamber, J. Pei, Data preprocessing, in: Data mining, Elsevier, 2012, pp. 83–124. doi:10.1016/b978-0-12-381479-1.00003-4.
[30] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, J. Heming, K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data, Inf. Sci. (2022). doi:10.1016/j.ins.2022.11.139.
[31] Y.P. Kondratenko, O.V. Korobko, O.V. Kozlov, Frequency Tuning Algorithm for Loudspeaker Driven Thermoacoustic Refrigerator Optimization, in: K. J. Engemann, A. M. Gil-Lafuente, J. M. Merigo (Eds.), Lecture Notes in Business Information Processing, volume 115 of Modeling and Simulation in Engineering, Economics and Management, Springer-Verlag, Berlin, Heidelberg: 2012, pp. 270-279. https://doi.org/10.1007/978-3-642-30433-0_27.
[32] C. D. Hayes, Power spectral density analysis, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, 1966.
[33] J. W. Cooley, V. Cizek, Discrete fourier transforms and their applications., Math. Comput. 50.182 (1988) 643. doi:10.2307/2008635.
[34] E. Sejdić, I. Djurović, J. Jiang, Time–frequency feature representation using energy concentration: An overview of recent advances, Digit. Signal Process. 19.1 (2009) 153–183. doi:10.1016/j.dsp.2007.12.004.
[35] E. Jacobsen, R. Lyons, The sliding DFT, IEEE Signal Process. Mag. 20.2 (2003) 74–80. doi:10.1109/msp.2003.1184347.
[36] Scipy.signal.stft — scipy v1.13.1 manual. URL: https://docs.scipy.org/doc/scipy-1.13.1/reference/generated/scipy.signal.stft.html.
[37] C. K. Chui, An Introduction to Wavelets, Elsevier Science & Technology Books, 2016.
[38] A. N. Akansu, R. A. Haddad, Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets, 2nd. ed., Academic Press, 2000.
[39] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521.7553 (2015) 436–444. doi:10.1038/nature14539.
[40] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE conference on computer vision and pattern recognition (CVPR), IEEE, 2016. doi:10.1109/cvpr.2016.90.
[41] S. Widodo, H. Brawijaya, S. Samudi, Stratified K-fold cross validation optimization on machine learning for prediction, Sinkron 7.4 (2022) 2407–2414. doi:10.33395/sinkron.v7i4.11792.
[42] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, Preprint, 2014. arXiv. doi:10.48550/arXiv.1412.6980.