Ensemble Deep Learning for Sustainable Multimodal UAV Classification

James McCoy, Atul Rawal, Danda B. Rawat, *Senior Member, IEEE*, and Brian M. Sadler, *Life Fellow, IEEE*

***Abstract*— Unmanned aerial vehicles (UAVs) have increasingly shown to be useful in civilian applications (such as agriculture, public safety, surveillance) and mission critical military applications. Despite the growth in popularity and applications, UAVs have also been used for malicious purposes. In such instances, their timely detection and identification has garnished rising interest from government, industry and academia. While much work has been done for detecting UAVs, there still exist limitations related to the impact of extreme environmental conditions and big dataset requirements. This paper proposes a novel ensemble deep learning framework that has hybrid synthetic and deep features to detect unauthorized or malicious UAVs by using acoustic, image/video and wireless radio frequency (RF) signals for robust UAV detection and classification. We present the performance evaluation of the proposed approach using numerical results obtained from experiments using acoustic, image/video and wireless RF signals. The proposed approach outperforms the existing related approaches for detecting malicious UAVs.**

***Index Terms*— Ensemble deep learning, multi-modal UAV classification, UAV detection, machine learning, CNN.**

I. Introduction

ADVANCES in inexpensive and widely available commercial and consumer grade unmanned aerial vehicles (UAVs) or "drones" have expanded their overall use. These rapid developments and deployment of wide spread applications of UAVs have led to an increased need to address various concerns related to public safety and privacy. It is critical detect and identify malicious UAVs flying in unauthorized or restricted areas [1]. Malicious UAVs can be considered as those that are outfitted with explosive payloads or that are used to collect data in restricted territory. Low altitude flights may allow them to operate in restricted areas without triggering traditional air-space security measures. Restricted air-spaces can be understood as areas (land or water/ocean) above which unauthorized UAVs are not permitted under certain conditions. To illustrate the overall concept, a typical system model is depicted in Fig. 1 where we can see UAVs, UAV-restricted area and UAV-monitoring area.

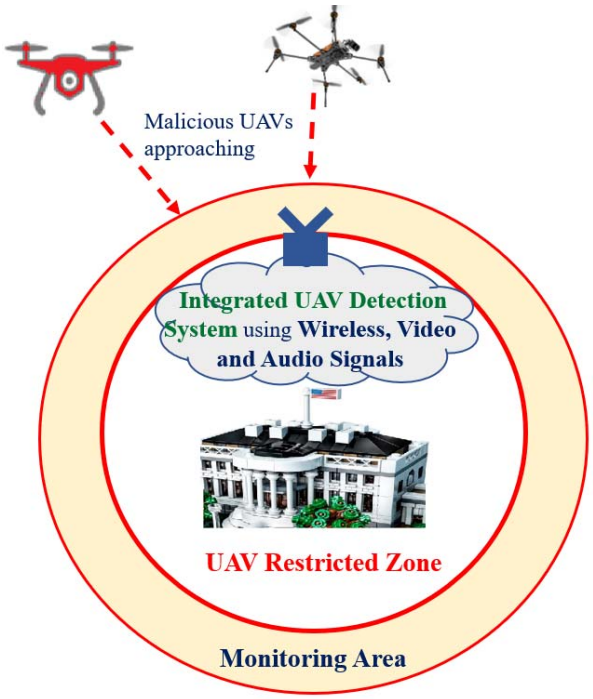

Fig. 1. System model for detecting malicious UAVs with UAV restricted zone and monitoring area.

The need to improve early detection of malicious UAVs with high confidence has grown with urgency as these devices are being increasingly used for malicious purposes. To improve overall UAV detection, researchers have sought to leverage audio, video, thermal, and radio frequency (RF) sensing. Despite advances in various methodologies, each sensing modality maintains its own set of advantages and limitations. For example, the performance of video and thermal based detection techniques are impacted by adverse weather conditions [2]. Also, the acoustic sound from the UAV's motor fan is useful when differentiating UAVs from other flying objects such as birds, aeroplanes, and other aerial objects, they require additional audio- or image/video-based detectors to capture surrounding sounds/scene for classification [3].

With modern UAVs capable of carrying higher payloads as well as having significant communications and computational resources, their usage has been widely applied to malicious

Manuscript received 17 November 2021; revised 9 March 2022; accepted 22 April 2022. Date of publication 13 May 2022; date of current version 29 November 2023. This work was supported in part by the DoD Center of Excellence in AI and Machine Learning (CoE-AIML), Howard University, with the U.S. Army Research Laboratory, under Contract W911NF-20-2-0277. The Associate Editor for this article was Z. Lv. *(Corresponding author: Danda B. Rawat.)*

James McCoy, Atul Rawal, and Danda B. Rawat are with the Department of Electrical and Computer Science, Howard University, Washington, DC 20059 USA (e-mail: db.rawat@ieee.org).

Brian M. Sadler is with the U.S. Army Research Laboratory, Adelphi, MD 20783 USA.

Digital Object Identifier 10.1109/TITS.2022.3170643

activities. For example, for cyber-eavesdropping, to transport illegal narcotics across borders, flying into sensitive or populated areas where their presence can cause potentially damaging accidents. This motivates the development of robust and efficient models to detect and identify malicious UAVs. To mitigate many of the shortcomings of each individual sensing modality, we propose an ensemble deep learning approach. The contributions of this paper include:

1) Design and evaluate ensemble learning approach for detecting malicious UAVs in restricted area using multi-modal datasets such as audio dataset, image/video dataset and RF signals.
2) Leverage multi-modal datasets with three-layered CNN model consisting of two convolutional layers and one fully connected layer for a multi-class ensemble model for UAV detection.
3) Compare the proposed ensemble learning approach with the related state of art approaches for UAV detection in restricted areas.

The remainder of the paper is organized as follows; Section II covers prior research related to UAV detection and identification. Section III presents the Convolutional Neural Network Architectures for our study. Section IV presents the UAV detection methodology. Section V presents the proposed approach. Section VI discusses the dataset and experimental setup The metrics and results are given in Section VI and Section VII respectively. Section VIII concludes the paper.

II. Background and Related Work

UAVs can be detected via several UAV related signals, such as thermal images, audio from the UAVs, and wireless RF signals [4]. Despite advances made in applying mathematical models to improve collective identification of malicious UAVs, these signals are vulnerable to environmental disruptions. Table I provides an overview of the limitations of current UAV detection approaches (next page). We present related work for different UAV datasets in the following subsections.

A. Image-Based Feature Classification

In [5], the authors proposed a technique to detect and track flying devices in the environment using a monocular camera. The study focused on a vision-based solution to detect and track the UAVs by cooperative flying nodes autonomously within the wide range of deployment. The proposed system leveraged template matching and morphological filtering to detect the flying node. The key challenges that the authors addressed are poor illumination conditions, target scale, and background. To overcome these challenges an innovative vision-based architecture for UAV detection and tracking was proposed in this study.

The work in [6] also proposed a vision based UAV detection system that leverages a CNN. Authors formulated the detection problem as two different tasks; the first task was to localize the target object and the background was removed in an input stream. The second task was to then further classify the detected object into three different classes such as birds, airplanes, and UAVs. Foreground object detection was done using the background removal method and the input stream was taken as static background in this study. A CNN was used for classification. A pre-trained architecture of MobileNetV2 was used for the classification consisting of three different layers: convolutional, pooling, and fully connected.

Authors in [7] proposed a system that employs the Haar feature cascade classifier to detect UAVs and then leveraged a CNN to recognize them. The proposed model incorporates a three-layered CNN model by leveraging two convolutional layers as well as a fully connected layer. The fully connected layer uses 30% dropout and the Adam optimizer was employed while training the model. The dataset used for the training was manually extracted from Google images that contained the distorted UAVs. The proposed system achieved 89% detection accuracy and 91.6%identification accuracy.

B. Audio-Based Feature Classification

The work in [8] proposed a system that leverages a Hidden Markov Model (HMM) for classification and identification of UAVs by extracting the patterns from audio. Experiments were carried out to record the acoustic variations from two UAVs. In [9], authors proposed a passive sensor that consists of a video camera and 120 microphone array where the proposed algorithm determines the audio levels a real-time environment. This extracted audio signal can be used to identify audio signatures and determine the source type. The acoustic camera can produce fully spherical acoustic images which is used to detect and track audio sources located in any direction. The tracking module then takes spherical acoustic images as an input and tracks objects from frame to frame even when their audio is low.

A study in [10] investigated the effectiveness of different deep neural networks including Recurrent Neural Network (RNN) and Convolutional Recurrent Neural Network (CRNN) to detect and identify malicious UAVs based on their audio. The authors have created audio samples of more than 1000 UAVs and augmented this data with different random noisy data to evaluate the feasibility of the system in a real time noisy environment.

Another study proposed a UAV detection system that leverages the acoustic characteristics of flying nodes to detect them [11]. This study uses machine learning models, i.e., CNN and SVM, for detection of the nodes along with empirical deployment configuration. Acoustic features include mel-frequency cepstral coefficients (MFCC), and short-time Fourier transform (STFT) for training. The SVM and CNN were trained, and experiments tested the ability to find the path of a flying UAV. To minimize the effect of noise and blind spots, sensing nodes were deployed in four different configurations and the best dataset was chosen for further processing.

C. Radio Frequency-Based Feature Classification

A UAV intrusion detection system was proposed in [12] that uses a RF signal detection technique. In this method time-frequency energy distribution features are extracted by converting single transient control and video signals using a

TABLE I
PROPERTIES OF VIDEO, AUDIO AND RF/WIRELESS BASED DETECTION APPROACHES

Detection Approach	Target Principal	Characteristics and Drawback
UAV detection using Video	UAV image and motion detection	Hard to differentiate between UAVs and other flying objects. Line of sight with range less than 350 ft needed.
UAV detection using Audio	Sound at high frequency around 40 kHz	Maynot work in noisy areas. Typical range is 25 ft to 30 ft
UAV detection using RF/Wireless Signal	UAV Control and Video wireless signals	May not work in high altitude scenarios and wireless signal quality at received depends on transmit power. Typical range is less than 1400 ft

short time Fourier transform (STFT). Authors applied principal component analysis (PCA) to minimize the dimensions of the RF feature vector. A support vector machine is trained using the remapped UAV RF signal feature data, and for estimation of the existence and number of interfering UAVs a k-nearest neighbor (kNN) algorithm was used. Authors in [13] proposed a solution for UAV detection using a CNN which is based on the RF emitted during communication between a UAV and its controller. A multi-layer CNN is employed, and to avoid over-fitting a dropout layer with 20% dropout rate is added. Experimental results showed that CNN model outperformed other NN architecture models discussed in literature.

Work in [14] proposed the detection and classification of signals coming from UAV controllers using Wi-Fi and Bluetooth sources passively. The detection of interference ensures the robustness of proposed methodology against false alarms and missed target detection. The interference signals are identified with a multistage detector for estimating the bandwidth and modulation features of the detected signals. The identification of signals is carried out via machine learning based classification techniques using RF fingerprints. Different types of UAVs are classified using an energy transient signal, as it is more robust to different types of noise.

In one hand, robust and efficient models to detect and identify malicious UAVs is urgently needed. On the other hand, despite advances in various methods that use different UAV datasets, each method has its own set of advantages and limitations. In this setup, our goal is to design and evaluate the ensemble deep learning approach to mitigate many of the shortcomings of each individual models to enhance the multimodal UAV classification.

III. CONVOLUTIONAL NEURAL NETWORK (CNN) ARCHITECTURE

A convolutional neural network (CNN) is a deep learning methodology that leverages a series of convolutional and pooling functions to extract feature classes from structured data samples. We consider a binary classification problem and use a CNN as a baseline. CNN models operate by passing data through a series of filtered convolutional layers. These layers are structured to make explicit assumptions on the inputs to improve the overall predictive values of the sequenced classifiers. The central block for the CNN is the convolutional layer. This layer represents a group of trainable filters which operate as parameters of this layer. The goal is to extract features from the input data while preserving the spatial relationships between the terms. Each model filter operates within the initial input window. The sliding window sequentially moves the input image for the forward propagation phase of each model layer until the last output layer. A convolutional layer is useful as it can represent an input, output and hidden layer within a model. We can represent the convolution layer for our 1D CNN model as:

$$\mathcal{F} \times \mathcal{I}(x) = \sum_{i=0}^{n} \mathcal{F}(i)\mathcal{I}(x-i) \tag{1}$$

where $\mathcal{F}$ operates as a function which represents layered filters applied to the data samples that are represented as single dimensional input as $\mathcal{I}$. The Convolution layer for feature extraction of our image and audio samples can be represented as:

$$\mathcal{F} \times \mathcal{I}(x, y) = \sum_{i=0}^{n} \sum_{j=0}^{n} \mathcal{F}(i, j)\mathcal{I}(x-i, y-j) \tag{2}$$

The equation (2) takes the horizontal and vertical of the features of the inputs x and y which are assessed based upon the layered filters represented as $\mathcal{F}$.

A. CNN Layers and Functions

A typical neural network has a collection of different hidden network layers which apply useful actions to improve classification of the input signal. The sub-layers of the CNN aim to reduce the spatial size of the vector through gradual reduction of the network parameters. The three hidden layers and functions used in our CNN models are: Pooling, Flattening, and Dropout layers.

1) Pooling Layer: One of the important CNN layers is pooling layer. The pooling layer allows us to reduce the dimensionality of the processed data by removing unimportant information from the input data. This layer partitions the input image progressively to reduce the number of associated parameters through input-derived window operations which collapse the values (such as using feature pooling and max-pooling [15]). The max- pooling layer down samples the input by dividing into rectangular pooling regions, then computes the maximum of each region. An average pooling layer differs as it performs down sampling by dividing the input into rectangular pooling regions, then computing the average values of each region.

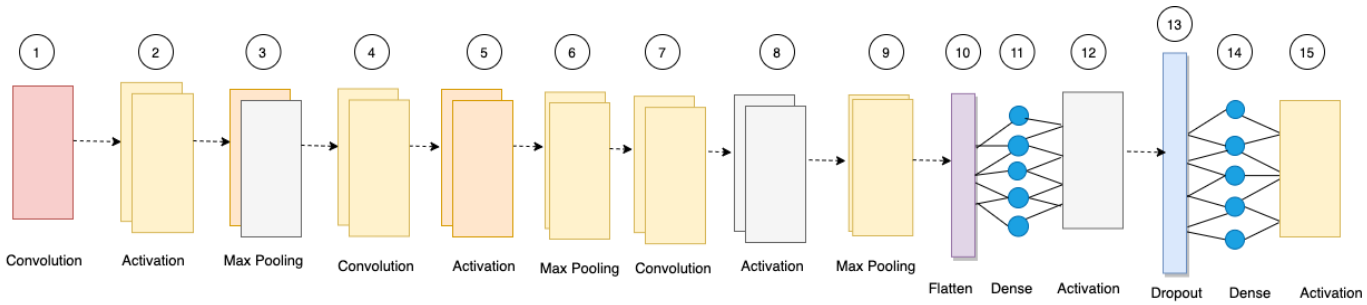

Fig. 2. Block diagram for 2D CNN (with 3 dimensional input).

2) Flattening: To reorganize data into a dimensional vector, flattening layer is used which connects CNN layer and fully connected layer. The flatten layer processes the multi spatial dimensional input data into one dimensional vector suitable to process in CNN.

3) Dropout Layer: A dropout layer in CNN randomly removes the input data elements by setting them to zero with a given probability and scales the remaining data elements This function of the dropout layer in CNN helps to change the network architecture for iterations which essentially prevents the CNN from over fitting.

4) Fully Connected Layer: Fully connected layer is a type of CNN layer which uses a linear operation where every input is connected to every output by a weight. Within a fully connected layer, all of the connected layer neurons are connected to all other neurons in the previous layer. Furthermore, connected layer combines all of the features learned by the previous layers across the dataset or image to identify the larger patterns in the dataset or image.

B. 2-D CNN

For classification of the image and audio samples we created a custom 2D CNN model. We present the block diagram for our 2D model in Fig. 2. The model summary for the image classifier is found in Table II and Table III shows the model summary for the Audio classifier. The summaries for the 2D CNNs contain the CNN layer description and the output shape.

The two-dimensional CNN model used for classification of image and audio samples contained a total of 15 layers. Excluding the input and output layers, there were 13 hidden layers. Those 13 hidden layers consisted of four Convolution layers, three max pooling layers, four rectified linear unit (ReLU) activation functions, two dense layers, a single dropout layer, and a single flatten function. The 2nd dense layer used the "sigmoid" activation function to improve the accuracy of the output layer. The size of the input image and audio samples were 242 × 241 dimensions.

1) Forward-Propagation and Back-Propagation in 2-D CNN: The convolution layer computation can be expressed as [16]

$$x^{l}_{i,j,k} = \sum_{q}\sum_{r}\sum_{s} w^{l-1}_{i,j,k} y^{l-1}_{i+q,j+r,k+s} + b^{l}_{i,j,k} \tag{3}$$

TABLE II
2D IMAGE CNN

Layer	Output Shape
1. Convolution	242x241x1
2. Activation	240x239x32
3. Max Pooling	240x239x32
4. Convolution	120x119x32
5. Activation	118x117x32
6. Max Pooling	118x117x32
7. Convolution	59x58x32
8. Activation	57x56x32
9. Max Pooling	57x56x64
10. Flatten	28x28x64
11. Dense	50176
12. Activation	64
13. Dropout	64
14. Dense	64
15. Activation	1

TABLE III
2D AUDIO CNN

Layer	Output Shape
1. Convolution (Input)	240x239x32
2. Activation	240x239x32
3. Max Pooling	120x119x32
4. Convolution	118x117x32
5. Activation	118x117x32
6. Max Pooling	59x58x32
7. Convolution	57x56x64
8. Activation	57x56x64
9. Max Pooling	28x28x64
10. Flatten	50176
11. Dense	64
12. Activation	64
13. Dropout	64
14. Dense	1
15. Activation	1

and

$$y^{l-1}_{i,j,k} = \sigma\left(x^{l-1}_{i,j,k}\right) \tag{4}$$

where $y^{l-1}_{i,j,k}$ represents the output of the layer $l-1$, b represents the bias, $w^{l-1,f}$ represents the weight at the layer $l-1$ and q, r, s and i, j, k, respectively represent the 3D coordinates of the weight of the filters and 3D coordinates of the object, $\sigma(.)$ represents the squashing function. In fully connected CNN layers, input and output layers are connected with different weights.

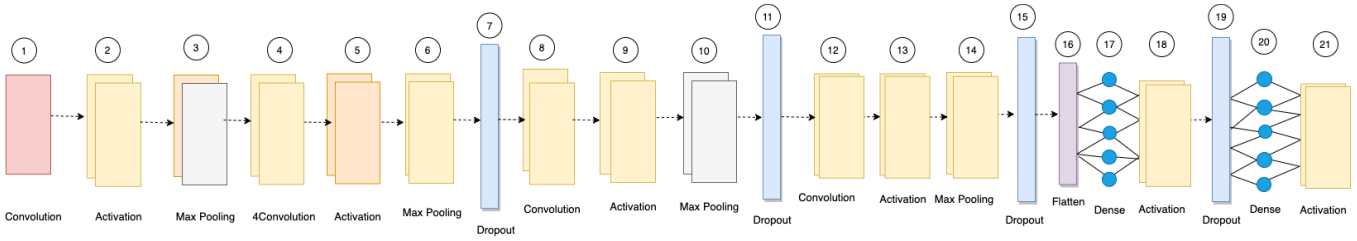

Fig. 3. Block diagram for 1-D CNN.

TABLE IV
1D CNN

Layer	Output Shape
1. Convolution (Input)	2047x32
2. Activation	2047x32
3. Max Pooling	2047x32
4. Convolution	2047x32
5. Activation	2047x32
6. Max Pooling	2047x32
7. Dropout	2047x32
8. Convolution	2047x64
9. Activation	2047x64
10. Max Pooling	2047x64
11. Dropout	2047x64
12. Convolution	2047x128
13. Activation	2047x128
14. Max Pooling	2047x128
15. Dropout	2047x128
16. Flatten	262016
17. Dense	128
18. Activation	128
19. Dropout	128
20. Dense	1
21. Activation	1

C. 1-D CNN

For RF classification, we created a custom 1D CNN model. We present the block diagram for our 1D model in Fig. 3. The model summary is found in Table IV. The one-dimensional CNN model used for classification of the RF samples contained a total of 21 layers. Of the 21 layers, there was an input layer, 19 hidden layers and an output layer. The hidden layers consisted of four convolutional layers, four max-pooling layers, five rectified linear unit (ReLU) activation layers, two dense layers, two dropout layers and one dropout layer. The size of the input RF samples was 2047 dimensions.

1) Forward-Propagation and Back-Propagation in 1-D CNN: CNN derives the accuracy and computational complexity from Forward Propagation and Backward Propagation processes. Forward Propagation feeds the input data to the neural network that moves the input from the input layer throughout the network toward the output layer. The Forward Propagation in 1D CNN can be expressed as [17]:

$$x_k^l = b_k^l + \sum_{i=1}^{N_{l-1}} Conv(w_{ik}^{l-1}, s_i^{l-1}) \tag{5}$$

where b_k^l represents the bias of the kth neuron at layer l, s_i^{l-1} represents the output of the ith neuron at layer $l-1$, w_{ik}^{l-1} represents the kernel from the ith neuron at layer $l-1$ to the kth neuron at layer l, and $Conv(.)$ represents the convolution process with zero padding. Each hidden layer accepts the input data, processes it as per the activation function and passes to the successive layer [17]. With the learning factor ϵ, the weights are updated as

$$w_{ik}^{l-1}(t+1) = w_{ik}^{l-1}(t) - \epsilon \frac{\partial E}{\partial w_{ik}^{l-1}}(t) \tag{6}$$

and the biases are updated as

$$b_k^1(t+1) = b_k^1(t) - \epsilon \frac{\partial E}{\partial b_k^l}(t) \tag{7}$$

where E represents the mean square error (MSE) between target and output vectors for a given input vector for the kth neuron at layer l, and t and $t+1$ represent the current and next time instances.

In summary, first, Forward Propagation happens from the input layer to the output layer to find outputs of each neuron at each layer s_i^l, $\forall i$ and $\forall l$. Second, back propagation computes E at the output layer and back-propagate it to first hidden layer to compute the $\frac{\partial E}{\partial w_{ik}^{l-1}}$ and $\frac{\partial E}{\partial b_k^l}$, $\forall k$ and $\forall l$., Finally, the weight and bias are updated using Eq. (6) and Eq. (7) respectively.

IV. Proposed Ensemble Method

To address many of the limitations with prior UAV detection, we present an ensemble model. Ensemble learning is a machine learning approach that combines multiple classification models in the prediction process using multiple datasets. Ensemble learning serves as a useful approach for the multi-class classification problem as it allows for multiple weak learners to be integrated to provide an overall strong learner. Furthermore, the diversity of the feature classes in our study is aided in the diversity of our selected feature classes. Ensemble learning operates best when the weak learners are sufficiently diverse. Previous research for different context and application has highlighted that ensemble learning performs well with classification problems [18]. In other words, ensemble approach integrates different classifiers into a single classifier to solve classification problems. The integrated model provides a final score that can be selected based upon a variety of methods.

Our experiment is divided into two categories, the first experiment is for a binary classification where we implement the UAV detection for a given image, audio or RF/wireless features. This experiment handles two use-cases for UAV detection: either "a UAV" was detected or "no UAV".

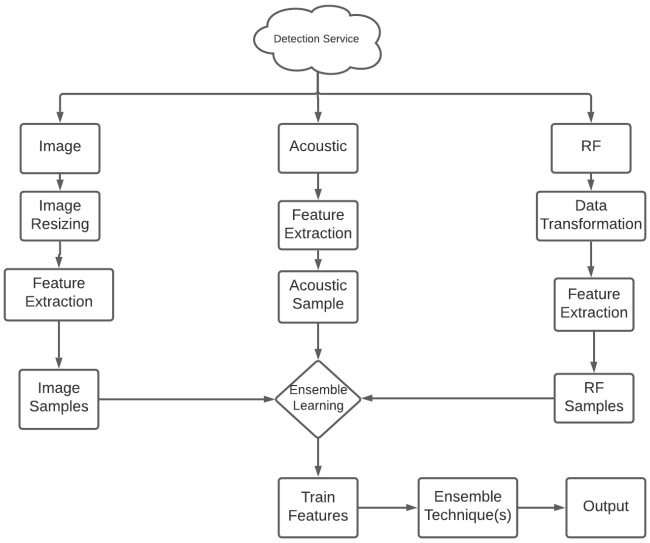

Fig. 4. Flow model for detecting UAVs.

These binary classifiers were used to build out the base classifiers for the ensemble model. Our proposed typical flow model is presented in Fig. 4.

Next, we discuss the criteria for ensemble method. Basically, there are three base ensemble criteria for ensemble method that we leverage: majority vote, average, and weighted average.

1) The first method is the max voting principle. The max voting method operates by combining the predictions from multiple other models. The model is useful as it combines the prediction scores of weak learners to provide an improved prediction. This max vote prediction is based on decision variables for a given class $d_{i,j}$ are represented as:

$$\text{MaxVoting} = \underset{j\in\{1,2\ldots C\}}{\arg\max} \sum_{i=1}^{n} d_{i,j} \tag{8}$$

where n represents the number of classifiers and C represents the number of applicable classes.

2) The next ensemble method is averaging, which is the process that leverages multiple base classifiers and combines them into a single averaged output. This ensemble model assumes equal weights for the individual base classifiers. This model can be representation as:

$$\text{Averaging} = \frac{1}{n}\sum_{i=1}^{n} p_i \tag{9}$$

where n represents the number of classifiers and p_i represents the prediction of the ith classifiers.

3) The last ensemble technique that we will compare is the weighted average. This technique integrates several models based upon their proportional contribution to the new model. This approach is useful as it allows us to apply weight values to each contributing base models when producing the final prediction. The weighted averaging approach uses the performance of the base classifiers as validation. This approach can be represented as:

$$\text{WeightedAverage} = \frac{1}{n}\sum_{i=1}^{n}(p^i w^i) \tag{10}$$

where w_i is the weight and p_i is the prediction of the ith classifier with the $\sum w_i = 1$ for $i = 1$ to n.

Next, we describe the construction of our ensemble model for the UAV detection and identification problem. Each of our data samples were collected with the goal of identifying UAVs operating within a restricted space. In the next section, we discuss data acquisition, preprocessing, and feature engineering for the data source.

V. Dataset and Experimental Setup

To implement the proposed method, we integrated public data sources from [3], [19]. To our knowledge, there were no available datasets which directly addressed the multi-class problem of UAV identification. The image and audio data were obtained from a publicly available dataset [20]. The dataset contained four classes labeled as UAVs, Thunder, Birds, and Planes broken up into two sections.

To acquire the audio and image samples, the authors collected image and audio data by using high resolution cameras and multiple microphones operating within the restricted area [3]. The RF samples were used from a public dataset which was collected over multiple experiments. The RF class samples developed using a public dataset used for detection and identification of UAVs. In the experiment [19], the authors leveraged flight controllers and mobile phones to send RF commands for the UAVs under analysis. To control the UAV devices, the mobile phones required mobile specific applications for each respective UAV.

The recorded RF data was around 10.25s and the RF UAV communication for each flight mode is roughly around 5.25s [19]. Furthermore, as mentioned in [19], the RF UAV database was populated with the required signatures by implementing and organizing dataset samples in a tree manner. For the purpose of this study, we focus on K-Fold classification binary classification of the RF samples. This consists of assessing the UAV detection system based upon the UAV being off and the background activities are recorded and the UAVs being on and the RF activities are recorded.

The data pre-processing is critical steps. To prepare the images for analysis, the images were resized to 241 $\times$ 241. For the audio samples, prior to beginning analysis, time intervals to which the analysis would take place needed to be selected. Then, it is needed to transform signal to a linear scale. The mel-Scale is a metric that serves to represent the acoustic frequency on a linear scale. The mel scale serves as a logarithmic transformation of a signal's frequency. "The Mel scale is defined as the perceptual scale of pitches judged by listeners to be equal distance from each other" [21]. The mel scale can be calculated as [21].

$$\text{melScale} = 2595\log_{10}\left(1+\frac{f}{700}\right) \tag{11}$$

TABLE V
TRAINING/TESTING BREAKDOWN

Class	Training	Testing
Images	448	192
Audio	448	192
RF	448	192

where f represents the Hertz conversion into mels. The mel spectrogram represents the portrayal of the sound within the time and frequency form.

To integrate the RF signals, we followed the Matlab scripts provided for data generation. Once we were able to process the labeled data samples, we created a python script to convert the CSV samples into.npy format for classification. Npy file format was used to transpose the samples and provide a more efficient format for processing. Table V shows the data distribution.

VI. PERFORMANCE EVALUATION METRICS

For evaluating the ensemble methods, true values are compared against the predicted values. We also calculate true positive (TP), false positive (FP), true negative (TN), and false-negative (FN) values. Next we calculate accuracy, precision, recall, and F-1 score.

The False Positive Rate (FPR) is a measure of accuracy for our models. This measure is defined as the probability of falsely rejecting the null hypothesis. As it relates to our study, this measure would serve to allow us to detect external objects which are not malicious UAVs operating within the restricted space. The False Positive Rate (FPR) (aka specificity) is calculated as:

$$FPR = \frac{FP}{N} = \frac{FR}{FP + N} \tag{12}$$

where the FP represents the number of false positives, N represents the number of true negatives and $FP+N$ represents the total number of negatives. This measure evaluates the probability that a false alarm will be raised within our model.

The True Positive Rate (TPR) is an accuracy measure which examines the probability that a true positive test will actually return a positive reading. Within the context of detecting Malicious UAVs, this measure will assist in determining the probability of positively detecting malicious UAVs. The (TPR), sensitivity and recall are calculated as follows:

$$TPR = \frac{TP}{TP + FN} = 1 - FNR \tag{13}$$

where the TP represents the number of true positives and FN represents the number of false negatives.

A False Negative Rate (FNR) is an accuracy measure that seeks to determine when the model wrongly indicates that the model wrongly classified the sample. As it relates to our study, this would represent a sample being falsely classified as either a UAV or no UAV. The FNR Is calculated as:

$$FNR = \frac{FN}{TP + FN} \tag{14}$$

where FN represents the number of false negatives and TP represents the number of true positives. The $TP + FN$ represents the numbers of positives.

Precision seeks to quantify the number of correct positive predictions made by the model. As it relates to our study, this measure will allow us to calculate the ratio of positively identified UAVs divided by the overall correctly identified samples. Precision is calculated as:

$$Precision = \frac{TP}{(TP + FP)} \tag{15}$$

where TP represents the number of true positives and FP represents the number of false positives.

Recall, which is also called the sensitivity measure. This measure represents the ratio of correct positive predictions to the total of overall positives samples. As it relates to our study, this would represent the ratio of positively identified UAVs divided by the overall positive UAV samples,

$$Recall = \frac{TP}{(TP + FN)} \tag{16}$$

where TP represents the number of true positive and FN represents the number of false negatives.

The last metric that we use in our evaluation is the F1-Score which relies on recall and precision to provide the accuracy level. The F1-score operates by computing the accuracy based upon the overall unweighted average. The F1 score is computed as:

$$F1 - score = 2 * \frac{Precision \times Recall}{Precision + Recall}. \tag{17}$$

This metric is evaluated by multiplying the result of the precision measure multiplied by the recall divided by the sum of the precision and recall score. The F1-score is useful because this measure provides a single score that balances the relative concerns of precision and recall. In the next section, we will present the accuracy measures of our CNN and ensemble models.

Next, the false alarm *aka* a type I error that evaluates the impact of failing to reject the null hypothesis. As it relates to our research, this would represent the likelihood that the model detects a UAV in the search area when the UAV is not present. The false-alarm rate is calculated as:

$$FalseAlarm = \frac{FP}{FP + TN} \tag{18}$$

where FP represents the number of false positives and TN represents the true negatives. $FP + TN$ represents the total number of true negatives.

The next type of error that we will evaluate is missed detection. Missed detection is a type II error. A missed detection error is a type two error which is the mistaken acceptance of the null hypothesis. The second kind of error is the mistaken acceptance of the null hypothesis due to a test procedure. A type II error (false-negative or missed detection) occurs if the investigator fails to reject a null hypothesis that is false in the population. This metric represents the probability that the test will miss a true positive.

$$MissedDetection = \frac{FN}{(FN + TP)} \tag{19}$$

TABLE VI
PERFORMANCE COMPARISON OF DIFFERENT MODELS

Model	Accuracy	Precision	Recall	F-Measure
Image Classifier	95.83	94.90	96.88	95.87
Audio Classifier	100	100	100	100
RF Classifier	92.71	91.84	93.75	92.78
Max Voting	96.88	96.88	96.88	96.88
Average	96.88	96.88	96.88	96.88
Weighted Average	97.40	95.05	100.0	97.46

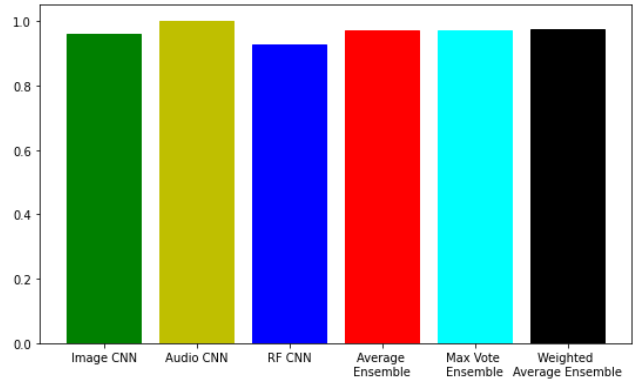

Fig. 5. Accuracy score of different approaches.

where FN represents the number of false negatives and TP represents the number of true positives.

VII. PERFORMANCE EVALUATION, MODEL ERRORS, AND DISCUSSION

A. Performance Evaluation

In this section, we compare the performance of the proposed ensemble methods with the CNN deep learning methods. We evaluated the performance of these models using the performance metrics outlined above. The accuracy scores were validated via a 10-fold validation. These performance measure can be found in Table VI. As it relates to the accuracy score, the Audio classifier had an accuracy score of 100%. All three ensemble models performed better than the baseline image and the RF CNN models. The RF model maintained the lowest accuracy score when compared to the other models.

As previously noted, models accuracy performance measures the ratio of true positives and true negatives to all positive and negative observations. This measure allows us to accurately predict whether the model will correctly predict the UAV/no UAV outcome of the predictions in our dataset. The accuracy measure is useful in providing a high-level evaluation of our models. Still, it cannot correctly highlight any potential errors that could be found in samples that have not been evaluated. The Accuracy score for the Audio classifier was 100%. The accuracy score for the Weighted Average approach was second highest with 97.40% followed by the Average and Max Voting score at 96.88%. The image classifier score was 95.83% and the lowest score of all of the models was the RF accuracy score at 92.71%.

A model's precision score represents its ability to correctly predict the positives out of all the positive predictions it made.

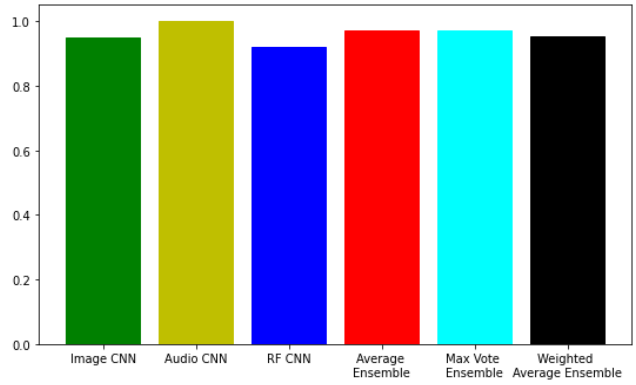

Fig. 6. Precision score of different approaches.

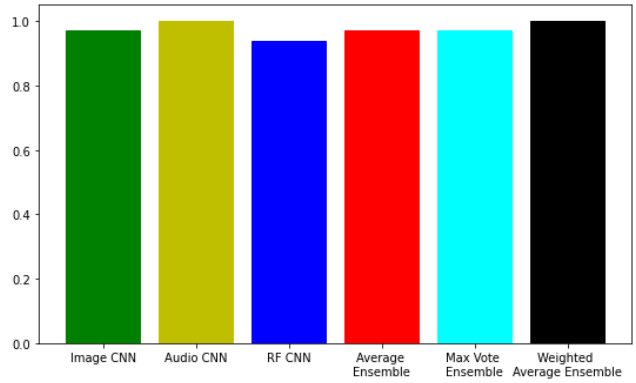

Fig. 7. Recall score of different approaches.

This accuracy measure is most beneficial when examining the effectiveness of a prediction when the classes are imbalanced. This metric is relevant to our model performance as we maintained a traditional 70% for training and 30% for testing. The precision measures allow us to evaluate our models' ability to identify all the UAVs operating within the restricted space. Per our study, the Audio classifier maintained a precision score of 100%. The precision score for the Max Voting and Average ensemble methods was 96.88%. The image had a precision score of 94.90%, and the lowest precision score was the RF classifiers at 91.84%.

The recall score of a learning model denoted the learning models' ability to predict the positives out of actual positives correctly. Unlike the precision measure, this measure evaluates the model's performance based on the accurate predictions related to all of the positive predictions. A higher recall score represents a model's ability to identify positive and negative samples. A high recall score would effectively identify the UAV and non UAV samples as it relates to our application. The audio classifier and the weighted average classifiers had a recall score of 100%. The RF classifier maintained the lowest recall score with 93.75%, while the image classifier, max voting classifier, and average classifiers maintained a score of 96.88%.

The last measure that we present in our paper is the F1 score. As previously noted, this score operates as a function of the precision and recall scores. This metric evaluates the performance of our models while providing equal weights to

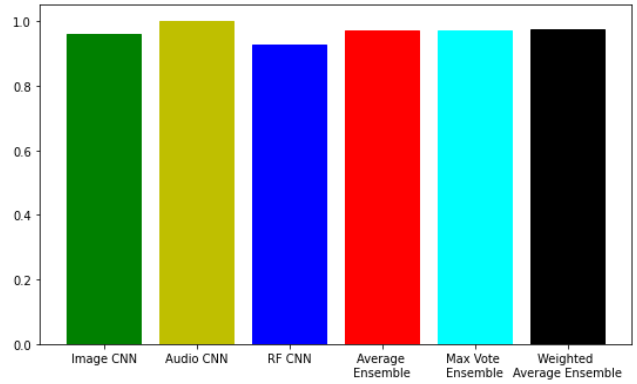

Fig. 8. F Measure score of different approaches.

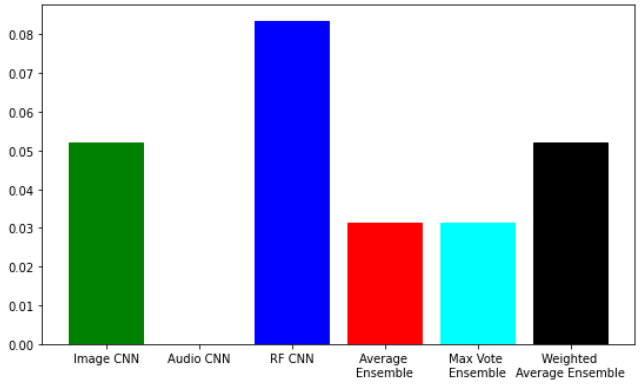

Fig. 9. False alarm scores of different approaches.

TABLE VII
TYPE I AND TYPE II ERROR COMPARISON OF DIFFERENT MODELS

Model	Type 1 Error (False Alarm)	Type II Error (Missed Detection)
Image Classifier	0.052	0.0312
Audio Classifier	0.00	0.00
RF Classifier	0.083	0.063
Max Voting	0.0312	0.0312
Average	0.0312	0.0312
Weighted Average	0.0312	0.00

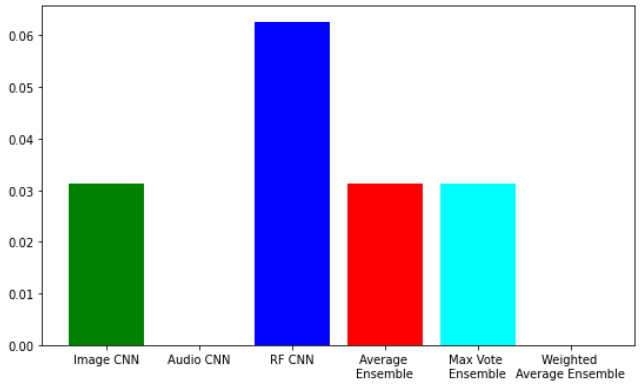

Fig. 10. Miss detection scores of different approaches.

the precision and recall when evaluating the model's performance. This measure differs from the accuracy metric as it isn't measured by the overall number of observations while providing a useful overall score for determining our models' overall output quality. Again, the Audio classifier maintained an F1 score of 100%. The weighted average method score was second-highest, with 97.46%. The max voting and the average classifier's F1 score were 96.88%, while the image classifier had an F1 score of 95.87%. Overall, the lowest F1 score of 92.78% belonged to the RF classifier.

Next, we present the false alarm and missed detection metrics for different models in Table VII.

The audio classifier had a false alarm score of 0%, which performed the best among all classifiers. The max voting and the average ensemble models had false alarm scores of .0312%. The image and weighted max classifiers had false alarm scores of .052, while the RF signal score was the highest at .083%.

The audio classifier and weighted average classifier had a missed detection score of 0%, which performed the best among all classifiers. The image, max voting, and the average classifiers had missed detection scores of .0312%. The RF classifier performed had the highest missed detection score at .052%

Across the performance metrics, our ensemble model performed better than the single modal classifiers. This is significant as it highlights the degree that single modal classifiers can effectively be integrated to improve the detection of drones. The implementation of our three ensemble models also provided insight as to how different strategies can be leveraged to improve detection even with partial knowledge or imprecise data. Collectively, our models showed the effectiveness of using multi-modal ensemble methods despite noted limitations in data. Furthermore, Type I (false alarm) and Type II (miss-detection) measures help provide insight into the impact of leveraging the model in the real world. Across all of our classifiers, RF classifier had the highest scores for Type I and Type II errors. The higher Type I and Type II errors across the single modals would be greater cause of concern but, could effectively be mitigated through leveraging an ensemble approach for multi-modal detection.

The complexity of the proposed ensemble approach is $\mathcal{O}(MNmn)$ where $m \times n$ is the kernel size and $M \times N$ is the input data size.

VIII. CONCLUSION AND FUTURE WORK

This paper presented a framework that consists of two base binary CNN classifiers that are integrated into an ensemble learning approach to improve detection of UAVs based upon multiple features such as audio, video/image and RF signals. This approach was pursued to mitigate many of the environmental limitations with feature-based classification systems. This approach differs from prior research as it integrates multiple feature classes for a single ensemble-based approach using CNN. Despite the relatively high accuracy, the model can be further improved by a range of techniques including integrating meta-learning ensemble methodologies such as stacking or boosting.

Typically, Ensemble methods have proven useful in improving the overall classification accuracy but also have been known to increase the model's computational cost when applied to the real-world application Ensemble learning models increase in complexity relative to the number of models that are combined into the model. Each integrated model increases the cost of complexity based upon the additional parameters that must be evaluated within the larger model. The cost associated with the parameters of multiple integrated classifiers that the detection algorithms must also account for the loss due to every possible miss-classification error to minimize the overall expected risk where ensemble models are notable is that they provide a reasonable trade-off between classification accuracy and computational time.

Our future research will focus on expanding the identification of UAVs by integrating physical properties, depth information, and more diverse data samples using the proposed ensemble model using 2-3/3-D CNN.

REFERENCES

[1] J. Gambrell. *Ship Tied to Israeli Billionaire Attacked off Oman, 2 Killed*. Accessed: Mar. 10, 2021. [Online]. Available: https://apnews.com/article/ship-israeli-billionaire-oman-c43203ff0262ba2c4bfafb31cc42fa50

[2] Z. Wu, K. Suresh, P. Narayanan, H. Xu, H. Kwon, and Z. Wang, "Delving into robust object detection from unmanned aerial vehicles: A deep nuisance disentanglement approach," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1201–1210.

[3] S. Jamil *et al.*, "Malicious UAV detection using integrated audio and visual features for public safety applications," *Sensors*, vol. 20, no. 14, p. 3923, Jul. 2020.

[4] G. Ding, Q. Wu, L. Zhang, Y. Lin, T. A. Tsiftsis, and Y.-D. Yao, "An amateur drone surveillance system based on the cognitive Internet of Things," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 29–35, Jan. 2018.

[5] R. Opromolla, G. Fasano, and D. Accardo, "A vision-based approach to UAV detection and tracking in cooperative applications," *Sensors*, vol. 18, no. 10, p. 3391, Oct. 2018.

[6] U. Seidaliyeva, D. Akhmetov, L. Ilipbayeva, and E. T. Matson, "Real-time and accurate drone detection in a video with a static background," *Sensors*, vol. 20, no. 14, p. 3856, Jul. 2020.

[7] B. Taha and A. Shoufan, "Machine learning–based drone detection and classification: State-of-the-art in research," *IEEE Access*, vol. 7, pp. 138669–138682, 2019.

[8] M. Nijim and N. Mantrawadi, "Drone classification and identification system by phenome analysis using data mining techniques," in *Proc. IEEE Symp. Technol. Homeland Secur. (HST)*, May 2016, pp. 1–5.

[9] J. Busset *et al.*, "Detection and tracking of drones using advanced acoustic cameras," in *Unmanned/Unattended Sensors Sensor Networks, Advanced Free-Space Optical Communication Techniques Applications*, vol. 9647. Bellingham, WA, USA: SPIE, 2015, Art. no. 96470F.

[10] S. Al-Emadi, A. Al-Ali, A. Mohammad, and A. Al-Ali, "Audio based drone detection and identification using deep learning," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 459–464.

[11] Y. Seo, B. Jang, and S. Im, "Drone detection using convolutional neural networks with acoustic STFT features," in *Proc. 15th IEEE Int. Conf. Adv. Video Signal Surveill. (AVSS)*, Nov. 2018, pp. 1–6.

[12] C. Xu, B. Chen, Y. Liu, F. He, and H. Song, "RF fingerprint measurement for detecting multiple amateur drones based on STFT and feature reduction," in *Proc. Integr. Commun. Navigat. Surveill. Conf. (ICNS)*, Sep. 2020, p. 4G1-1.

[13] S. Al-Emadi and F. Al-Senaid, "Drone detection approach based on radio-frequency using convolutional neural network," in *Proc. IEEE Int. Conf. Informat., IoT, Enabling Technol. (ICIoT)*, Feb. 2020, pp. 29–34.

[14] M. Ezuma, F. Erden, C. K. Anjinappa, O. Ozdemir, and I. Guvenc, "Detection and classification of UAVs using RF fingerprints in the presence of Wi-Fi and Bluetooth interference," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 60–76, 2020.

[15] Y.-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 111–118.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, May 2017, vol. 60, no. 6, pp. 84–90.

[17] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: A survey," *Mech. Syst. Signal Process.*, vol. 151, Apr. 2021, Art. no. 107398.

[18] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 161–168.

[19] M. S. Allahham, M. F. Al-Sa'd, A. Al-Ali, A. Mohamed, T. Khattab, and A. Erbad, "Dronerf dataset: A dataset of drones for RF-based detection, classification and identification," *Data Brief*, vol. 26, Oct. 2019, Art. no. 104313.

[20] S. JamilKaggle. (2020). *Malicious UAVs Detection Dataset, Kaggle Malicious UAVs Dataset*. [Online]. Available: https://www.kaggle.com/sonain/malicious-uavsdetection

[21] D. O'Shaughnessy, *Speech Communication: Human and Machine* (Series in Electrical Engineering). Reading, MA, USA: Addison-Wesley, 1987.

James McCoy received the M.S. degree in computer science from the University of the District of Columbia, DC, USA, in 2018. He is currently pursuing the Ph.D. degree with Howard University, under the supervision of Dr. D. B. Rawat, where his research focus is on the intersection of machine learning, cyber security, and UAVs.

Atul Rawal received the first Ph.D. degree in nanoengineering from North Carolina A&T State University, Greensboro, USA, in 2020. He is currently pursuing the second Ph.D. degree in electrical engineering with a specialization in artificial intelligence/machine learning at Howard University, under the supervision of Dr. D. B. Rawat.

Danda B. Rawat (Senior Member, IEEE) is currently the Associate Dean of Research and Graduate Studies, a Full Professor of electrical engineering and computer science (EECS), the Director of the Data Science and Cybersecurity Center, and the Director of the DoD Center of Excellence in Artificial Intelligence and Machine Learning (CoE-AIML), Howard University. He is a fellow of the Institution of Engineering and Technology (IET). He was a recipient of the NSF CAREER Award in 2016, the Department of Homeland Security (DHS) Scientific Leadership Award in 2017, and the U.S. Air Force Research Laboratory (AFRL) Summer Faculty Visiting Fellowship in 2017, among others. He is also an ACM Distinguished Speaker (2021–2023).

Brian M. Sadler (Life Fellow, IEEE) received the B.S. and M.S. degrees in electrical engineering from the University of Maryland, College Park, and the Ph.D. degree in electrical engineering from the University of Virginia, Charlottesville. He is currently the U.S. Army Senior Scientist for Intelligent Systems and a Fellow of the Army Research Laboratory (ARL), Adelphi, MD, USA. His research interests include multi-agent intelligent systems, signal processing, and information science. He has been a Distinguished Lecturer of the IEEE Communications Society and the IEEE Signal Processing Society.