2-8-2022

Jamming Detection and Classification in OFDM-based UAVs via Feature- and Spectrogram-tailored Machine Learning

Y. Li
Purdue University Northwest

J. Pawlak
Purdue University Northwest

J. Price
Purdue University Northwest

K. Al Shamaileh
Purdue University Northwest

Q. Niyaz
Purdue University Northwest

See next page for additional authors

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 19, 2022, accepted February 5, 2022, date of publication February 8, 2022, date of current version February 16, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3150020

Jamming Detection and Classification in OFDM-Based UAVs via Feature- and Spectrogram-Tailored Machine Learning

YUCHEN LI[1], JERED PAWLAK[2], JOSHUA PRICE[1], KHAIR AL SHAMAILEH[1], (Member, IEEE), QUAMAR NIYAZ[1], SIDIKE PAHEDING[3], (Member, IEEE), AND VIJAY DEVABHAKTUNI[4], (Senior Member, IEEE)

[1]Electrical and Computer Engineering Department, Purdue University Northwest, Hammond, IN 46323, USA
[2]Panduit Corporation, Lockport, IL 60441, USA
[3]Department of Applied Computing, Michigan Technological University, Houghton, MI 49931, USA
[4]Electrical and Computer Engineering Department, The University of Maine, Orono, ME 04469, USA

Corresponding author: Khair Al Shamaileh (kalshama@pnw.edu)

This work was supported by the National Science Foundation, Secure and Trustworthy Cyberspace Program, under Award 2006662.

ABSTRACT In this paper, a machine learning (ML) approach is proposed to detect and classify jamming attacks against orthogonal frequency division multiplexing (OFDM) receivers with applications to unmanned aerial vehicles (UAVs). Using software-defined radio (SDR), four types of jamming attacks; namely, barrage, protocol-aware, single-tone, and successive-pulse are launched and investigated. Each type is qualitatively evaluated considering jamming range, launch complexity, and attack severity. Then, a systematic testing procedure is established by placing an SDR in the vicinity of a UAV (i.e., drone) to extract radiometric features before and after a jamming attack is launched. Numeric features that include signal-to-noise ratio (SNR), energy threshold, and key OFDM parameters are used to develop a *feature-based* classification model via conventional ML algorithms. Furthermore, spectrogram images collected following the same testing procedure are exploited to build a *spectrogram-based* classification model via state-of-the-art deep learning algorithms (i.e., convolutional neural networks). The performance of both types of algorithms is analyzed quantitatively with metrics including detection and false alarm rates. Results show that the spectrogram-based model classifies jamming with an accuracy of 99.79% and a false-alarm of 0.03%, in comparison to 92.20% and 1.35%, respectively, with the feature-based counterpart.

INDEX TERMS Cybersecurity, convolutional neural networks (CNNs), deep learning, jamming, machine learning (ML), orthogonal frequency division multiplexing (OFDM), software-defined radio (SDR), spectrogram, unmanned aerial vehicles (UAVs).

I. INTRODUCTION

Recently, unmanned aerial vehicles (UAVs) have been widely adopted in various civil, military, and scientific applications such as climate monitoring, disaster management, merchandise delivery, search and rescue operations, space exploration, and wildlife tracking [1]–[4]. According to [5], the UAV market will be witnessing a growth from USD 27.4B in 2021 to USD 58.4B by 2026. This projected growth is mainly attributed to the increasing demand for automation and the rapid advances in enabling technologies.

Several efforts have been dedicated to promote the control and navigation of UAVs [6]–[10]. However, few have addressed the associated cybersecurity challenges despite their potential in compromising UAVs performance, which in some cases, may result in catastrophic consequences [11]. For example, it was shown that an attacker can engineer a drone to sniff wireless signals from other nearby drones, disconnect them from their legitimate networks, and form an army of zombie drones [12]. Another example is the GPS jamming attack that plummeted 46 drones during a Hong Kong show and caused a damage of at least USD 127,500 [13]. Hence, further research on the cybersecurity of UAVs that addresses the detection and mitigation of their associated cyberattacks is of a grave significance. Here, jamming detection is of a

The associate editor coordinating the review of this manuscript and approving it for publication was Moussa Ayyash.

particular interest and is tackled with two approaches that enable attack detection and classification. Jamming mitigation, on the other hand, is outside the scope of this effort. Nonetheless, several methods were reported in literature, where the use of artificial intelligence (i.e., enforced learning) and path planning were proposed [14]–[18].

II. RELATED WORK

Cyberattacks on UAVs include data interception, data manipulation, and denial-of-service (i.e., jamming). Data interception/manipulation attacks are often mitigated with broadcast authentication [19]–[23] and secure location verification [24], [25]. The former applies cryptographic and non-cryptographic schemes; whereas the latter verifies the locations of UAVs with distance bounding, group verification, Kalman filtering, multilateration, and traffic modeling. Although these methods have shown promise in improving UAVs security, the added hardware and/or software to the existing protocols as well as time-stamping adjustments were major constraints that setback their ready acceptance in foreseeable future. Also, these methods are inefficient for detecting jamming, where the UAV-controller communication is interrupted with interference to impose security threats and cease information exchange [26]–[28]. With the readily available software-defined radio (SDR), attackers can easily launch this interference to disturb a UAV trajectory, potentially leading to collisions. Hence, developing affordable jamming detection techniques that also comply with the existing standards are of utmost importance. These techniques must facilitate high detection rate and low false-alarm rate. Furthermore, they should enable jamming classification to allow for selecting the optimum countermeasure routine that ensures operational security through informed decisions.

In our previous work, the impacts of four jamming types on UAV security were analyzed qualitatively (i.e., range, complexity, severity) and quantitatively with conventional machine learning (ML) algorithms [29]. These algorithms were exploited for jamming detection/classification based on extracted signal features. In this work, deep learning models, i.e., four configurations of convolutional neural networks (CNNs), are adopted based on spectrogram images. The spectrogram-based approach improved the classification accuracy from 92.2% (i.e., feature-based approach) to 99.79% and reduced the false-alarm rate from 1.35% to 0.03% as will be presented in greater detail in Sections IV and V. Finally, this work contributes an additional dataset (i.e., spectrogram images) for training and testing ML classifiers. This dataset and the spectrogram-based approach proposed herein, were not provided nor explored in [29]. Also, this work differs from other existing techniques in the following aspects:

1) In contrast to imposing modifications to the existing protocols [19]–[25], [30], readily available radiometric features and spectrogram images are used to develop ML models for detecting and classifying jamming.
2) In comparison to the simulation-based attack scenarios [31]–[39], this work utilizes SDR for launching jamming attacks that facilitate detection and classification with realistic environments and training datasets.
3) Here, jamming detection/classification via deep learning models is introduced. These models are trained and tested with spectrograms that characterize the jamming spectrum. This approach outperforms its feature-based counterpart in classification accuracy.
4) The datasets that are collected and used to develop the feature- and spectrogram-based classification models (i.e., features, images) are made publicly available.

It is worth mentioning that ML was proposed for satellite communications, vehicle Ad Hoc networks (VANETs), 5G networks, Internet of Things (IoT), and UAVs with applications including jamming detection [40]–[44], object detection, trajectory optimization, swarm communication, situational awareness, and malicious attack mitigation [45]–[47].

The remaining of this is paper is summarized as follows: Section III describes the jamming types entailed in this work, the experimental setup, and attack scenarios. Section IV presents the feature-based conventional ML models for detecting/classifying jamming. Section V elaborates on the spectrogram-based deep learning models via CNNs. Finally, conclusions and future work are given in Section VI.

III. JAMMING ATTACKS AND EXPERIMENTAL SETUP

The attack scenario and experimental setup for four jamming types are presented herein. Holy Stone HS720E is used for testing. This drone has a communication range and transmission power of 1000 meters and 16 dBm, respectively. It also uses IEEE 802.11 orthogonal frequency division multiplexing (OFDM) at 2.4 GHz [48]. B210 SDR from National Instruments and GNURadio are exploited to launch attacks within 40 MHz bandwidth to accommodate all subcarriers.

A. TYPES OF JAMMING ATTACKS

1) *Barrage:* In this type, noise from normal distribution is launched at the communication band to increase interference level at the receiver (i.e., UAV). Therefore, barrage is often used when the transmission frequency is unknown to the jammer. Barrage jamming is simple to launch; however, its efficiency reduces as the transmission bandwidth increases.

2) *Single-Tone:* Here, a high-power interference is launched to interfere with the center frequency that the target uses for data exchange. This interference signal is generally denoted as $J(t) = A_j cos(2\pi f_0 t + \theta_j)$, where A_j is the jamming amplitude, f_0 is the center frequency, and θ_j is a phase shift.

3) *Successive-Pulse:* In this type, pulse-sequence is launched to interfere with the target's operation band, and is given as:

$$J(t) = A_j \sum_{n=1}^{N_j} \delta(t - nT) \tag{1}$$

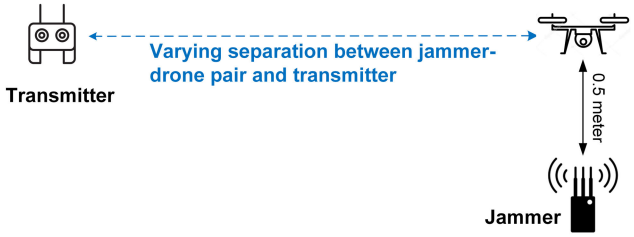


FIGURE 1. Experimental setup to obtain effective jamming range.

TABLE 1. Measured Range of a Successful Jamming Attempt.

Type	Barrage	Single-tone	Success.-pulse	P-aware
Range (m)	80	145	350	155

TABLE 2. Qualitative Analysis for the four Jamming Types.

		Complexity			
		1	2	3	4
Severity	1			Success.-pulse	
	2				P-aware
	3		Single-tone		
	4	Barrage			

where N_j is the jamming tones. The period T is set such that 312.5 KHz frequency spacing is realized between generated pulses (i.e., subcarrier spacing in IEEE 802.11 OFDM).

4) *Protocol-Aware:* This type transmits low interference via shot-noise pulses to corrupt the ongoing transmissions while minimizing detection probability. In other words, the jammer simulates the transmitter of the targeted protocol without affecting other standards occupying the same bandwidth [49].

B. EXPERIMENTAL SETUP

Two experimental environments are established to evaluate the qualitative and quantitative impacts of the jamming types. The qualitative evaluation analyzes severity, launch complexity, and effective jamming range. The quantitative evaluation entails radiometric extractions (i.e., signal features, spectrogram images) through data collection under different jamming scenarios. Data is used for training and validating ML algorithms for jamming detection and classification.

1) *Qualitative Evaluation:* The separation between the jammer (i.e., B210 SDR) and drone is fixed to 0.5 meter. To measure the effective jamming range, the separation between the jammer-drone pair and the transmitter is increased gradually for each jamming type in an unobstructed outdoor setup, as shown in Figure 1. Here, effective jamming is defined as a complete loss of signal and is reported in Table 1 for each type. Results indicate that barrage has the most jamming range among all types due to spreading interference over all OFDM subcarriers in comparison to interfering with the center (or selected) frequencies as in single-tone and successive-pulse jamming or transmitting shot-noise as in protocol-aware jamming. Table 2 depicts the qualitative findings for launch complexity and severity in a scale of 1 to 4, where 4 is the highest score. Barrage has the least launch complexity as it does not require extensive knowledge about the communication bandwidth. Nonetheless, it has the highest severity. Single-tone jamming is relatively simple to launch. Nevertheless, this type is inefficient in scenarios where multiple frequencies or subcarriers are used. Successive-pulse jamming with $N_j = 64$ pulses has a moderate launch complexity as interference pulses need careful positioning with respect to the center and subcarrier frequencies. The output power, P_j, of the jammer is distributed on pulses in a way that the interference pulse power is P_j/N_j. Therefore, it has

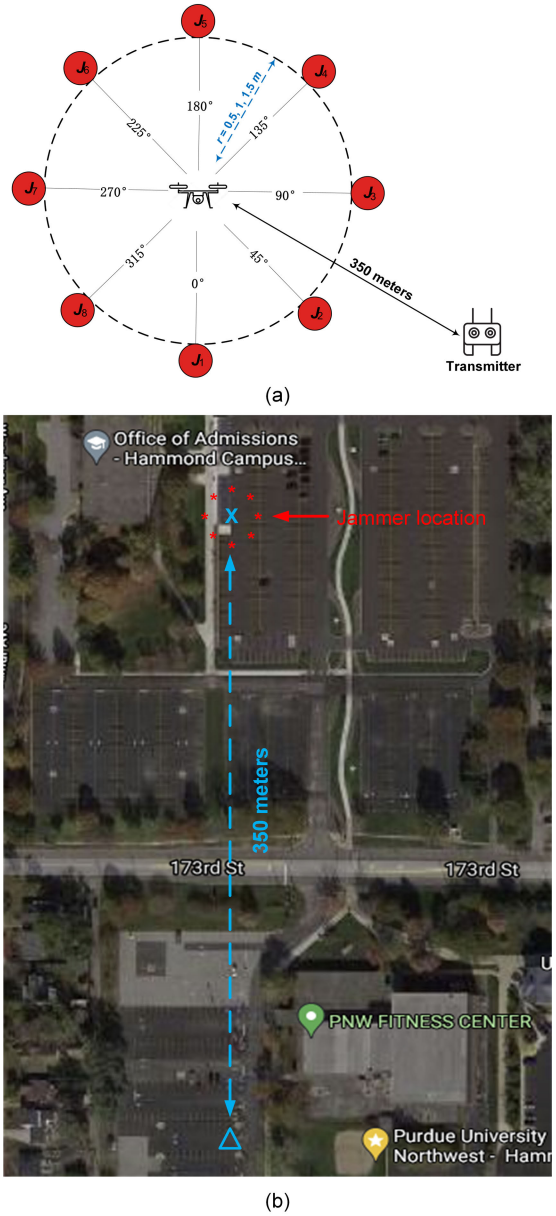


FIGURE 2. Extraction of signal features and spectrogram images under no-jamming/jamming scenarios at different jammer locations: (a) testing setup and (b) testing location from Google maps. The △, ∗, and × represent the trnasmitter, jammer, and drone, respectively.

TABLE 3. Distribution of Samples in the Training and Testing Datasets.

	Training set distribution (70%)		Testing set distribution (30%)	
	No. of records	Avg. ± Dev.	No. of records	Avg. ± Dev.
Clean	7026	16.54±0.43	3045	16.53±0.44
Barrage	2374	3.31±1.95	1018	3.28±1.98
Single-tone	2348	4.55±3.07	1030	4.70±3.10
Success.-pulse	2368	1.36±0.71	999	1.35±0.78
P-aware	2379	2.17±1.57	978	2.29±1.48

TABLE 4. List of Features Used in each Case.

	Features		
Case	OFDM Estimator	Energy Detector	SNR Probe
1	Subcarrier Spacing Symbol Time Subcarrier Length CP Length	Avg Received Power Threshold	Avg Signal Power Avg Noise Power SNR
2	Subcarrier Spacing Subcarrier Length CP Length	Avg Received Power Threshold	Avg Signal Power Avg Noise Power SNR
3	Subcarrier Spacing Subcarrier Length CP Length	Avg Received Power Threshold	Avg Signal Power SNR

the least severity. Protocol-aware jamming has the highest launch complexity as it requires a thorough knowledge of the communication protocol. It also has a moderate severity since limited-power interference is launched at the transmission bandwidth to maintain low detection probability.

2) *Quantitative Evaluation:* Radiometric data (i.e., signal features, spectrogram images) are collected for ML training/classification. The goal here is to develop models that not only detect jamming, but also identify its type. To collect such data, the transmitter-drone separation is set to 350 meters, which is the minimum separation where all jamming types are effective. Then, without jamming presence, features and images are obtained at the drone with B210 SDR and GNU-Radio modules. The same procedure is repeated in the presence of each of the jamming types, where a second SDR is utilized as jammer at eight locations J_i, $i = 1, 2, \ldots 8$, around the drone, one at a time. This procedure is performed for radii $r = 0.5$, 1, and 1.5 meters as shown in Figure 2.

IV. FEATURE-BASED CLASSIFICATION

As discussed in section III, B210 SDRs and GNURadio are used to launch different jamming attacks and extract radiometric data. Figures 3(a) and 3(b) show simplified GNU-Radio flow graphs for launching the attacks and extracting features, respectively. Nine features are extracted to train ML algorithms for detecting and classifying jamming attacks. Of these features, four are specific to OFDM (i.e., *subcarrier length*, *cyclic prefix* (CP) *length*, *subcarrier spacing*, and *symbol time*). The subcarrier length represents the number of subcarriers being used. The CP length is utilized to control symbol overlapping, and the subcarrier spacing is the frequency separation between subcarriers, which is the reciprocal of symbol time [50]. The ① *OFDM Estimator* block shown in Figure 3(b) is used to extract these features [51]. The ② *Energy Detector* block is used to extract the *average received power* and *threshold* [51]. The threshold is a binary indicator that returns 1 once the *average received power* exceeds a certain level and returns 0 otherwise. Finally, three more features; namely, *signal-to-noise ratio* (SNR), *average signal power*, and *average noise power* are extracted from the ③ *SNR Estimator Probe* block. It is paramount to point out that the *average received power* in ② conveys noise energy; whereas the *average signal power* in ③ presents the estimated signal power excluding noise power. At the end of the experiment featured in Figure 2, a total of 23,565 signal samples are collected. Of these samples, 10,071 are obtained under no jamming; whereas 3,392, 3,367, 3,378, and 3,357 are obtained in the presence of barrage, single-tone, successive-pulse, and protocol-aware jamming, respectively. The complete dataset with all the 23,565 samples is provided in [52]. To develop the ML classifiers, this dataset is divided into training and testing sets as detailed in Table 3, which suggest a balanced distribution among the jamming types, leading to high detection and classification accuracy. During the processing of features, it is found that the (*symbol time, subcarrier length*) and (*threshold, average noise power*) pairs are highly correlated. Thus, different ML models are explored by reducing the dimension of the features dataset. In Case 2, *symbol time* is eliminated; whereas *symbol time* and *average noise power* are eliminated in Case 3. The list of features in each case is given in Table 4. The models are built with six conventional algorithms: Decision Tree (DT), K-Nearest Neighbors (KNN), Logistic Regression (LR), Multi-layer Perceptron (MLP), Naive Bayes (NB), and Random Forest (RF). The metrics for model evaluation are demonstrated in (2) and include the detection rate (DR), precision, recall, F-score (FS), and false-alarm rate (FAR).

$$DR = \frac{Correctly\ Predicted\ Samples}{Samples\ in\ the\ Dataset} \tag{2.a}$$

$$Precision = \frac{True\ Positive\ Samples}{True\ Positive + False\ Positive\ Samples} \tag{2.b}$$

$$Recall = \frac{True\ Positive\ Samples}{True\ Positive + False\ Negative\ Samples} \tag{2.c}$$

$$F - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.d}$$

$$FAR = \frac{False\ Positive\ Samples}{False\ Positive + True\ Negative\ Samples} \tag{2.e}$$

DR denotes the percent of correctly detected samples over total dataset samples. Precision is defined as the number of positive samples predicted as positive (i.e. true positive) divided by the sum of true positive and negative samples predicted as positive (i.e. false positive). Recall is the number of true positive samples divided by the sum of true positive and positive samples predicted as negative (i.e. false negative). F-score is computed from precision and recall to represent their harmonic mean. Lastly, FAR is the number of

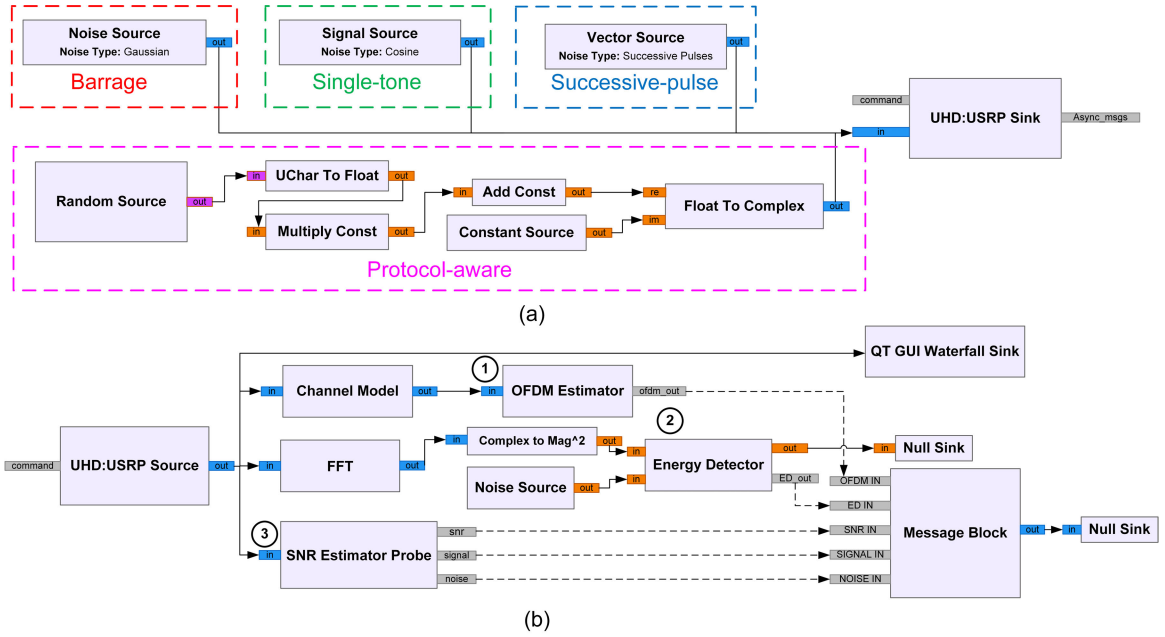


FIGURE 3. Simplified GNURadio flow graph for (a) launching the jamming attacks and (b) extracting the radiometric features.

false positive samples divided by the sum of false positive and true negative samples predicted by the model. Two- and five-class ML models are created for each of the cases summarized in Table 5. The two-class models predict whether a jamming attack is launched or not; whereas the five-class models detect the jamming attack and identify its type (i.e., barrage, single-tone, successive-pulse, and P-aware). During model development, 10-fold cross-validation is used in the training/validation stages. Once a model is trained, evaluation is performed on the test set; and the DR, F-score, and FAR are computed. Grid search is used to find the optimal hyper-parameters for each algorithm. The performance of the developed classifiers for the two- and five-class models are given in Table 5. All classifiers are executed on a 64-bit Windows 8 machine with Intel®Core™i7-6900K CPU @ 3.20 GHz processor and 128 GB memory. The two-class model classifiers achieved almost 100% DR and validation accuracy (VA) in classifying records into "no-jamming" or "presence of jamming". Moreover, seven features (i.e., Case 3) are found sufficient for developing an efficient and trustworthy two-class model. On the other hand, the RF model has the highest VA of 91.80%, 92.20%, and 86.23% for Cases 1, 2, and 3, respectively, among the five-class models. Also, RF achieved the highest DR and F-score in almost all cases with a DR of 92.11%, 92.20%, and 85.95% as well as an F-score of 0.92, 0.92, and 0.86 for Cases 1, 2, and 3, respectively. Finally, RF results in the highest training and testing times of 5.4s and 0.410s, respectively, in comparison to the other algorithms for its associated large number of decision trees. Eliminating the symbol time from the dataset (i.e., Case 2) has a marginal effect in improving classification. However, eliminating both symbol time and average noise power (i.e., Case 3) degrades the performance significantly. Figures 4(a)-(c) show the confusion matrices of the five-class RF model for each case. None of the clean records are misclassified as jamming records. Rather, misclassification occurs only among the jamming types; particularly, barrage and protocol-aware, which is attributed to the similarity in their spectral properties (i.e., interference in these types targets the entire transmission bandwidth, but at different intensity levels). The weighed FAR values are obtained from Figure 4 to be 1.35% for Case 1, 1.33% for Case 2, and 2.38% for Case 3. Finally, there is no false-alarm in the two-class models regardless of the number of features used in training/validation.

The validity of the feature-based models for detecting and classifying jamming attacks is further analyzed considering samples with different SNR levels. To this end, the extracted SNRs for all scenarios are plotted in Figure 5. Five sub-datasets, summarized in Table 6, are created to represent all jamming types. Sub-datasets 1, 2, 3, 4, and 5 have samples with SNR intervals of {0-1}, {1-2}, {2-3}, {3-4}, and {4-5} dB, respectively, and are established from the testing set. The samples with SNR values when there is no jamming are excluded to emphasize classification accuracy only among the four jamming types. The six classifiers developed earlier are tested with these five sub-datasets, and testing entailed the three cases with nine, eight, and seven features. The resulting DRs are illustrated in Figure 6 with the following observations in mind: 1) The overall accuracy dropped due to removing the clean samples from the sub-datasets, i.e., "Clean" samples are not within any of the SNR intervals. 2) No misclassification as "no jamming" occurred in almost all classifiers. 3) The least accuracy is obtained with

TABLE 5. Metrics for the Two- and Five-class Jamming Detection Models (VA: Validation Accuracy, DR: Detection Rate, FS: F-score, CTR (in seconds): CPU Training Time, CTE (in seconds): CPU Testing Time.

Performance metrics for five-class models											
	Case 1: Nine Features			Case 2: Eight Features			Case 3: Seven Features			Time (Case 2)	
ML Classifier	VA (%)	DR (%)	FS	VA (%)	DR (%)	FS	VA (%)	DR (%)	FS	CTR(sec)	CTE(sec)
LR	82.45 (± 0.65)	82.90	0.82	82.75 (± 0.67)	82.73	0.82	79.42 (± 0.76)	78.95	0.79	0.860	0.002
KNN	84.47 (± 0.74)	84.23	0.84	84.87 (± 0.74)	83.50	0.84	83.70 (± 0.72)	83.40	0.83	0.131	0.130
NB	79.30 (± 0.80)	78.74	0.79	79.40 (± 0.80)	78.33	0.78	77.50 (± 0.79)	77.80	0.77	0.002	3.550
DT	91.60 (± 0.70)	92.52	0.93	91.90 (± 0.64)	91.75	0.92	84.96 (± 0.75)	84.75	0.85	0.058	≈ 0
RF	**91.80 (± 0.06)**	**92.11**	**0.92**	**92.20 (± 0.60)**	**92.20**	**0.92**	**86.23 (± 0.79)**	**85.95**	**0.86**	**5.404**	**0.411**
MLP	78.02 (± 1.70)	79.60	0.79	77.50 (± 2.13)	76.25	0.75	77.46 (± 1.80)	75.60	0.72	1.807	0.005
Performance metrics for two-class models											
LR	**100.00 (± 0.00)**	**100.00**	**1.00**	**100.00 (± 0.00)**	**100.00**	**1.00**	**100.00 (± 0.00)**	**100.00**	**1.00**	**0.022**	**0.003**
KNN	99.92 (± 0.07)	99.89	1.00	99.93 (± 0.06)	99.94	1.00	99.93 (± 0.06)	99.96	1.00	0.135	0.135
NB	99.80 (± 0.09)	99.79	1.00	99.77 (± 0.12)	99.85	1.00	99.77 (± 0.11)	99.86	1.00	0.006	≈ 0
DT	100.00 (± 0.02)	99.98	1.00	100.00 (± 0.02)	99.98	1.00	99.98 (± 0.03)	100.00	1.00	0.009	≈ 0
RF	**100.00 (± 0.00)**	**100.00**	**1.00**	**100.00 (± 0.00)**	**100.00**	**1.00**	**100.00 (± 0.00)**	**100.00**	**1.00**	**2.344**	**0.203**
MLP	99.72 (± 0.60)	99.98	1.00	99.23 (± 2.50)	99.98	1.00	99.70 (± 0.50)	99.89	1.00	1.112	0.001

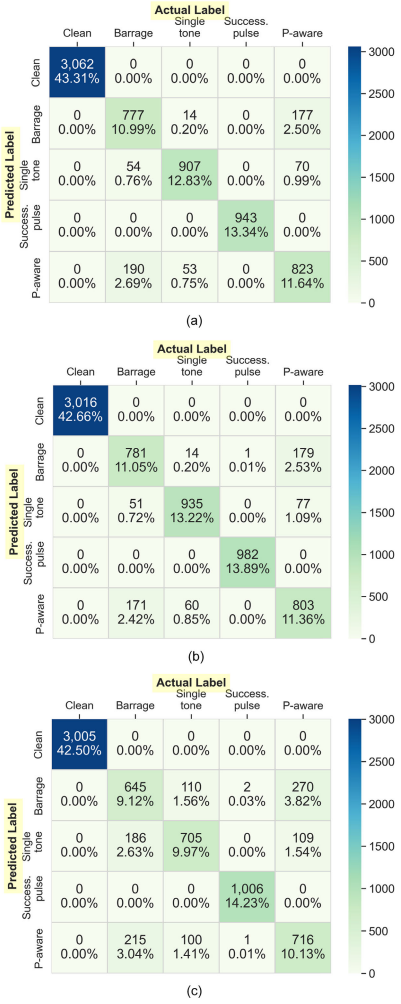


FIGURE 4. Confusion matrices of the five-class RF model for (a) nine features, (b) eight features, and (c) seven features.

sub-datasets 3 (i.e., SNRs ∈ {2-3} dB) and 4 (i.e., SNRs ∈ {3-4} dB), which is attributed to the high number of protocol-aware samples in comparison to the samples from the other jamming types. It is noteworthy to point out here that

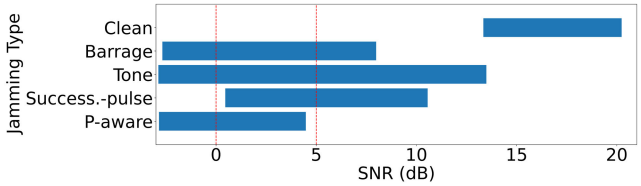


FIGURE 5. The measured SNRs for the clean (i.e., no jamming) and jamming scenarios for the four jamming types.

TABLE 6. Distribution of Samples in each of the Five Sub-datasets. All Sub-datasets are Obtained from the Original Testing Set.

Sub-dataset	Barrage	Single-tone	Success.-pulse	P-aware	Total Samples
1: SNRs ∈ {0-1}	62	45	299	111	517
2: SNRs ∈ {1-2}	78	53	608	129	868
3: SNRs ∈ {2-3}	143	69	119	265	596
4: SNRs ∈ {3-4}	172	100	18	386	676
5: SNRs ∈ {4-5}	336	153	7	24	520

protocol-aware jamming has the highest misclassification as depicted in the confusion matrices presented in Figure 4. As a result, this SNR-based investigation shows that imbalances in the dataset (e.g., imbalance in the number of samples for each jamming type) significantly affect the classification quality and accuracy. Therefore, the datasets utilized for training and testing the feature-based ML classifiers in this work (i.e., Table 3) are balanced and have adequate number of jamming and clean samples to facilitate high detection and classification accuracy.

V. SPECTROGRAM-BASED CLASSIFICATION

To improve the five-class classification accuracy, deep learning models trained with spectrogram images obtained from no-jamming/jamming scenarios are developed. These models have multiple processing layers that use backpropagation to model the parameters of complex datasets (e.g., image, speech), thereby facilitating precise classification [53]. Here, CNNs are used for their leading advantage in processing images by not only efficiently extracting image properties (e.g., size, color, pattern), but also pooling a large number

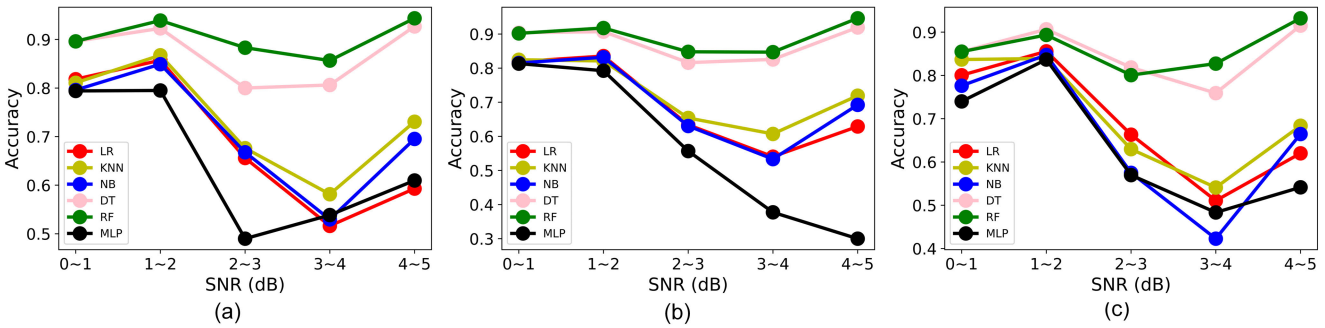


FIGURE 6. The resulting accuracy of the feature-based classifiers as a function of the five SNR intervals. (a) Case 1 with nine features, (b) Case 2 with eight features, and (c) Case 3 with seven features.

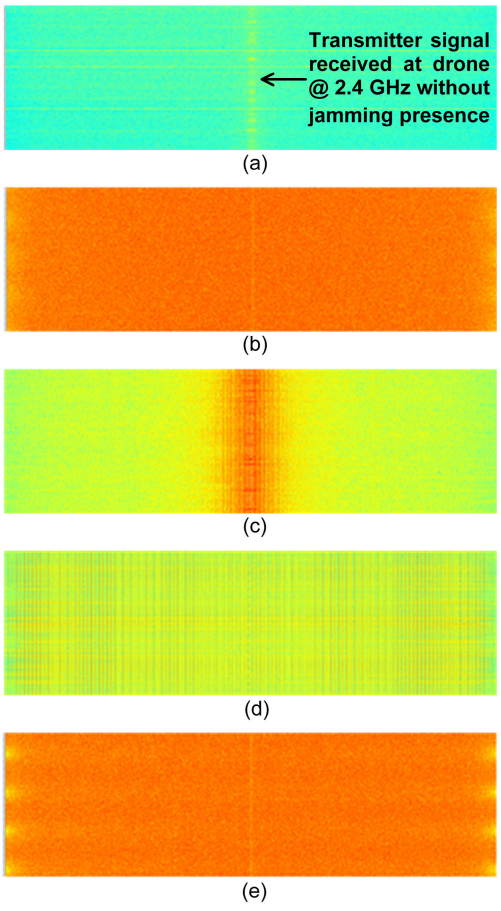


FIGURE 7. Spectrograms under (a) no jamming, (b) barrage, (c) single-tone, (d) successive-pulse, and (e) P-aware jamming.

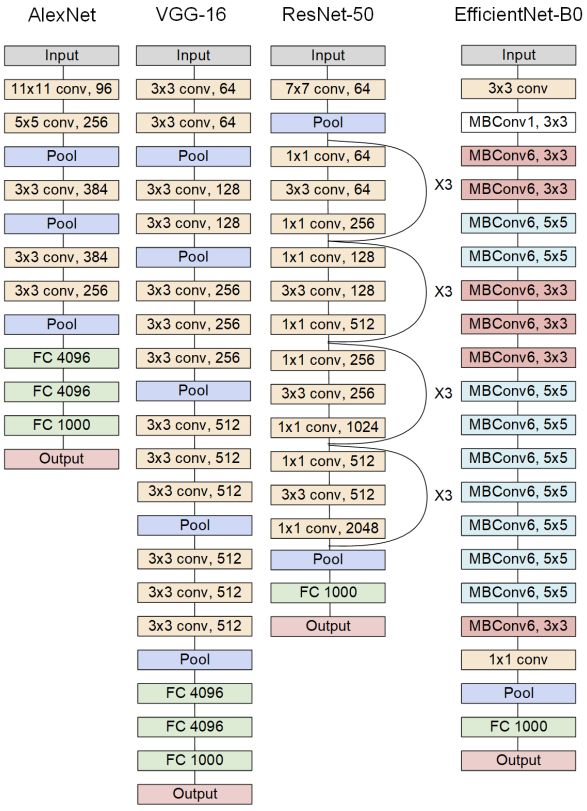


FIGURE 8. The configurations of the four CNN-based classifiers.

of pixels to reduce calculations. The configuration of CNNs consists of input layer, convolution layer, pooling layer, fully-connected layer, and output layer. The input layer feeds images to the hidden layers. The convolution layer contains convolution kernels for extracting features, and their size gradually decreases, or remains constant, as more convolution layers are added. The pooling layer retains the highest-scoring features and discards others with low scores. It also reduces model parameters; thus, reduces computations at later layers. The fully-connected layer is similar to a regular neural network (i.e.,neurons in one layer are connected to those in the next layer). The output layer returns the probability of each class. Weights are adjusted in the network via backpropagation.

Spectrogram images are collected with SDR and QT GUI Waterfall Sink block. Python scripts are developed to capture screenshots during testing. Here, 762 images are collected under no jamming and 204 images are collected for each of the jamming types. The standard image size is scaled down

TABLE 8. Performance Metrics of the CNN Models (VA: Validation Accuracy, DR: Detection Rate, FS: F-score, GTR: GPU Training Time, GTE: GPU Testing Time, CTR: CPU Training Time, CTE: CPU Testing Time).

Performance metrics for five-class models							
ML Classifier	**VA (%)**	**DR (%)**	**FS**	**GTR (sec)**	**GTE (sec)**	**CTR (sec)**	**CTE (sec)**
AlexNet	100.00	99.36	0.99	174	0.82	6765	4.90
VGG-16	94.03	94.50	0.94	1479	5.81	70932	63.30
ResNet-50	99.82	98.10	0.98	1118	2.72	58359	31.84
EfficientNet-B0	**98.55**	**99.79**	**1.00**	**1530**	**2.53**	**39476**	**31.22**
Performance metrics for two-class models							
ML Classifier	**VA (%)**	**DR (%)**	**FS**	**GTR (sec)**	**GTE (sec)**	**CTR (sec)**	**CTE (sec)**
AlexNet	100.00	99.15	0.99	171	0.76	6048	4.86
VGG-16	99.91	99.36	0.99	1478	5.77	52837	63.43
ResNet-50	100.00	99.36	0.99	1114	2.47	52334	32.00
EfficientNet-B0	**99.91**	**100.00**	**1.00**	**1489**	**2.28**	**39351**	**31.55**

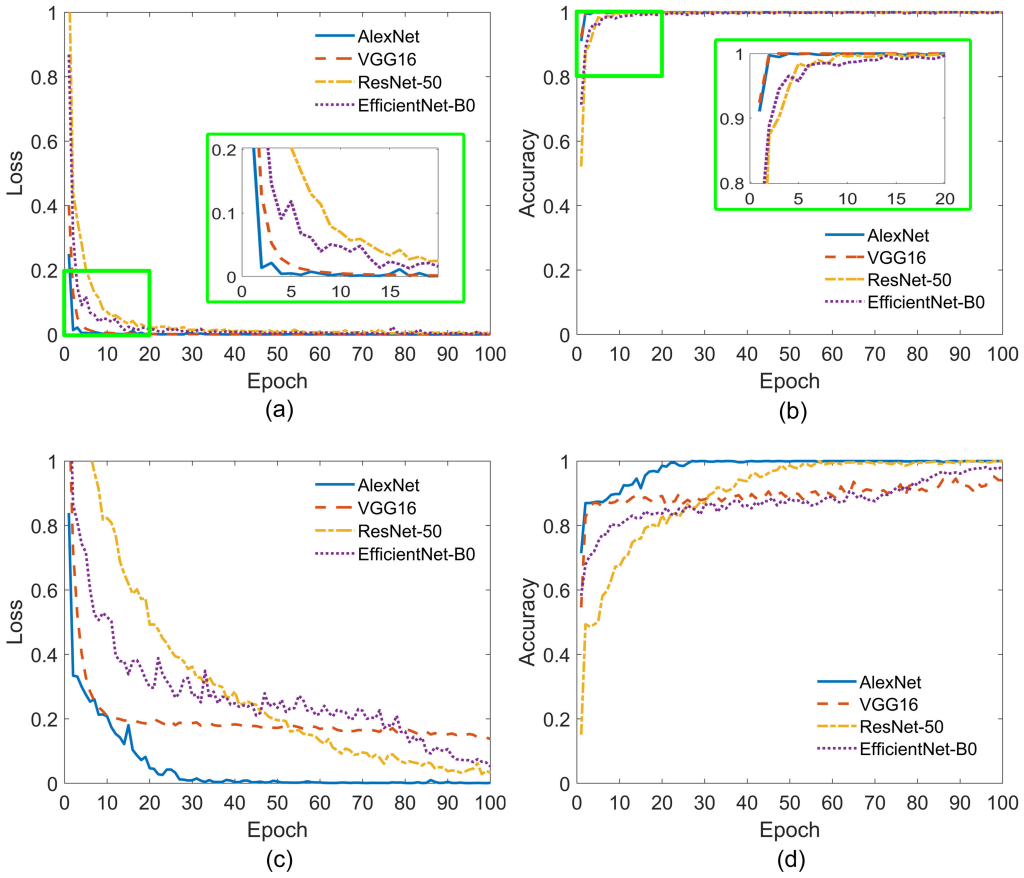


FIGURE 9. Two-class models (a) loss and (b) accuracy. Five-class models (c) loss and (d) accuracy.

from 1688 × 990 × 3 to 422 × 248 × 3 to reduce training time. These images are separated into 70% training and 30% testing. Figure 7 shows sample images in different scenarios. The complete image dataset is made available on [52].

Spectrogram-based classification is realized with four CNN configurations: AlexNet, VGG-16, ResNet-50, and EfficientNet-B0. Figure 8 shows their structures and Table 7 details their parameters. AlexNet uses ReLu activation function and dropout method [54]. ReLu increases training speed and the dropout is added in the first two fully-connected layers to minimize overfitting. It starts with a convolution layer of 11 × 11 kernel size and 96 filters, which reduces to 5 × 5 and 256 filters. It also consists of three convolution layers with 3 × 3 kernel size and three pooling layers. These layers are followed by three fully-connected layers and an output layer. The VGG configuration adds more convolution layers to facilitate accuracy via deep neural networks [55]. However, an excessive addition of such layers potentially

TABLE 7. Parameters of the Images and Deep Learning Algorithms. Stochastic Gradient Descent Solver with 100 Epochs is Considered.

Case	Parameter	Value
Raw image	image size	$1688 \times 990 \times 3$
	image type	.jpg
Pre-processing	image size	$422 \times 248 \times 3$
	image type	.jpg
AlexNet	Learning rate	0.001
	Kernel size	$11 \times 11, 5 \times 5, 3 \times 3$
	Kernel stride	4, 2, 1
	Batch size	64
VGG-16	Learning rate	0.0001
	Kernel size	3×3
	Kernel stride	2, 1
	Batch size	32
ResNet-50	Learning rate	0.0001
	Kernel size	$7 \times 7, 3 \times 3, 1 \times 1$
	Kernel stride	2, 1
	Batch size	32
EfficientNet-B0	Learning rate	0.001
	Kernel size	$5 \times 5, 3 \times 3, 1 \times 1$
	Kernel stride	2, 1
	Batch size	32

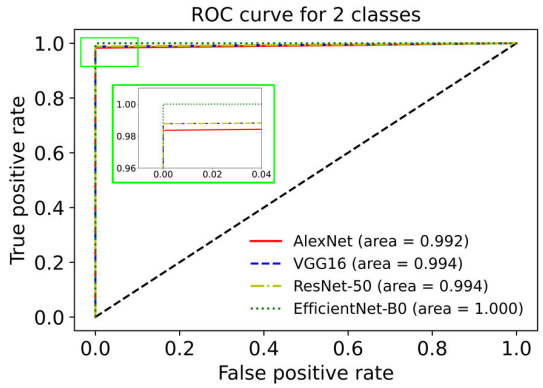


FIGURE 10. ROC curve of the two-class CNN models.

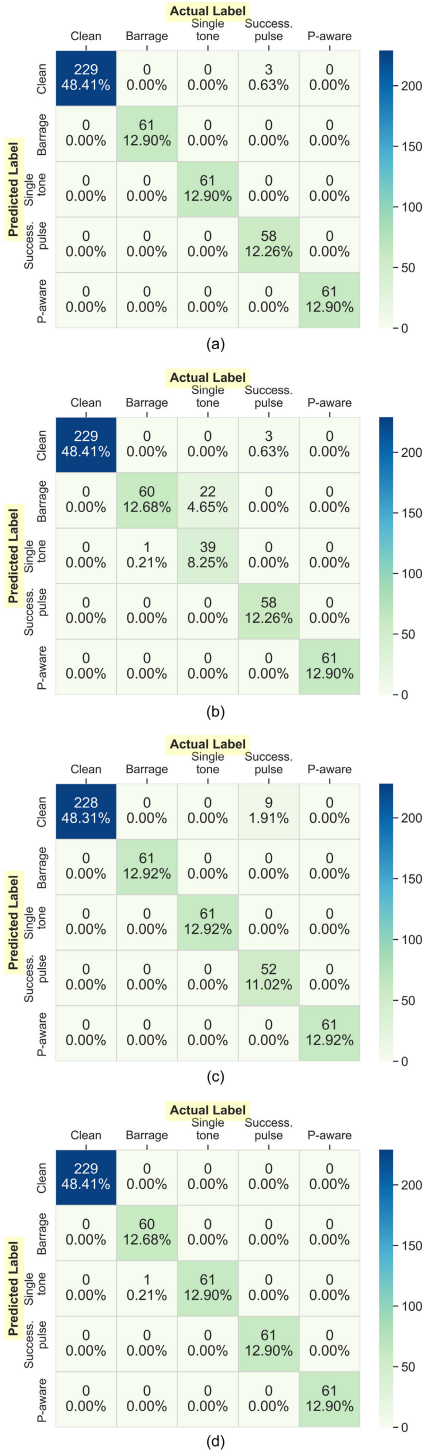


FIGURE 11. Confusion matrices of the five-class CNN models: (a) AlexNet, (b) VGG-16, (c) ResNet-50, and (d) EfficientNet-B0.

leads to gradient dispersion that results in training divergence. Here, VGG-16 is used for image training with five groups of two or three convolution layers of 3×3 kernel size together with five pooling layers, three fully-connected layers, and an output layer. The ResNet configuration addresses the vanishing gradient problem by exploiting batch normalization and by skipping connections among convolution layers [56]. It also comes in different structures including ResNet-18/34/50/101/152. Here, ResNet-50 is adopted, which consists of a 7×7 convolution layer and groups of 1×1, 3×3, and 1×1 convolution layers. It also has two pooling, one fully-connected, and output layers. Lastly, EfficientNet improves accuracy through model scaling and branches into B0–7 [57]. In this work, EfficientNet-B0 is used for its compact architecture, which is characterized by a 3×3 convolution layer followed by moving reverse bottleneck convolution (MBConv) layers with either 3×3 or 5×5 kernels. It also conveys 1×1 convolution, pooling, fully-connected, and output layers.

The training/testing of the four CNN models is performed in two systems. The first uses a 64-bit Windows 8, Intel® Corei7-6900K CPU @ 3.20 GHz processor and 128 GB RAM. The second uses Google Colab with 16 GB RAM and Tesla P100 GPU. All Python codes use Tensorflow with Keras interface. Table 8 shows the DR, VA, F-score, and the training/testing times for the CNN classifiers. EfficientNet-B0 has the highest DR of 100% and 99.79% for the two- and five-class models, respectively;

TABLE 9. Comparison between the Proposed Approach and other State-of-the-art Approaches.

Ref.	Dataset Type	Dataset Source	ML Type	DR (%)	Application	Jamming Type
[31]	Features	Simulations	K-means	-	VANET	Constant, Smart (Detection)
[32]	Features	Simulations	RF, SVM, MLP	97.50	5G Networks	Barrage (Detection)
[40]	Spectrograms	Simulations	CNN & SVM	93.10	Satellites	Barrage, Pilot-tone, Intermittent (Detection)
[41]	Spectrograms	Measurements	CNN & RNN	86.10	OFDM	Barrage, Reference Signal (Detection and Classification)
[42]	Features	Measurements	DT, AdaBoost, SVM	97.00	OFDM	Constant, Reactive (Detection)
[43]	Features	Simulations	DT, RF, SVM	99.06	IoT Networks	Intermittent (Detection)
		Measurements	DT, RF, SVM, KNN	89.70		
[44]	Features	Borrowed	MLP, MLP&SVM	94.51	5G Networks	Constant, Random, Deceptive, Reactive (Detection and classification)
This Work	Features	Measurements	LR, KNN, NB, DT, RF, MLP	92.20	UAVs	Barrage, Single-tone, Success.-pulse, P-aware (Detection and classification)
	Spectrograms	Measurements	CNN	99.79		

whereas, AlexNet results in the lowest training/testing times, highest VA, and fastest convergence rate as shown in Figure 9(a)-(d). It is also found that the training and testing times for the CNN models are significantly higher than those obtained by the conventional ML algorithms, which is attributed to the CNNs deep and complex architectures. However, since detection times (i.e., GTE, CTE) result from classifying 472 images, the average processing times of the five-class EfficientNet-B0 model to classify an image are 0.005s with GPU and 0.066s with CPU, enabling real-time jamming detection and classification. Figure 10 shows the receiver operating characteristic (ROC) of the two-class models and indicates that EfficientNet-B0 outperforms other classifiers in jamming detection. Lastly, the weighted FARs are computed from the confusion matrices, shown in Figure 11, to be 0.6% for AlexNet, 1.55% for VGG-16, 1.86% for ResNet-50, and 0.03% for EfficientNet-B0. It is noteworthy to mention that complexity and severity of a given jamming type have no contribution to its classification accuracy. For example, barrage jamming is the simplest to launch, whereas protocol-aware has the most launch complexity. Yet, their feature- and spectrogram-based misclassifications are nearly 2.5% and 0%, respectively. Similarly, barrage has the highest severity among the four jamming types, whereas successive-pulse has the lowest severity. Nonetheless, their feature- and spectrogram-based misclassifications are $< 1\%$ and 0%, respectively, as demonstrated in the confusion matrices in Figures 4 and 11. Table 9 shows a comparison between the proposed method and those reported in literature in detecting and/or classifying jamming attacks with applications to satellite communications, OFDM, VANETs, and 5G/IoT networks. This work entailed four jamming attacks with the highest detection/classification accuracy. Moreover, six conventional and four deep learning models are trained and tested with realistic datasets of extracted signal features and images obtained after rigorous measurement routines.

VI. CONCLUSION

An ML method is proposed to detect/classify four types of jamming attacks on OFDM receivers with application to UAVs. Each attack is built with B210 SDR and launched against a drone to qualitatively analyze its impacts considering severity, complexity, and jamming range. Then, an SDR is used in proximity to the drone to record key OFDM parameters, threshold, signal power, noise power, and SNR for the feature-based approach as well as spectrogram images for the spectrogram-based approach. The former approach is explored with six algorithms and the latter is realized with four CNN algorithms to achieve higher jamming detection/classification accuracy. All models are validated with metrics including detection and false alarm rates, and showed that jamming is detected with 92.2% and 99.79% confidence following the feature- and spectrogram-based classifiers, respectively. This method requires the integration of a data extraction module with the UAV receiver to obtain real-time signal features and/or images to facilitate the detection and classification routines. This integration potentially imposes the need for interface circuits adjoined with a careful analysis of power aspects and hardware imperfection. Future work will entail exploring more jamming types (e.g., deceptive, reactive), incorporating maximum-likelihood classification with advanced SNR probing, and investigating UAV-specific anti-jamming solutions (e.g., trajectory optimization).

REFERENCES

[1] M. Messinger and M. Silman, "Unmanned aerial vehicles for the assessment and monitoring of environmental contamination: An example from coal ash spills," *Environ. Pollut.*, vol. 218, pp. 889–894, Nov. 2016.
[2] A. Bhardwaj, L. Sam, Akanksha, F. J. Martín-Torres, and R. Kumar, "UAVs as remote sensing platform in glaciology: Present applications and future prospects," *Remote Sens. Environ.*, vol. 175, pp. 196–204, Mar. 2016.
[3] R. Allison, J. Johnston, G. Craig, and S. Jennings, "Airborne optical and thermal remote sensing for wildfire detection and monitoring," *Sensors*, vol. 16, no. 8, p. 1310, Aug. 2016.
[4] J. Qi, D. Song, H. Shang, N. Wang, C. Hua, C. Wu, X. Qi, and J. Han, "Search and rescue rotary-wing UAV and its application to the Lushan ms 7.0 earthquake," *J. Field Robot.*, vol. 33, no. 3, pp. 290–321, May 2016.
[5] *Unmanned aerial vehicles UAV market*. Accessed: Aug. 10, 2021. [Online]. Available: https://www.marketsandmarkets.com/Market-Reports/unmanned-aerial-vehicles-uav-market-662.html
[6] J. Paredes, C. Jacinto, R. Ramírez, I. Vargas, and L. Trujillano, "Simplified fuzzy-PD controller for behavior mixing and improved performance in quadcopter attitude control systems," in *Proc. IEEE ANDESCON*, Oct. 2016, pp. 1–4.

[7] J. Braga, H. Velho, G. Conte, P. Doherty, and E. Shiguemori, "An image matching system for autonomous UAV navigation based on neural network," in *Proc. 14th Int. Conf. Control, Automat., Robot. Vis. (ICARCV)*, Nov. 2016, pp. 1–6.

[8] M. Mullins, K. Foerster, N. Kaabouch, and W. Semke, "A multiple objective and behavior solution for unmanned airborne sense-and-avoid systems," in *Proc. AUVSI Unmanned Syst. North Amer. Conf.*, Las Vegas, NV, USA, 2012, pp. 1254–1266.

[9] M. Mullins, K. Foester, and N. Kaabouch, "Traffic alerting system for manned-unmanned aircraft airspace conflicts," in *Proc. ND EPSCoR/IDeA State Conf.*, 2017.

[10] H. Reyes, N. Gellerman, and N. Kaabouch, "A cognitive radio system for improving the reliability and security of UAS/UAV networks," in *Proc. IEEE Aerosp. Conf.*, Mar. 2015, pp. 1–9.

[11] *Drones are Quickly Becoming a Cybersecurity Nightmare*. Accessed: Aug. 10, 2021. [Online]. Available: https://threatpost.com/drones-breach-cyberdefenses/143075

[12] C. Albanesius. *SkyJack Software Finds and Hijacks Drones*. Accessed: Sep. 1, 2021. [Online]. Available: https://U.K..pcmag.com/security-devices-2/8285/skyjack-software-finds-and-hijacks-drones

[13] S. McCarthy. *HK$1 Million in Damage Caused by GPS Jamming That Caused 46 Drones to Plummet During Hong Kong Show*. Accessed: Sep. 1, 2021. [Online]. Available: https://sg.news.yahoo.com/hk-1-million-damage-caused-080848555.html

[14] K. Ibrahim, S. X. Ng, I. M. Qureshi, A. N. Malik, and S. Muhaidat, "Anti-jamming game to combat intelligent jamming for cognitive radio networks," *IEEE Access*, vol. 9, pp. 137941–137956, 2021.

[15] G. Han, L. Xiao, and H. V. Poor, "Two-dimensional anti-jamming communication based on deep reinforcement learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 2087–2091.

[16] B. Duan, D. Yin, Y. Cong, H. Zhou, X. Xiang, and L. Shen, "Anti-jamming path planning for unmanned aerial vehicles with imperfect jammer information," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2018, pp. 729–735.

[17] H. Wang, J. Chen, G. Ding, and J. Sun, "Trajectory planning in UAV communication with jamming," in *Proc. 10th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2018, pp. 1–6.

[18] L. Xiao, X. Lu, D. Xu, Y. Tang, L. Wang, and W. Zhuang, "UAV relay in VANETs against smart jamming with reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4087–4097, May 2018.

[19] M. Strohmeier, V. Lenders, and I. Martinovic, "On the security of the automatic dependent surveillance-broadcast protocol," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 1066–1087, 2nd Quart., 2015.

[20] M. R. Manesh and N. Kaabouch, "Analysis of vulnerabilities, attacks, countermeasures and overall risk of the automatic dependent surveillance-broadcast (ADS-B) system," *Int. J. Crit. Infrastruct. Protection*, vol. 19, pp. 16–31, Dec. 2017.

[21] K. D. Wesson, T. Humphreys, and B. Evans. (2021). *Can Cryptography Secure Next Generation Air Traffic Surveillance?*. Accessed: Aug. 21, 2021. [Online]. Available: http://users.ece.utexas.edu/~bevans/papers/2015/nextgen/

[22] C. Giannatto, Jr., "Challenges of implementing automatic dependent surveillance broadcast in the nextgen air traffic management system," Ph.D. dissertation, Univ. Maine, Orono, ME, USA, 2015.

[23] B. Danev, H. Luecken, S. Capkun, and K. El Defrawy, "Attacks on physical-layer identification," in *Proc. 3rd ACM Conf. Wireless Netw. Secur. (WiSec)*, 2010, pp. 89–98.

[24] S. Brands and D. Chaum, "Distance-bounding protocols," in *Proc. Workshop Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 1993, pp. 344–359.

[25] B. Xiao, B. Yu, and C. Gao, "Detection and localization of sybil nodes in VANETs," in *Proc. Workshop Dependability Issues Wireless Ad Hoc Netw. Sensor Netw. (DIWANS)*, 2006, pp. 1–8.

[26] *Security Analysis of Unmanned Aircraft Systems*. Accessed: Aug. 21, 2021. [Online]. Available: http://dl.comp.nus.edu.sg/handle/1900.100/6167

[27] D. He, S. Chan, and M. Guizani, "Communication security of unmanned aerial vehicles," *IEEE Wireless Commun.*, vol. 24, no. 4, pp. 134–139, Aug. 2016.

[28] J. Coffed. (2014). *The Threat of GPS Jamming: The Risk to an Information Utility*. Accessed: Aug. 21, 2021. [Online]. Available: https://rntfnd.org/wp-content/uploads/Exelis-GPS-Vulnerability-Assessment-February2014.pdf

[29] J. Pawlak, Y. Li, J. Price, M. Wright, K. Al Shamaileh, Q. Niyaz, and V. Devabhaktuni, "A machine learning approach for detecting and classifying jamming attacks against ofdm-based uavs," in *Proc. 3rd ACM Workshop Wireless Secur. Mach. Learn.*, 2021, pp. 1–6.

[30] M. Sliti, W. Abdallah, and N. Boudriga, "Jamming attack detection in optical UAV networks," in *Proc. 20th Int. Conf. Transparent Opt. Netw. (ICTON)*, Jul. 2018, pp. 1–5.

[31] D. Karagiannis and A. Argyriou, "Jamming attack detection in a pair of RF communicating vehicles using unsupervised machine learning," *Veh. Commun.*, vol. 13, pp. 56–63, Jul. 2018.

[32] Y. Arjoune, F. Salahdine, M. S. Islam, E. Ghribi, and N. Kaabouch, "A novel jamming attacks detection approach based on machine learning for wireless communication," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2020, pp. 459–464.

[33] L. Mokdad, J. Ben-Othman, and A. T. Nguyen, "DJAVAN: Detecting jamming attacks in vehicle ad hoc networks," *Perform. Eval.*, vol. 87, pp. 47–59, May 2015.

[34] A. Nguyen, L. Mokdad, and J. Ben Othman, "Solution of detecting jamming attacks in vehicle ad hoc networks," in *Proc. 16th ACM Int. Conf. Model., Anal. Simul. Wireless Mobile Syst.*, 2013, pp. 405–410.

[35] J. Grover, N. K. Prajapati, V. Laxmi, and M. Gaur, "Machine learning approach for multiple misbehavior detection in vanet," in *Proc. 1st Int. Conf. Adv. Comput. Commun. (ACC)*, Kochi, India, Jul. 2011, pp. 644–653.

[36] H. Liu, B. Lang, M. Liu, and H. Yan, "CNN and RNN based payload classification methods for attack detection," *Knowl.-Based Syst.*, vol. 163, pp. 332–341, Jan. 2019.

[37] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[38] X. Wang, X. Wang, and S. Mao, "RF sensing in the Internet of Things: A general deep learning framework," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 62–67, Sep. 2018.

[39] C. Liu, J. Wang, X. Liu, and Y.-C. Liang, "Deep CM-CNN for spectrum sensing in cognitive radio," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2306–2321, Oct. 2019.

[40] S. Gecgel and G. K. Kurt, "Intermittent jamming against telemetry and telecommand of satellite systems and a learning-driven detection strategy," in *Proc. 3rd ACM Workshop Wireless Secur. Mach. Learn.*, Jun. 2021, pp. 43–48.

[41] S. Gecgel, C. Goztepe, and G. K. Kurt, "Jammer detection based on artificial neural networks: A measurement study," in *Proc. ACM Workshop Wireless Secur. Mach. Learn. (WiseML)*, 2019, pp. 43–48.

[42] O. Puñal, I. Aktaş, C.J. Schnelke, G. Abidin, K. Wehrle, and J. Gross, "Machine learning-based jamming detection for IEEE 802.11: Design and experimental evaluation," in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw.*, Jun. 2014, pp. 1–10.

[43] B. Upadhyaya, S. Sun, and B. Sikdar, "Machine learning-based jamming detection in wireless IoT networks," in *Proc. IEEE VTS Asia Pacific Wireless Commun. Symp. (APWCS)*, Aug. 2019, pp. 1–5.

[44] M. Hachimi, G. Kaddoum, G. Gagnon, and P. Illy, "Multi-stage jamming attacks detection using deep learning combined with kernelized support vector machine in 5G cloud radio access networks," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Oct. 2020, pp. 1–5.

[45] P. S. Bithas, E. T. Michailidis, N. Nomikos, D. Vouyioukas, and A. G. Kanatas, "A survey on machine-learning techniques for UAV-based communications," *Sensors*, vol. 19, no. 23, p. 5170, 2019.

[46] Q. Wu, H. Wang, X. Li, B. Zhang, and J. Peng, "Reinforcement learning-based anti-jamming in networked UAV radar systems," *Appl. Sci.*, vol. 9, no. 23, p. 5173, Nov. 2019.

[47] X. Lu, L. Xiao, C. Dai, and H. Dai, "UAV-aided cellular communications with deep reinforcement learning against jamming," *IEEE Wireless Commun.*, vol. 27, no. 4, pp. 48–53, Aug. 2020.

[48] *Manual & Drivers*. Accessed: Aug. 21, 2021. [Online]. Available: http://holystone.com/en/supports/Drivers.html

[49] A. Hussain, N. A. Saqib, U. Qamar, M. Zia, and H. Mahmood, "Protocol-aware radio frequency jamming in Wi-Fi and commercial wireless networks," *J. Commun. Netw.*, vol. 16, no. 4, pp. 397–406, Aug. 2014.

[50] Y. Cho, J. Kim, W. Yang, and C. Kang," *Introduction OFDM*. Wiley, pp. 111–151, 2010. [Online]. Available: https://ieeexplore.ieee.org/book/5675894

[51] S. Müller and C. Richardson. *GitHub—Gnuradio/Gr-Inspector: Signal Analysis Toolbox for GNU Radio*. Accessed: Aug. 21, 2021. [Online]. Available: https://github.com/gnuradio/gr-inspector

[52] *GitHub: UAVs Jamming Detection and Classification*. Accessed: Nov. 10, 2021. [Online]. Available: https://github.com/michaelevol/uavs_jamming_detection

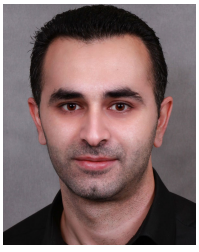
[53] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[54] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1097–1105.
[55] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[57] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

YUCHEN LI received the B.Sc. degree in communications engineering from the Tianjin University of Technology, Tianjin, in 2019, and the M.Sc. degree in electrical and computer engineering from Purdue University Northwest, in 2021. His research interests include machine learning, deep learning, cybersecurity, and wireless communications.

JERED PAWLAK received the B.Sc. degree in electrical engineering and the B.Sc. degree in computer engineering from Purdue University Northwest, in 2021. He is currently working as a Firmware Engineer at Panduit. His research interests include machine learning and software development.

JOSHUA PRICE received the B.Sc. degree in electrical engineering and the B.Sc. degree in computer engineering from Purdue University Northwest, in 2021. He is currently pursuing the M.Sc. degree in electrical and computer engineering. He has returned to the Purdue campus to continue his higher education. His research and learning interests include cybersecurity, software development, wireless security, and circuit design.



KHAIR AL SHAMAILEH (Member, IEEE) received the B.Sc. degree in communications and electronics engineering and the M.Sc. degree in wireless communications engineering from the Jordan University of Science and Technology, in 2009 and 2011, respectively, and the Ph.D. degree in engineering from The University of Toledo, USA, in 2015. He joined the ECE Department, Purdue University Northwest, as an Assistant Professor, in 2016. His research interests include physical layer security, microwave modeling, RF circuit design, sensor networks, localization algorithms, and applied optimization to engineering problems. His research has been sponsored by the National Science Foundation.

QUAMAR NIYAZ received the B.Sc. and M.Sc. degrees in computer science and engineering from Aligarh Muslim University, in 2009 and 2013, respectively, and the Ph.D. degree from The University of Toledo, in 2017. He has been an Assistant Professor in computer engineering with the ECE Department, Purdue University Northwest, since 2017. He has published papers in the areas of computer and networks security, applied machine learning, and cybersecurity education. His research has been sponsored by the National Science Foundation.

SIDIKE PAHEDING (Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Dayton. He is currently an Assistant Professor with the Department of Applied Computing, Michigan Technological University. Prior to joining Michigan Tech, in 2020, he was a Visiting Assistant Professor at Purdue University Northwest. His research interests include a variety of topics in image/video processing, machine learning, deep learning, computer vision, and remote sensing. He is also an Associate Editor of the *Signal, Image, and Video Processing* journal (Springer) and *ASPRS Journal Photogrammetric Engineering & Remote Sensing*, and serves as a guest editor/a reviewer for several reputed journals.

VIJAY DEVABHAKTUNI (Senior Member, IEEE) received the B.Eng. degree in electrical and electronics engineering, the M.Sc. degree in physics from the Birla Institute of Technology and Science, Pilani, India, in 1996, and the Ph.D. degree in electronics from Carleton University, Ottawa, Canada, in 2003. He held the competitive Natural Sciences and Engineering Research Council of Canada (NSERC) Postdoctoral Fellowship and spent the tenure researching with Dr. J. W. Haslett with the University of Calgary, Calgary, Canada, from 2003 to 2004. In 2005, he taught with Penn State Behrend. From 2005 to 2008, he held the Canada Research Chair of Computer-Aided High-Frequency Modeling and Design with Concordia University, Montreal, Canada. In 2008, he joined the Department of Electrical Engineering and Computer Science, The University of Toledo, Toledo, as an Associate Professor, and was promoted to a Professor, in 2013. In 2018, he joined Purdue University Northwest, Hammond, as the Chair of the Electrical and Computer Engineering Department, and in 2020, he joined The University of Maine as the Chair of the Electrical and Computer Engineering Department. He secured external funding close to $5M in his research areas (sponsoring agencies include AFOSR, AFRL, CFI, NASA, NIST, NSERC, NSF, ONR, and industry partners). He has authored 250 peer-reviewed papers. His research interests include applied electromagnetics, biomedical applications of wireless sensor networks, computer-aided design, device modeling, image processing, infrastructure monitoring, neural networks, RF/microwave design, unmanned aerial vehicles, and virtual reality. In Canada and USA, he graduated 75 theses students at the M.S. and Ph.D. levels and won student nominated teaching excellence awards. He served as an Associate Editor for the *International Journal of RF and Microwave Computer-Aided Engineering* under the Editor-in-Chief Dr. I. Bahl. He is also a Professional Engineer of the Association of Professional Engineers and Geoscientists of Alberta.