# A Machine Learning Approach for Detecting and Classifying Jamming Attacks Against OFDM-based UAVs

Jered Pawlak, Yuchen Li, Joshua Price, Matthew Wright, Khair Al Shamaileh, Quamar Niyaz, and Vijay Devabhaktuni
Purdue University Northwest
{pawlak7,li3647,price152,wrigh458,kalshama,qniyaz,vjdev}@pnw.edu

## ABSTRACT

In this paper, a machine learning (ML) approach is proposed to detect and classify jamming attacks on unmanned aerial vehicles (UAVs). Four attack types are implemented using software-defined radio (SDR); namely, barrage, single-tone, successive-pulse, and protocol-aware jamming. Each type is launched against a drone that uses orthogonal frequency division multiplexing (OFDM) communication to qualitatively analyze its impacts considering jamming range, complexity, and severity. Then, an SDR is utilized in proximity to the drone and in systematic testing scenarios to record the radiometric parameters before and after each attack is launched. Signal-to-noise ratio (SNR), energy threshold, and several OFDM parameters are exploited as features and fed to six ML algorithms to explore and enable autonomous jamming detection/classification. The algorithms are quantitatively evaluated with metrics including detection and false alarm rates to evaluate the received signals and facilitate efficient decision-making for improved reception integrity and reliability. The resulting ML approach detects and classifies jamming with an accuracy of 92.2% and a false-alarm rate of 1.35%.

## CCS CONCEPTS

• **Security and privacy** → **Denial-of-service attacks**; **Mobile and wireless security**.

## KEYWORDS

Jamming, machine learning (ML), orthogonal frequency division multiplexing (OFDM), software-defined radio (SDR), unmanned aerial vehicles (UAVs).

**ACM Reference Format:**
Jered Pawlak, Yuchen Li, Joshua Price, Matthew Wright, Khair Al Shamaileh, Quamar Niyaz, and Vijay Devabhaktuni. 2021. A Machine Learning Approach for Detecting and Classifying Jamming Attacks Against OFDM-based UAVs. In *3rd ACM Workshop on Wireless Security and Machine Learning (WiseML '21), June 28–July 2, 2021, Abu Dhabi, United Arab Emirates.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3468218.3469049

## 1 INTRODUCTION

Unmanned aerial vehicles (UAVs) have recently found widespread use in civil, military, and scientific applications such as search and rescue missions, merchandise mailing, wildlife tracking, disaster management, climate monitoring, and space exploration [1–4]. The UAV market was estimated at USD 19.3 Billion in 2019 and is projected to reach USD 45.80 Billion by 2025 [5]. The key factors that contribute to the growth of this market include the increasing demand for automation and the rapid advances in enabling technologies. Over the last decade, efforts have been devoted on UAV navigation and control for safe operation and integration [6–17]. However, cybersecurity challenges have not received a similar attention although UAVs are prone to cyberattacks (e.g., jamming, spoofing) that compromise their performance, and in some cases, lead to catastrophic consequences [18]. Thus, as UAVs utilization continues this exponential increase, operation integrity becomes an unavoidable challenge for secure and trustworthy deployments.

Cyberattacks on UAVs are classified into data interception, data manipulation, and denial of service (i.e., jamming). The latter interrupts the communication between the UAV and the controller by transmitting an interference signal at the same frequency band to impose security threats and cease information exchange [19–21]. This interference can be broadcasted wirelessly with the readily available inexpensive software-defined radio (SDR) hardware to interfere with an aircraft's trajectory, potentially resulting in collisions. Hence, affordable jamming detection solutions that also comply with the operation standards of the existing infrastructure are of grave importance. These solutions must facilitate high detection probability and low false alarm and misdetection probabilities.

To prevent cyberattacks on UAVs, secure broadcast authentication [22–26] and secure location verification [27, 28] methods were proposed. The former uses cryptographic and non-cryptographic schemes; whereas the latter attempts to verify the location of UAVs with distance bounding, Kalman filtering, multilateration, group verification, and traffic modeling. Although the reported methods have shown great potential in improving security against jamming attacks, additional hardware and/or software to the existing protocols and time stamping adjustments were major drawbacks that setback their ready acceptance in the foreseeable future. Also, UAVs differ from traditional networks in their increased complexity, channel disparity, range, power requirements, and data properties. Thus, UAV-tailored anti-jamming solutions must balance security improvement, scalability, and compatibility with existing standards.

In this work, the impacts of four jamming types on UAV security are analyzed qualitatively and quantitatively. Jamming range, launch complexity, and severity are evaluated for each type. Then, effective feature engineering is performed to develop machine learning (ML) models for autonomous jamming detection and classification. It is noteworthy to mention that ML was adopted in UAV

*WiseML '21, June 28–July 2, 2021, Abu Dhabi, United Arab Emirates*

ACM ISBN 978-1-4503-8561-9/21/06.
https://doi.org/10.1145/3468218.3469049

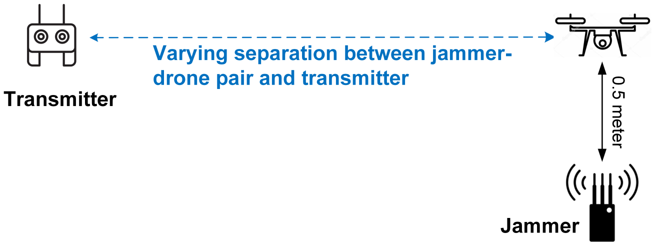
Transmitter
Varying separation between jammer-drone pair and transmitter
0.5 meter
Jammer

**Figure 1: Testing setup to obtain effective jamming range**

applications including object detection, intelligent swarm communication, trajectory optimization, situational awareness, malicious attack mitigation (e.g., eavesdropping), and anti-jamming [29–31].

The proposed approach differs from other reported techniques in the following aspects: 1) In contrast to imposing modifications on the existing protocols for jamming detection [22–28, 32], the readily available radiometric features are used herein to train ML algorithms for detecting the presence and type of jamming. 2) In comparison to the simulation-based attack scenarios reported in [33–36], this work exploits SDR for establishing jamming attacks that facilitate detection/classification with realistic environments and training datasets. 3) This work entails a comparative analysis of different traditional ML models. This analysis conveys evaluation metrics such as detection rate, F-score, and false alarm rate.

The remaining of this paper is organized as follows: Section 2 describes jamming attack types, testing setup, and attack scenarios. Section 3 entails feature extraction and ML training and modeling. Conclusions and future work are provided in Section 4.

## 2 ATTACK TYPES & EXPERIMENTAL SETUP

In this section, the attack scenarios and experimental setup for four jamming attacks are elaborated. Holy Stone HS720E is used as a test drone, which has an unobstructed range of 1000 meters, a maximum transmission power of 16 dBm, and uses IEEE 802.11 orthogonal frequency division multiplexing (OFDM) at 2.4 GHz [37]. USRP B210 SDR and GNURadio software are used to launch the attacks within 40 MHz bandwidth to accommodate all subcarriers.

### 2.1 Types of Jamming Attacks

1) *Barrage Jamming*: In this type, a noise from Gaussian distribution is transmitted across the communication bandwidth to increase the noise level at the targets's receiver. Thus, barrage jamming is often used when the transmission frequency is unknown to the jammer. Although this type is simple to generate, its jamming efficiency reduces with the increase in the signal transmission bandwidth.
2) *Single-tone Jamming*: Here, a high-power interfering signal is transmitted at the center frequency that the target uses for data exchange. This signal is in the form $J(t) = A_j cos(2\pi f_0 t + \theta_j)$, where $A_j$ is the jamming signal amplitude, $f_0$ is the center frequency, and $\theta_j$ is a phase shift. Noise mainly interferes with a single frequency.
3) *Successive-pulse Jamming*: In this type, a pulse-sequence is transmitted to interfere the target's communication, and is given as:

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

where $N_j$ is the number of jamming tones and $T$ is the period, which is set to create a 312.5 KHz frequency spacing between the generated pulses (i.e., subcarrier spacing in IEEE 802.11 OFDM).
4) *Protocol-aware Jamming*: This type entails low interference energy and low detection probability. The jammer is built to simulate the transmitter of the targeted protocol to corrupt its data without interfering other standards in the same bandwidth. This is realized with shot-noise pulses to disrupt the ongoing transmission [38].

### 2.2 Experimental Setup

Two setups are established to evaluate the qualitative and quantitative impacts of the jamming types. The qualitative evaluation is concerned with analyzing severity, complexity, and effective jamming range. The quantitative evaluation tackles the extraction of the radiometric features through exhaustive data collection under different jamming scenarios. Collected data is used to train, validate, and test ML algorithms for jamming detection and classification.
1) *Qualitative Evaluation*: The separation between the jammer (i.e., USRP B210 SDR) and the drone is fixed to 0.5 meter. The separation between the jammer-drone pair and the transmitter is gradually increased for each jamming type in an unobstructed outdoor setup, as shown in Figure 1, to obtain the effective jamming range, defined as complete loss of signal. Table 1 depicts the tested effective range for the jamming types. Results show that barrage jamming has the predominant impact due to distributing interference over all OFDM subcarriers compared 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. Qualitative findings based on attack complexity and severity are given in Table 2 (scale of 1 through 4, where 4 is the highest score). Barrage jamming is the least complex to launch as it does not require extensive knowledge about the communication protocol. Nonetheless it results in the highest severity. Single-tone jamming is relatively simple to launch. However, this type is inefficient in protocols where multiple frequencies or subcarriers are used. Successive-pulse with $N_j = 64$ 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 a jamming device is distributed on pulses such that the interference pulse power is $P_j/N_j$. Hence, successive-pulse jamming has the lowest severity. Protocol-aware jamming has the highest launch complexity as it assumes thorough knowledge of the communication standard. Nevertheless, it has a moderate severity as it launches limited-power interference at the transmission bandwidth to maintain low detection probability.
2) *Quantitative Evaluation*: This evaluation begins with feature extraction for ML training/classification. The goal here is to train a classifier to not only detect jamming, but also to specify its type. First, the transmitter-drone separation is set to 350 meters in an unobstructed outdoor environment. Then, signal features are extracted with no-jamming using B210 SDR and modules in GNURadio (details in Section 3). The same process is repeated in the presence of each of the four jamming types. To this end, another SDR is used as

**Table 1: Effective Jamming Range for each Jamming Type**

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

**Table 2: A Qualitative Analysis for the Jamming Types**

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

a jammer in eight locations $J_i, i = 1, 2, \ldots 8$, around the drone. Then, the same signal features are extracted. This process is performed for four radii $r = 0.5$, 1, 1.5, and 2 meters, as shown in Figure 2.

In this testing setup, 10,071 signal samples are collected under no jamming; whereas 13,494 samples are obtained under jamming presence (i.e., 23,565 overall signal samples). Jamming samples are divided into 3,392, 3,367, 3,378, and 3,357 for barrage, single-tone, successive-pulse, and protocol-aware, respectively, and are given in [39]. Figures 3(a) and 3(b) show the GNURadio flow graphs for launching the jamming attacks and extracting the signal features.

## 3 FEATURE EXTRACTION & ML MODELING

Nine features are extracted to train the ML algorithms for detecting and classifying the jamming attacks. Four features are OFDM-specific and include the *subcarrier length*, *cyclic prefix* (CP) *length*, *subcarrier spacing*, and *symbol time*. The subcarrier length is the number of subcarriers being used, the CP length ensures no symbol overlapping, the subcarrier spacing is the frequency separation between the subcarriers, which is the reciprocal of the symbol time [40]. These features are obtained from the ① *OFDM Estimator* block shown in 3(b) [41]. Two other signal features are extracted using the ② *Energy Detector* block: *average received power* and *threshold* [41]. The threshold returns a binary 0 when no jamming occurs and binary 1 once the *average received power* exceeds a certain level. Finally, three additional features are extracted from the ③ *SNR Estimator Probe* block, which are *signal-to-noise ratio* (SNR), *average signal power*, and *average noise power*. It is paramount to point out here that the *average received power* in ② conveys the noise energy; whereas the *average signal power* in ③ represents the estimated signal power excluding noise power. The nine extracted features are used for training and developing ML models for jamming detection and classification. These features are also analyzed for dimensionality reduction. It is found that the (*symbol time, subcarrier length*) and (*threshold, average noise power*) feature pairs are highly correlated. Therefore, the ML models are developed considering two additional cases where 1) *symbol time* and 2) *symbol time* and *average noise power* are eliminated from the datasets. Table 3 lists the features for each case.

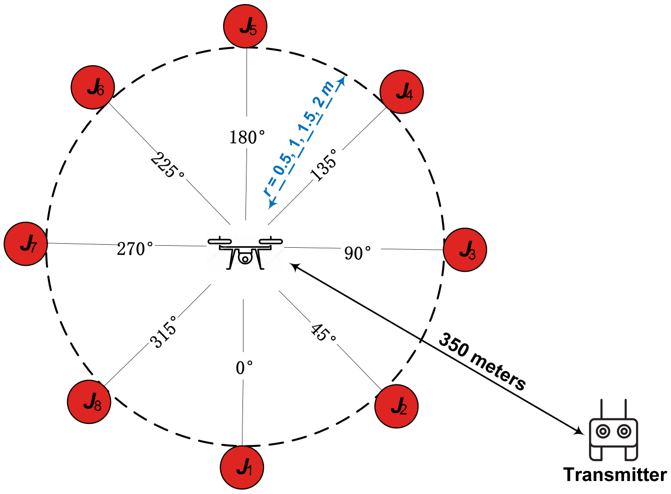

**Figure 2: The setup for extracting signal features under no-jamming and jamming scenarios considering different jammer locations**

**Table 3: List of Selected Features for Cases 1-3**

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

The ML models are built using Scikit-learn—a Python-based ML library. Six conventional algorithms are used for developing the models; specifically, Decision Tree (DT), K-Nearest Neighbors (KNN), Logistic Regression (LR), Multi-layer Perceptron (MLP), Naive Bayes (NB), and Random Forest (RF). Classification performance metrics for model evaluation are given in Eqn. (2) and include detection rate (DR), precision, recall, F-score (FS), and false-alarm rate (FAR). The DR of a model is the percentage of correctly detected samples over the total samples in the dataset. The precision is 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). The 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). The F-score is computed as a harmonic mean of precision and recall. Finally, the FAR is the number of false positive samples divided by the sum of false positive and true negative samples predicted by the model.

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

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

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

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

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

Two- and five-class ML models are developed for the three cases listed in Table 3. In the two-class model, the algorithm predicts whether jamming is launched or not; whereas the five-class model

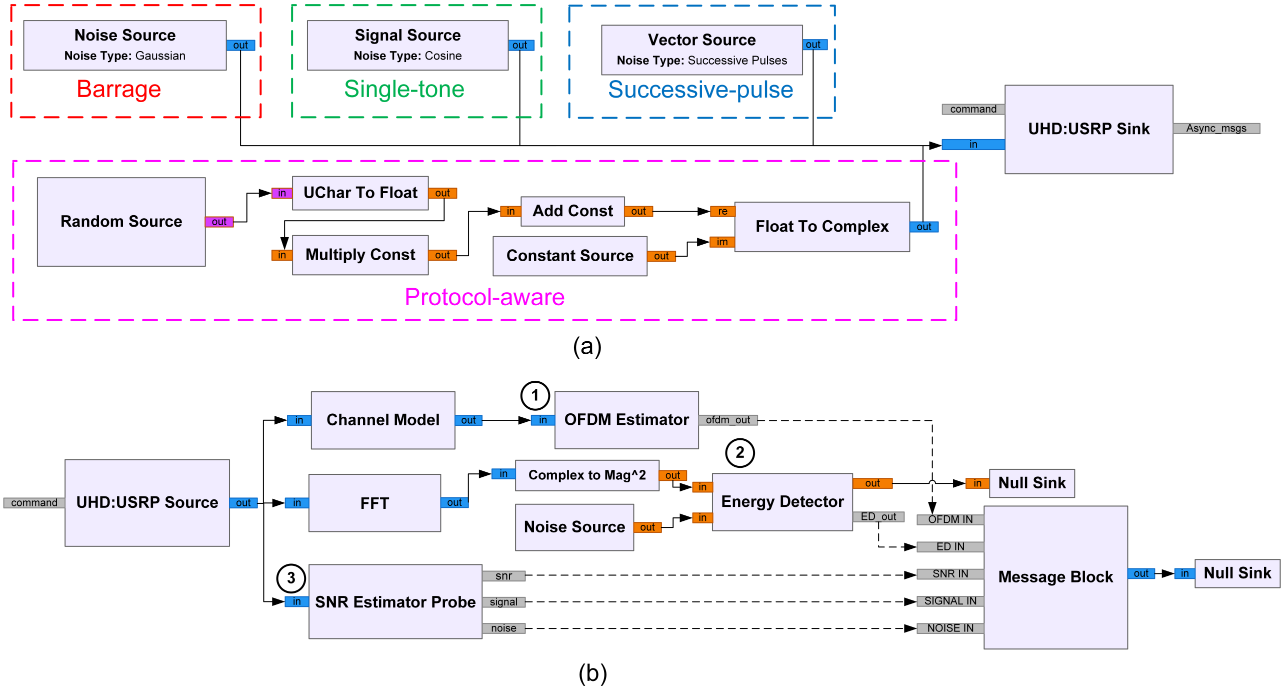

**Figure 3: Abbreviated GNURadio flow graph for (a) launching the jamming attacks and (b) extracting the radiometric features**

**Table 4: Performance Metrics for Two-class and Five-class Jamming Detection Models (VA: Validation Accuracy, DR: Detection Rate, FS: F-score)**

| | **Performance metrics for five-class models** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Case 1: Nine Features** | | | **Case 2: Eight Features** | | | **Case 3: Seven Features** | | |
| **ML Classifier** | **VA (in %)** | **DR (in %)** | **FS** | **VA (in %)** | **DR (in %)** | **FS** | **VA (in %)** | **DR (in %)** | **FS** |
| 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 |
| 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 |
| 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 |
| 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 |
| **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** |
| 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 |
| | **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** |
| 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 |
| 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 |
| 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 |
| **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** |
| 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 |

detects jamming presence and classifies its type (i.e., barrage, single-tone, successive-pulse, and P-aware). The dataset is split in training (70%) and testing (30%). 10-fold cross-validation is used during training and validation stages. Once a model is trained, the evaluation is performed on the test set; and the DR, F-score, and FAR are computed. It is noteworthy to point out that grid search is used to find the optimal hyper-parameters for each algorithm. Table 4 depicts the performance of the developed classifiers for the two- and five-class models. All classifiers in the two-class models achieved almost 100% validation accuracy (VA) and DR in classifying records into "no-jamming" or "presence of jamming". In addition, it is found that seven features are sufficient for developing an efficient and trustworthy two-class ML model. On the other hand, the five-class models show that RF has the highest VA of 91.80%, 92.20%, and 86.23% in cases 1, 2, and 3, respectively. Furthermore, the RF model 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. It is also noted that

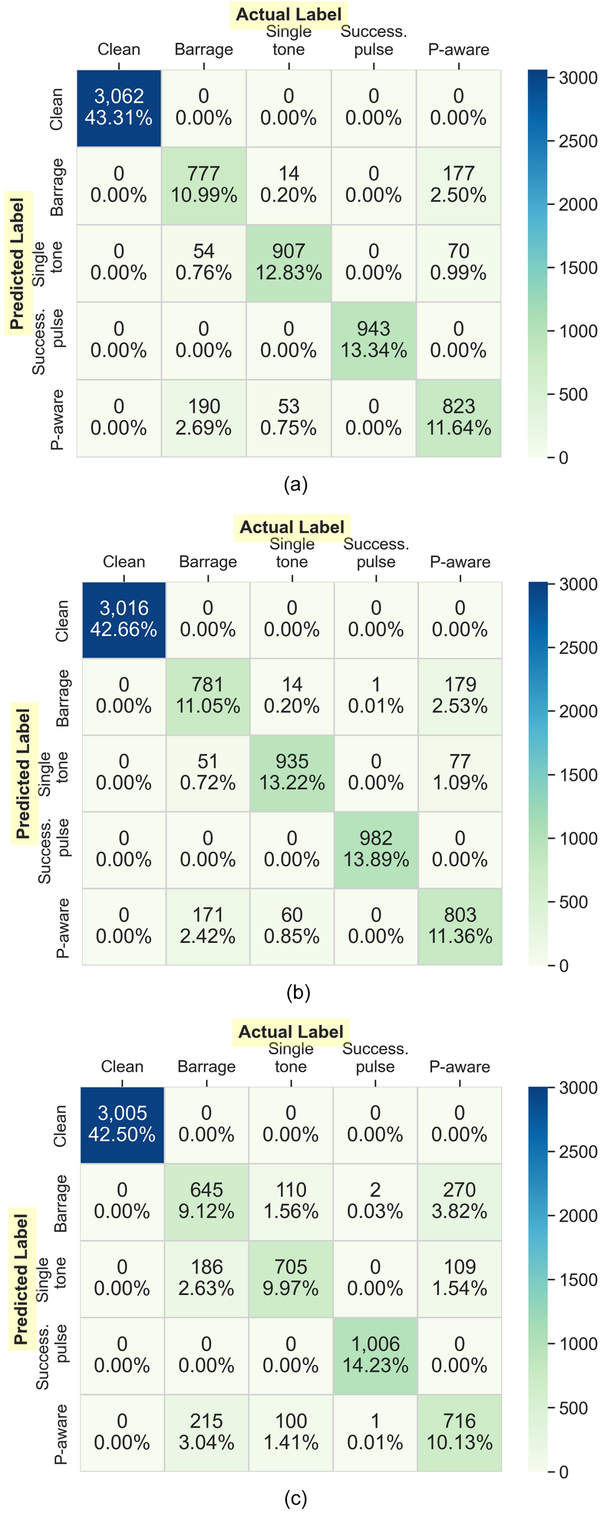

**Figure 4: Confusion matrix of the five-class RF model for (a) nine features, (b) eight features, and (c) seven features**

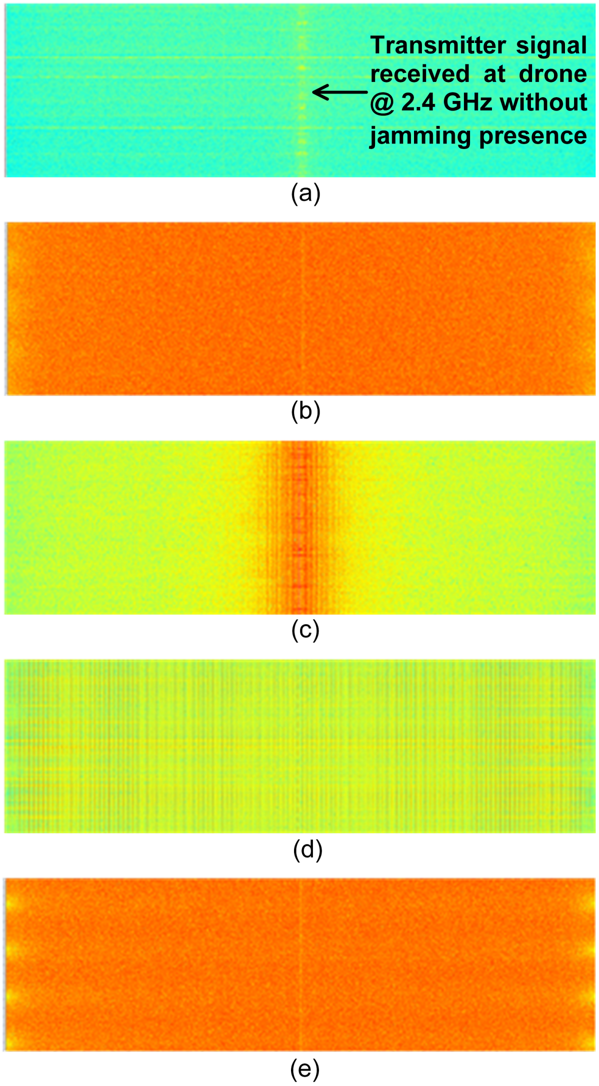

**Figure 5: Waterfall images for (a) clear signal, (b) barrage, (c) single-tone, (d) successive-pulse, and (e) P-aware jamming**

eliminating the symbol time from the features set (i.e., case 2) has a marginal effect in improving classification. However, eliminating the average noise power (i.e., case 3) has a significant impact on the overall performance. Figures 4(a)-(c) demonstrate the confusion matrices of the five-class RF model for each case. It is clearly shown that none of the clean (i.e., non-jamming) records are mis-classified as jamming records. Rather, mis-classification occurs among the jamming types. The weighted FAR values are computed from the confusion matrices to be 1.35% for case 1, 1.33% for case 2, and 2.38% for case 3. Figure 5 depicts the waterfall plot of a no jamming scenario together with other plots with jamming presence. These plots highlight the unique spectral characteristics for each scenario, with some similarity between barrage and protocol-aware jamming. The similarity between these two types is thought to be the main reason for their high mis-classification (i.e., classification confusion) in comparison to the other types. Finally, it is noteworthy to point out that there is no false-alarm in the two-class models regardless of the number of features used during training and validation.

## 4 CONCLUSION

In this paper, an ML approach is proposed to detect and classify four types of jamming attacks on UAVs. Each attack is implemented using B210 SDR and launched against a drone that uses OFDM communication to qualitatively analyze its impacts considering severity, complexity, and jamming range. Then, an SDR is used in proximity to the drone in systematic testing scenarios to record signal features including key OFDM parameters, threshold, signal power, noise power, and SNR. These features are used to train six algorithms for jamming detection/classification. All algorithms are validated quantitatively with metrics including detection and false alarm rates, and showed that jamming is detected and classified with 92.2% confidence. Future work will entail implementing a comprehensive complexity analysis for the developed classifiers, exploring more jamming types (e.g., deceptive, reactive), incorporating maximum-likelihood-based classification and advanced SNR probing, extracting features for jamming detection/classification with various UAV altitudes, and investigating UAV-specific anti-jamming solutions (e.g., flight scheduling, path optimization).

## ACKNOWLEDGMENT

This work is funded by National Science Foundation, Secure and Trustworthy Cyberspace program, under Award # 2006662.

## REFERENCES

[1] M. Messinger and M. Silman. Unmanned aerial vehicles for the assessment and monitoring of environmental contamination: An example from coal ash spills. *Environmental pollution*, 218:889–894, 2016.

[2] A. Bhardwaj, L. Sam, F. Martín-Torres, R. Kumar, et al. Uavs as remote sensing platform in glaciology: Present applications and future prospects. *Remote sensing of environment*, 175:196–204, 2016.

[3] R. Allison, J. Johnston, G. Craig, and S. Jennings. Airborne optical and thermal remote sensing for wildfire detection and monitoring. *Sensors*, 16(8):1310, 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. *Journal of Field Robotics*, 33(3):290–321, 2016.

[5] Unmanned aerial vehicles UAV market. https://www.marketsandmarkets.com/Market-Reports/unmanned-aerial-vehicles-uav-market-662.html Accessed on April 10, 2021.

[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 *2016 IEEE ANDESCON*, pages 1–4. IEEE, 2016.

[7] P. Oettershagen, T. Stastny, T. Mantel, A. Melzer, K. Rudin, P. Gohl, G. Agamennoni, K.s Alexis, and R. Siegwart. Long-endurance sensing and mapping using a hand-launchable solar-powered uav. In *Field and Service Robotics*, pages 441–454. Springer, 2016.

[8] J. Braga, H. Velho, G. Conte, P. Doherty, and É. Shiguemori. An image matching system for autonomous uav navigation based on neural network. In *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1–6. IEEE, 2016.

[9] J. Tiemann, F. Schweikowski, and C. Wietfeld. Design of an uwb indoor-positioning system for uav navigation in gnss-denied environments. In *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–7. IEEE, 2015.

[10] M. Mullins, M. Holman, K. Foerster, N. Kaabouch, and W. Semke. Dynamic separation thresholds for a small airborne sense and avoid system. In *AIAA Infotech @ Aerospace (I@A) Conference*, page 5148, 2013.

[11] M. Mullins, K. Foerster, N. Kaabouch, and W. Semke. Incorporating terrain avoidance into a small uas sense and avoid system. In *Infotech@ Aerospace 2012*, page 2504. 2012.

[12] M. Mullins, K. Foerster, N. Kaabouch, and W. Semke. A multiple objective and behavior solution for unmanned airborne sense-and-avoid systems. *AUVSI's Unmanned Systems North America*, 2012.

[13] F. Martel, M. Mullins, W. Semke, N. Kaabouch, et al. Cooperative miniature collision avoidance system flight testing for small unmanned aircraft systems. In *Proceedings of AUVSI conference*, 2011.

[14] M. Mullins, K. Foester, and N. Kaabouch. Traffic alerting system for manned-unmanned aircraft airspace conflicts. In *ND EPSCoR/IDeA State Conference*. 2017.

[15] K. Foerster, B. Whitney, J. Hahn, N. Kaabouch, and W. Semke. A health monitoring system for uas utilizing a miniature airborne sense and avoid system. In *AIAA Infotech@ Aerospace Conference*, page 4654, 2013.

[16] H. Reyes, N. Gellerman, and N. Kaabouch. A cognitive radio system for improving the reliability and security of uas/uav networks. In *2015 IEEE Aerospace Conference*, pages 1–9, 2015.

[17] H. Reyes, N. Kaabouch, W. Semke, and S. Salle. Fuzzy logic method for link loss detection during unmanned aerial vehicle flights. In *Infotech@ Aerospace 2012*, page 2574. 2012.

[18] Drones are quickly becoming a cybersecurity nightmare. https://threatpost.com/drones-breach-cyberdefenses/143075 Accessed on April 10, 2021.

[19] Security analysis of unmanned aircraft systems. http://dl.comp.nus.edu.sg/handle/1900.100/6167 Accessed on April 21, 2021.

[20] D. He, S. Chan, and M. Guizani. Communication security of unmanned aerial vehicles. *IEEE Wireless Communications*, 24(4):134–139, 2016.

[21] Jeff C. The threat of GPS jamming: The risk to an information utility, 2014. https://rntfnd.org/wp-content/uploads/Exelis-GPS-Vulnerability-Assessment-February2014.pdf, Last Accessed on April 21, 2021.

[22] M. Strohmeier, V. Lenders, and I. Martinovic. On the security of the automatic dependent surveillance-broadcast protocol. *IEEE Communications Surveys Tutorials*, 17(2):1066–1087, 2015.

[23] M. Manesh and N. Kaabouch. Analysis of vulnerabilities, attacks, countermeasures and overall risk of the automatic dependent surveillance-broadcast (ads-b) system. *International Journal of Critical Infrastructure Protection*, 19:16–31, 2017.

[24] K. D. Wesson, T. Humphreys, and B. Evans. Can cryptography secure next generation air traffic surveillance?, 2021. http://users.ece.utexas.edu/~bevans/papers/2015/nextgen/ Technical Report, Accessed on April 21, 2021.

[25] C. Giannatto Jr. Challenges of implementing automatic dependent surveillance broadcast in the nextgen air traffic management system. 2015.

[26] B. Danev, H. Luecken, S. Capkun, and K. El Defrawy. Attacks on physical-layer identification. In *Proceedings of the third ACM conference on Wireless network security*, pages 89–98, 2010.

[27] S. Brands and D. Chaum. Distance-bounding protocols. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 344–359. Springer, 1993.

[28] B. Xiao, B. Yu, and C. Gao. Detection and localization of sybil nodes in vanets. In *Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks*, pages 1–8, 2006.

[29] P. Bithas, E. Michailidis, N. Nomikos, D. Vouyioukas, and A. Kanatas. A survey on machine-learning techniques for uav-based communications. *Sensors*, 19(23):5170, 2019.

[30] Q. Wu, H. Wang, X. Li, B. Zhang, and J. Peng. Reinforcement learning-based anti-jamming in networked uav radar systems. *Applied Sciences*, 9(23):5173, 2019.

[31] X. Lu, L. Xiao, C. Dai, and H. Dai. Uav-aided cellular communications with deep reinforcement learning against jamming. *IEEE Wireless Communications*, 27(4):48–53, 2020.

[32] M. Sliti, W. Abdallah, and N. Boudriga. Jamming attack detection in optical uav networks. In *2018 20th International Conference on Transparent Optical Networks (ICTON)*, pages 1–5. IEEE, 2018.

[33] D. Karagiannis and A. Argyriou. Jamming attack detection in a pair of rf communicating vehicles using unsupervised machine learning. *Vehicular Communications*, 13:56–63, 2018.

[34] L. Mokdad, J. Ben-Othman, and A. Nguyen. Djavan: Detecting jamming attacks in vehicle ad hoc networks. *Performance Evaluation*, 87:47–59, 2015.

[35] A. Nguyen, L. Mokdad, and J. Ben Othman. Solution of detecting jamming attacks in vehicle ad hoc networks. In *Proceedings of the 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems*, pages 405–410, 2013.

[36] J. Grover, N. Kumar Prajapati, V. Laxmi, and M. Gaur. Machine learning approach for multiple misbehavior detection in vanet. In *International conference on advances in computing and communications*, pages 644–653. Springer, 2011.

[37] Manual & Drivers. http://holystone.com/en/supports/Drivers.html Accessed on April 21, 2021.

[38] A. Hussain, N. Saqib, U. Qamar, M. Zia, and H. Mahmood. Protocol-aware radio frequency jamming in wi-fi and commercial wireless networks. *Journal of communications and networks*, 16(4):397–406, 2014.

[39] UAV Jamming Dataset. https://drive.google.com/file/d/1z9vyEP2Z2pybsGp2IncGYBwpoVUPSG_3/view?usp=sharing Accessed on May 28, 2021.

[40] Y. Cho, J. Kim, W. Yang, and C. Kang. *Introduction to OFDM*, pages 111–151. 2010.

[41] S. Müller and C. Richardson. GitHub - gnuradio/gr-inspector: Signal Analysis Toolbox for GNU Radio. https://github.com/gnuradio/gr-inspector Accessed on April 21, 2021.