Classification and Source Location Indication of Jamming Attacks Targeting UAVs via Multi-output Multiclass Machine Learning Modeling

M. Alkhatib [(1)], M. McCormick [(1)], L. Williams [(1)], A. Leon [(1)], L. Camerano [(1)], K. Al Shamaileh [(1)], V. Devabhaktuni [(2)], and N. Kaabouch [(3)]

[(1)] Electrical and Computer Engineering Department, Purdue University Northwest, Hammond 46323, IN, USA
[(2)] Electrical and Computer Engineering Department, Illinois State University, Normal 61761, IL, USA
[(3)] School of Electrical Engineering and Computer Science, University of North Dakota, Grand Forks 58202, ND, USA

E-mail: kalshama@pnw.edu

***Abstract*—This paper introduces machine learning (ML) as a solution for the detection and range localization of jamming attacks targeting the global positioning system (GPS) technology, with applications to unmanned aerial vehicles (UAVs). Different multi-output multiclass ML models are trained with GPS-specific sample datasets obtained from exhaustive feature extraction and data collection routines that followed a set of realistic experimentations of attack scenarios. The resulting models enable the classification of four attack types (i.e., barrage, single-tone, successive-pulse, protocol-aware), the jamming direction, and the distance from the jamming source by yielding a detection rate (DR), misdetection rate (MDR), false alarm rate (FAR), and F-score (FS) of 98.9%, 1.39%, 0.28%, and 0.989, respectively.**

***Index Terms*—Global positioning system (GPS), jamming classification, jamming localization, machine learning (ML), unmanned aerial vehicles (UAVs).**

I. Introduction

UNMANNED aerial vehicles (UAVs) have been utilized recently in many applications, including search and rescue missions, surveillance, construction, delivery of goods, agriculture, and smart cities [1–3]. This increased exploitation of UAVs incentivizes attackers to disturb their operation with cyberattacks of irreversible consequences, featured by compromising sensitive information (e.g., payload, technology) as well as damaging private properties and public infrastructure. For example, a U.S RQ-170 surveillance drone was captured by the Iranian forces in 2011 with the use of cyberwarfare. Later in 2012, Iran announced the hacking of the drone controls and the building of a similar copy [4].

With the readily available and affordable software-defined radio (SDR) units, cyberattacks can conveniently be launched for targeting the UAV's onboard global positioning system (GPS) module. Location spoofing and jamming are common attacks; the former is concerned with transmitting a falsifying GPS-like signal to redirect targets toward a desired destination, whereas the latter entails launching an interference to block the authentic GPS signal to the target's impact location awareness. Detecting and anticipating the source of such attacks facilitate timely actions and countermeasures against attackers. Hence, this paper addresses GPS jamming detection/classification and suggests an approximate location of the source w.r.t to the target (i.e., UAV).

Various approaches were examined recently in literature to detect jamming presence. For instance, jamming classification according to received signal strength (RSS) was proposed in [5]. However, this approach was evaluated within a simulation framework that overlooks other factors influencing practical RSS measurements (e.g., channel characteristics), leading to a compromised overall accuracy. The use of machine learning (ML) models trained with the in-phase and quadrature-phase signal components was explored in [6]. Although this approach has shown an acceptable performance, it utilized a dataset that failed to capture other GPS-related features; not to mention that it involved a two-stage detection and classification process that often introduces an additional computational overhead. A "return to-home" jamming mitigation solution based on estimating the angle of arrival at the jammed UAV was investigated in [7]. Other solutions benefited from null steering and adaptive notch filtering techniques [8, 9]. Nevertheless, such solutions require expensive sophisticated hardware (e.g., antenna arrays) and introduce a high computational complexity. Jamming localization techniques have also received significant attention lately. Such techniques span the use of RSS, carrier-to-noise density power ratio, and network topology information [10–14]. Nevertheless, these techniques necessitate knowledge of target location and require a large number of nodes as well as sensory hardware components.

This work presents a single-stage hybrid approach to detect and localize jammers, providing the following advantages:

1. The solution proposed herein offers multi-output multiclass ML models that enable concurrent jamming classification and range localization.
2. This solution exploits a set of features obtained from the existing onboard GPS receiver module. Therefore, no additional hardware is needed.
3. The training and validation datasets for developing the ML models convey feature samples extracted from measurement setups of staged scenarios with and without the presence of attacks.
4. This solution does not assume location awareness of the jammed UAV or require multiple localizing nodes.

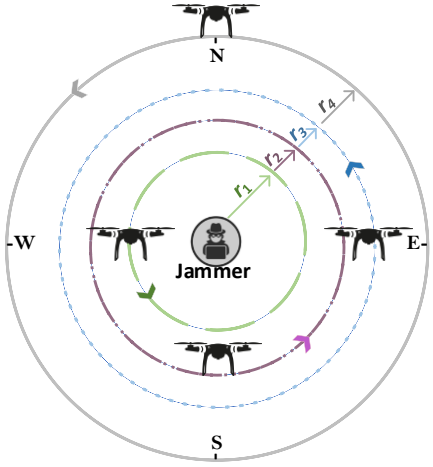

Fig. 1. The experimental setup for capturing signal features with and without the presence of four jamming attack types. Dashed, dotted-dashed, dotted, and solid lines represent circles with radii $r_1 = 3$, $r_2 = 10$, $r_3 = 17$, and $r_4 = 24$ meters, respectively.

The remaining of this article is organized as follows: Section II presents the experimental setup for extracting signal features with and without the presence of attacks. Section III details the preprocessing of the resulting samples dataset according to correlation and importance. It then elaborates on the development and evaluation of different ML models for jamming detection and localization. Finally, conclusions are given in Section IV.

II. Experimental Setup

Fig. 1 depicts the experimental configuration for extracting signal features and collecting samples. It comprises four circles centered at the jamming source, each is with a unique radius (i.e., 3, 10, 17, and 24 meters). Attacks are designed with GNURadio software and are launched with a B-210 SDR from National Instruments. Fig. 2 illustrates a simplified flow graph for launching four jamming attack types; namely, barrage, single-tone, successive-pulse, and protocol-aware. Barrage jamming is concerned with launching an interference that spans a particular bandwidth, and is especially useful when the transmission band is unknown to the jammer. Single-tone is considered effective as long as interference is launched at the same target communication frequency. Successive-pulse is created with launching a train of discrete pulses within the target transmission bandwidth. Lastly, protocol-aware is focused on launching a low-power pulsating interference to minimize the probability of detection.

TABLE I
DIFFERENT GAIN LEVELS AS A FUNCTION OF EFFECTIVE JAMMING RANGE

Gain (dB)	*Attenuator* (dB)	*Jamming Range* (m)
45	10	5.5
50	10	13.5
55	10	15.0
60	10	27.0

The target UAV is an open-source drone from COEX, featuring a u-blox M8 GPS receiver and a PX4 flight controller that is capable of logging multiple GPS features during the flight. The experimental setup is performed in two phases: *Phase* 1 entailed confining the jamming to a range where all types are deemed ineffective in order to prevent interference with other nearby electronic devices. To test this range, barrage jamming is launched from an SDR at the lowest gain (i.e., specified in GNURadio) considering fixed attenuation settings (i.e., obtained with 50-ohm 10-dB attenuator). Here, barrage jamming is selected for its highest severity in covering larger ranges compared to the other types. Then, gain settings are varied such that GPS reception (i.e., tested with a Garmin satellite receiver) is restored at a jammer-receiver distance of 27 m. Table I shows the different gain settings and the associated effective jamming range. In *Phase* 2, experiments are carried on by logging samples of authentic features as well as others undergoing jamming attacks via the aforementioned drone's onboard GPS module. Each jamming type is launched considering all circles, and for each circle, the drone is placed at four locations, one at a time. These locations are at the north, south, east, and west of the jammer position. To collect samples leading to balanced datasets that account for the diversity of satellite constellations and physical layer conditions, data logging is performed over four days such that sets of authentic and jammed samples are collected each day as illustrated in Table II. A total of 14 features are logged in this experiment with an overall 17,960 samples per feature. Tables III and IV summarize the logged features and the distribution of the collected samples, suggesting a balanced set consisting of 9,904 and 8,056 attack and authentic samples, respectively.

TABLE II
FEATURE LOGGING ROUTINE OVER FOUR DAYS

	r_1	r_2	r_3	r_4
Day 1	P-aware	Single-tone	Succ.-pulse	Barrage
Day 2	Single-tone	Succ.-pulse	Barrage	P-aware
Day 3	Succ.-pulse	Barrage	P-aware	Single-tone
Day 4	Barrage	P-aware	Single-tone	Succ.-pulse

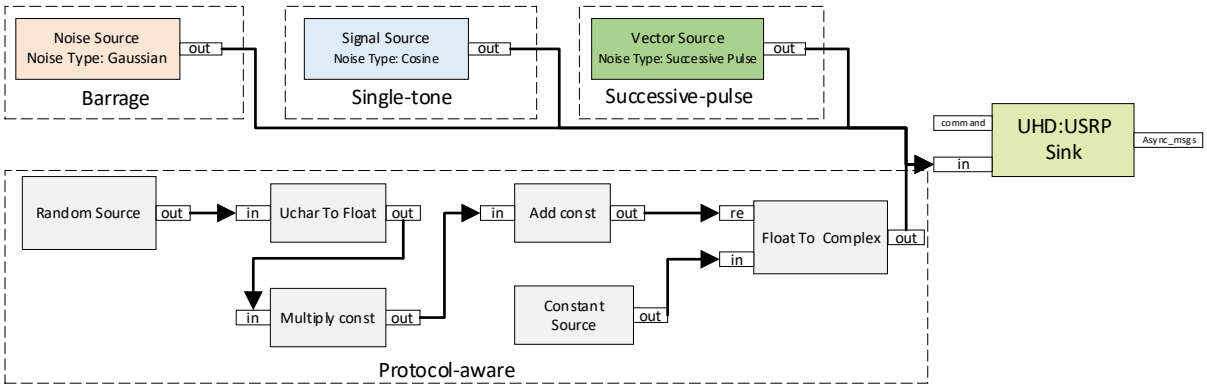

Fig. 2. Simplified GNURadio flow graph for launching the jamming attacks [15].

TABLE III
THE EXTRACTED GPS FEATURES

Extracted Feature	Short Description	Unit
s_var	GPS speed accuracy estimate	m/s
c_var	GPS course accuracy estimate	Radians
eph	GPS horizontal position accuracy	Meters
epv	GPS vertical position accuracy	Meters
hdop	Horizontal dilution of precision	–
vdop	Vertical dilution of precision	–
noise	GPS noise per millisecond	dB
jam	Indication of jamming occurrence	–
vel_m_s	GPS ground speed	m/s
vel_n	GPS North velocity	m/s
vel_e	GPS East velocity	m/s
vel_d	GPS Down velocity	m/s
COG	Course over ground	Radians
sat	Number of satellites	–

III. CLASSIFIERS DEVELOPMENT AND TESTING

Feature samples are combined into a dataset and output Labels 1–3 are created, corresponding to circle radius at which samples are collected, direction of the jammer w.r.t the drone, and jamming type, respectively. Each output Label has the following five classes:

Label 1: < $r_{1–4}$, *N*>
Label 2: < *south*, *north*, *east*, *west*, *N*>,
Label 3: <*barrage*, *single-tone*, *p-aware*, *success.-pulse*, *N*>

where *N* refers to authentic reception (i.e., no jamming). It is paramount to point out that Label 1 yields a separation distance (in meters), with r_i being the jammer-receiver separation. Dataset preprocessing is performed by analyzing the correlation of features using Spearman algorithm, which assumes nonlinearity among sample points. The resulting heatmap of this analysis is shown in Fig. 3, suggesting (*eph*, *epv*), (*eph*, *s_var*), and (*sat*, *hdop*) as strongly correlated feature pairs, based on a threshold of |0.8| for high correlation.

TABLE IV
DISTRIBUTION OF THE COLLECTED SAMPLES

	r_1	r_2	r_3	r_4	*Total Samples*
Barrage	565	1,165	530	563	2,823
Single-tone	544	526	528	550	2,148
P-aware	528	617	584	547	2,276
Succ.-pulse	1,012	532	546	567	2,657
Clean	2,000	2,020	2,010	2,026	8,056

Such an analysis is followed by a feature importance study utilizing the mean decrease in impurity method for determining the features to be discarded. This study is carried out for each of the three output labels as presented in Figs. 4(*a*)-(*c*). According to the resulting correlated pairs and their relative importance characterized in Fig. 4, the *eph* and *sat* features are removed from the dataset. Finally, a standard scaling is applied to all the samples, such that $x_{ij}' = (x_{ij} - \mu_j)/\sigma_j$, where x_{ij}' is the scaled i^{th} sample of the j^{th} feature, and μ_j and σ_j are the mean and standard deviation of the sample values within the j^{th} feature, respectively.

The developed dataset is used for training and evaluating a variety of ML models; namely, random forest (RF), k-nearest neighbor (KNN), multi-layer perceptron (MLP), logistic regression (LR), decision tree (DT), support vector machine (SVM), and naïve Bayes (NB). The training and evaluation components are performed on a MacBook Air laptop with an M1 CPU running at @ 3.2 GHz and 8 GB of DDR4-4266 MHz memory. A 10-fold cross-validation process is performed with 70% and 30% of the dataset samples allocated for training and testing, respectively. Random search is applied for optimizing each model, yielding the hyperparameters presented in Table V. The developed dataset used in this work can be found at [16].

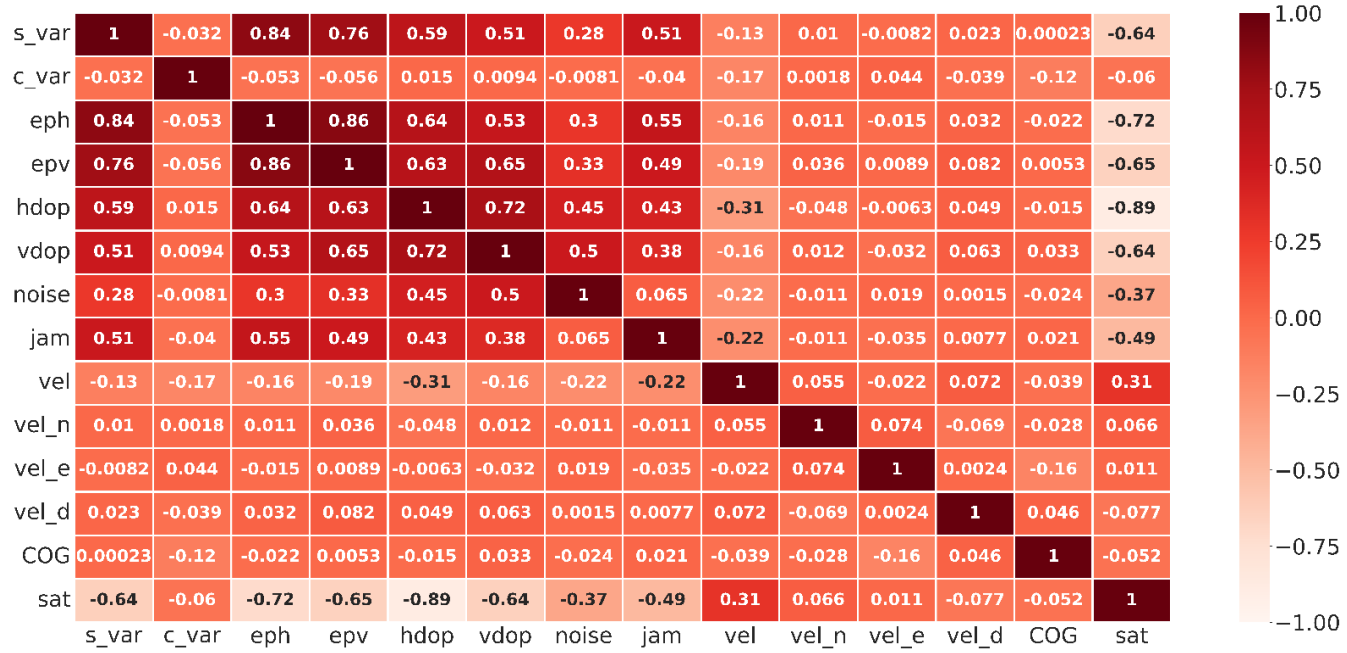

Fig. 3. Correlation heatmap of GPS features obtained with Spearman algorithm.

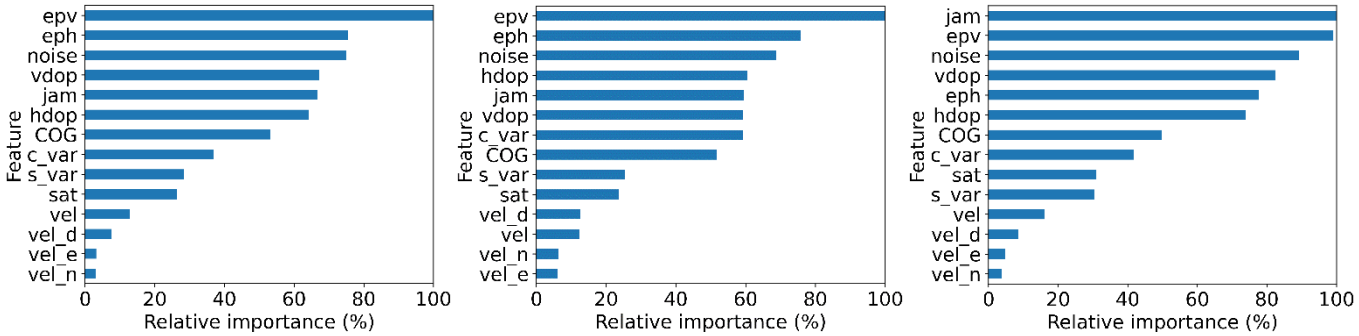

Fig. 4. Relative importance of features with respect to (*a*) output Label 1, (*b*) output Label 2, and (*c*) output Label 3.

TABLE V
OPTIMIZED HYPERPARAMETERS FOR THE ADOPTED ML MODELS W.R.T LABEL 1

Category	Classifier	Parameter
Ensemble	**RF**	Quality of split criterion: Log loss Max. tree depth: 21 Min. number of samples at a leaf node: 33 Min. number of samples to split a node: 183 Number of trees: 129 Cost-complexity pruning parameter: 9.3E–3
Instance	**KNN**	Leaf size: 48 Number of neighbors: 12 Weight function: Distance Nearest neighbor comp. algorithm: Brute Distance metric: Euclidean Power parameter for distance metric: 4
Discriminative	**SVM**	Norm used in penalty: L2 Loss function: Squared Hinge Dual optimization algorithm: True Max. number of iterations: 1117 Regularization parameter: 9.59
Regularization	**LR**	Optimization: Newton conjugate gradient Norm used in penalty: None Regularization parameter: 1.773 Max. number of iterations: 119
Tree	**DT**	Quality of split criterion: Log loss Max. tree depth: 21 Min. number of samples at a leaf node: 33 Min. number of samples to split a node: 183 Node split strategy: Best Cost-complexity pruning parameter: 9.3E–3
Neural network	**MLP**	Optimization:Limited-memory Broyden–Fletcher–Goldfarb–Shanno Hidden layers and neurons: 453,207, and 374 Activation function: Relu Max. number of iterations: 939 L2 regularization term strength: 2.18E–5 Early stopping: True
Probabilistic	**Gaussian NB**	Smoothing stability parameter: 1.11E–10

The performance of the developed models is evaluated using the detection rate (DR), precision, recall, F-score (FS), false alarm rate (FAR), and misdetection rate (MDR). The DR is used to evaluate the percentage of samples that have been accurately classified. Precision is a measure of the classifier's ability to correctly categorize negative samples (i.e., authentic) as negatives and positive samples (i.e., attacks) as positives. The recall, on the other hand, assesses the classifier's in accurately predicting all positive samples. The FAR estimates the likelihood of detecting false positives, while the MDR provides the likelihood of failing to detect an attack. These metrics are obtained with the true positive (*TP*), which represents the positive samples predicted as positive, true negative (*TN*), which corresponds to negative samples predicted as negative, false positive (*FP*), which denotes negative samples predicted as positive, and false negative (*FN*), which indicates positive samples predicted as negative, and are expressed as:

$$\text{DetectionRate (DR)} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1.a}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1.b}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{1.c}$$

$$\text{F-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1.d}$$

$$\text{FalseAlarmRate (FAR)} = \frac{FP}{FP + TN} \tag{1.e}$$

$$\text{MisdetectionRate (MDR)} = \frac{FN}{TP + FN} \tag{1.f}$$

The scores for all evaluation metrics are given in Table VI. MLP exhibited the optimum overall performance for accurately capturing output Labels 1–3, averaging a DR, FS, FAR, and MDR of 98.9%, 0.989, 0.28%, and 1.39%, respectively, followed by KNN, which averaged a DR of 97.8%, FS of 0.978, FAR of 0.61% and MDR of 2.78%. On the other hand, NB yielded the worst performance among all models.

The prediction time (PT) for each model is also calculated and recorded in Table VI. DT, SVM, and LR models resulted in the lowest PT of 1.26 ms, 2.23 ms and 2.33 ms, respectively, at the expense of their DRs. Furthermore, it is noteworthy to point out that the excellent detection quality of the KNN model is associated with the highest PT (i.e., 2296 ms) due to its characteristics in searching the entire training dataset to determine the nearest neighbors during prediction. These PTs are computed using all samples in the testing dataset (i.e., 5,388 samples), leading to 50 μs per sample, based on the PT of MLP model. With such a high prediction accuracy and low PT, real-time jamming detection, classification, and range localization can be achieved without resorting to additional hardware resources.

TABLE VI
ML CLASSIFIERS DISTANCE EVALUATION SCORES (L_1: OUTPUT LABEL 1, L_2: OUTPUT LABEL 2, L_3: OUTPUT LABEL 3)

Model	DR (%)			Precision (%)			FS			FAR (%)			MDR (%)			PT (ms)
	L_1	L_2	L_3	L_1	L_2	L_3	L_1	L_2	L_3	L_1	L_2	L_3	L_1	L_2	L_3	
RF	94.71	89.79	95.38	94.83	90.03	95.44	0.946	0.895	0.953	1.63	2.87	1.39	7.92	15.6	6.6	1056
KNN	97.17	97.91	98.32	97.18	97.91	98.32	0.971	0.979	0. 983	0.76	0.60	0.49	3.85	2.6	1.9	2296
MLP	**98.30**	**99.05**	**99.34**	**98.31**	**99.05**	**99.34**	**0.983**	**0.991**	**0.993**	**0.42**	**0.24**	**0.20**	**2.39**	**1.0**	**0.80**	**272**
LR	70.04	63.14	74.91	68.38	60.75	74.91	0.677	0.590	0.735	9.10	10.8	7.59	43.43	53.20	33.7	2.33
DT	90.08	75.92	91.94	90.21	76.35	92.17	0.900	0.756	0.919	2.45	5.76	2.04	13.83	34.18	11.0	1.26
SVM	65.97	57.14	68.74	70.68	55.78	70.82	0.604	0.494	0.630	11.8	13.98	10.84	51.11	62.66	44.92	2.23
NB	59.56	51.78	61.62	65.28	42.04	65.49	0.568	0.414	0.602	11.59	15.43	11.06	65.10	70.07	52.47	6.35

IV. CONCLUSION

To conclude, a real-time jamming detection, classification, and location approximation solution with applications to UAVs is proposed. Experimental scenarios were established for extracting signal features with and without the presence of four types of jamming attacks, resulting in a dataset of 17,960 samples. This dataset was preprocessed for features correlation and importance to reduce its dimensionality. Then, 70% of the overall samples were used for training seven multi-class multi-output ML models, whereas the remaining 30% were utilized for testing. The MLP model had the optimum performance, characterized by an average DR, FS, FAR, and MDR of 98.9%, 0.989, 0.28%, and 1.39%, respectively, in conjunction with a PT of 50 µs/sample. For future work, the MLP model will be integrated with a reinforcement learning environment to address the mitigation of jamming via flightpath rescheduling in considering static and mobile jammers. Such an environment will exploit the direction of jamming in efforts to navigate away from the jamming source.

ACKNOWLEDGMENT

This research is funded by the National Science Foundation, Secure and Trustworthy Cyberspace under Award no. 2006662.

REFERENCES

[1] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Communications Surveys & Tutorials*, vol.18, no.4 pp.2624–2661, 2016.

[2] P. Radoglou-Grammatikis, P. Sarigiannidis, T. Lagkas, and I. Moscholios, "A compilation of UAV applications for precision agriculture, "*Computer Networks*, vol. 172, p.107148, 2020.

[3] N. Mohamed, J. Al-Jaroodi, I. Jawhar, A. Idries, and F. Mohammed, "Unmanned aerial vehicles applications in future smart cities," *Technological Forecasting and Social Change*, vol. 153, 2020.

[4] BBC NEWS. [Online]. Available: Iran 'building copy of captured US drone' RQ-170 Sentinel - BBC News.

[5] E. Elezi, G. Çankaya, A. Boyac, and S. Yarkan, "A detection and identification method based on signal power for different types of electronic jamming attacks on GPS signals," 2019 *IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications* (*PIMRC*), pp. 1–5, 2019.

[6] G. O'Mahony, K. McCarthy, P. Harris, and C. Murphy, "Developing novel low complexity models using received in-phase and quadrature-phase samples for interference detection and classification in wireless sensor network and GPS edge devices," *Ad Hoc Networks*, vol. 120, p.102562, 2021.

[7] B. Van den Bergh and S. Pollin, "Keeping UAVs under control during GPS jamming," *IEEE Systems Journal*, vol. 13, no. 2, pp. 2010–2021, 2018.

[8] A. Osman, M. Moussa, M. Tamazin, M. Korenberg, and A. Noureldin, "DOA elevation and azimuth angles estimation of GPS jamming signals using fast orthogonal search," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no.5, pp.3812–3821, 2020.

[9] Y. Chien, "Design of GPS anti-jamming systems using adaptive notch filters," *IEEE Systems Journal*, vol. 9, no. 2, pp. 451–460, 2013.

[10] W. Aldosari, M. Moinuddin, A. Aljohani, and U. Al-Saggaf, "Distributed extended kalman filtering based techniques for 3-d uav jamming localization," *Sensors*, vol. 20, no. 22, pp. 6405, 2020.

[11] Liu, H. Liu, W. Xu, and Y. Chen, "Error minimizing jammer localization through smart estimation of ambient noise," 2012 *IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems* (*MASS 2012*), pp. 308–316, 2012.

[12] T. Zhang, X. Ji, Z. Zhuang, and W. Xu, "JamCatcher: A mobile jammer localization scheme for advanced metering infrastructure in smart grid," *Sensors*, vol. 19, no.4, pp. 909, 2019.

[13] D. Borio, C. Gioia, A. Štern, F. Dimc and G. Baldini, "Jammer localization: From crowdsourcing to synthetic detection," *The 29th International Technical Meeting of The Satellite Division of the Institute of Navigation* (ION GNSS+ 2016), pp. 3107–3116, 2016.

[14] H. Liu, X. Wenyuan, Y. Chen, and Z. Liu," Localizing jammers in wireless networks," *IEEE International Conference on Pervasive Computing and Communications,* pp. 1–6, 2009.

[15] Y. Li, J. Pawlak, J. Price, K. Al Shamaileh, Q. Niyaz, S. Paheding, and V. Devabhaktuni, "Jamming detection and classification in OFDM-based UAVs via feature- and spectrogram-tailored machine learning," *IEEE Access*, vol. 10, pp. 16859–16870, 2022.

[16] Dataset. [Online]. Available: https://shorturl.at/nwGK8.