2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC), 18-22 September 2022, Portsmouth, Virginia, USA
DOI:10.1109/DASC55683.2022.9925778

# Radar Discrimination of Small Airborne Targets Through Kinematic Features and Machine Learning

Timothée Doumard, Fabio Gañán Riesco, Ivan Petrunin, Dimitrios Panagiotakopoulos
*School of Aerospace, Transport and Manufacturing*
*Cranfield University*
Bedford, England
{timothee.doumard, fabio.ganan}@gmail.com
{i.petrunin, d.panagiotakopoulos}@cranfield.ac.uk

Cameron Bennett, Stephen Harman
*Aveillant*
Impington, England
{cameron.bennett,stephen.harman}@aveillant.com

***Abstract*— This work studies binary classification problem for small airborne targets (drones vs other) by means of their trajectory analysis. For this purpose a set of the kinematic features extracted from drone trajectories using radar detections with a classification scheme that utilises Random Forests is proposed. The development is based on experimental data acquired from the Holographic radar from Aveillant Ltd. An approach for real-time classification is proposed, where an adaptive sliding window procedure is employed to make predictions over time from trajectories. Several models utilising different kinematic features (angle, slope, velocity, and their combination) are studied. The best model achieves an accuracy of more than 95%. In addition, fundamental issues with imbalanced datasets in the context of this topic are raised and illustrated using the collected data.**

***Keywords—Drone, classification, trajectory, motion, machine learning***

## I. INTRODUCTION

In recent years, the number of unmanned aerial vehicles (UAVs or drones) available commercially has increased dramatically due to the constant development of new technologies and declining costs of these devices. Drones offer numerous possibilities: from the inclusion of cameras for filming sports, building works, maintenance, assisting state security forces such as police or fire and rescue teams, to helping in the agriculture sector by planting seeds or analysing the soil nutrients with camera sensors. While the increased availability of UAVs brings many positive impacts to society, it also increases access for malicious applications. The number of crimes and illegal acts involving the use of drones is growing and may become a concern. The most common problem is the intrusion of drones into protected airspaces and restricted areas around key facilities. These breaches can cause irreparable damage to infrastructure, interfere with Air Traffic Control (ATC) and in extreme cases, can result in fatal consequences for civilians. In the case of airports, the use of drones in airspace around is highly restricted, as they can cause terrible damage if they are sucked into the aircraft engines during taking off or landing. In addition, there are other possible cases such as espionage, carrying weapons and other malicious activities. Therefore, early and reliable detection and classification of drones is found to be an important task in preventing crimes, as well as ensuring safety and security of people, important assets and critical infrastructure.

The aim of this work is inspired by this need and is related to the development of a binary classifier for drone recognition. The proposed system receives trajectories and outputs a binary prediction: “drone” or “other” (i.e. not a drone). In this work, an emphasis is put on using exclusively trajectory-based features so that the classification can operate reliably in conditions without micro-Doppler signatures. In addition, it is important that the classification approach is explainable for customers in safety-critical scenarios.

## II. RELATED WORKS

The problem of the drone classification from a sensory data has attracted significant attention in the literature. Various approaches have been explored, where signatures related to the type of drone could be extracted from the micro-Doppler features in radar data ( [1] [2] [3] [4] [5] [6]), acoustic signatures and image-based features ( [7] [8] [9]). While being useful and effective (especially for micro-Doppler based classification) there is an additional source of the information that can be providing more drone classification information – kinematic features. Several studies have explored such features in combination with AI techniques. Work [10] utilised Artificial Neural Network (ANN) with 30 neurons for drone classification using trajectory-related features. A total of 15 features and 5 more radar cross section (RCS) related features were explored. The classification performance was above 99% with all features, but somewhat reduced to 98.7% when 14 features were down selected.

Study [11] proposed two-stage classification between non drone and drone (divided in turn into rotary and fixed wing) classes. The method from [10] is used, studying the effect of the number of plots per segment. 20 kinematic-based and 30 signature-based features are used for each segment. The tracks were generated using specific real data. In the paper a Support Vector Machine (SVM) with radial basis kernel is trained. An accuracy of 95% was achieved, but the importance of each feature wasn’t determined. So, the performance of trajectory features remains unknown.

In work [12] the authors trained a K-Nearest Neighbour (KNN) classifier to differentiate between drones and aircrafts. They used x, y coordinates extracted from videos of installations at an airport. In this KNN, k=5. 60-point trajectories were selected based on the ratio between length and accuracy, which has reached 90%.

A drone vs bird classifier was developed in [13] using multiple features extracted from the trajectory information acquired from videos. Four trajectory features were extracted and used for training: rotation angle, curvature, velocity, and acceleration. The average value of each feature was used for each track. 1192 tracks were used (804 bird and 388 drone). The authors applied Principal Component Analysis (PCA) to reduce from 4 to 2-dimensional feature space. The final classifier was a SVM with a non-linear Gaussian kernel. A rather sparse dataset was used. This and the low number of features may be the reason for poor accuracy, compared to others found in the literature. According to the authors, the accuracy was just above 80%.

This paper [14] continue the work [13] with the modified dataset, where 110 kite tracks were added. An average of 85% accuracy was obtained this time.

In work [15] the authors analysed use of recurrent neural networks (RNN) for drone identification based on trajectory information. 5000 simulated flight tracks of birds and drones were generated for this purpose using different models. Each trajectory had a size of 60 points. The x, y, z components of the instantaneous velocity and acceleration, and the instantaneous turn rate were extracted. Three feature combinations were considered: using raw position as input (3 features), using velocity, acceleration and turn rate (7 features), and 3 Convolutional Neural Networks (CNN) layers before the RNN. The accuracy was, respectively: 66.3%, 80.4% and 99.3%. These last results must be taken with caution as the data were simulated and not extracted from real data.

The authors of work [16] developed a neural network (NN)-based system for detecting and classifying drones and birds by using trajectory information. The trajectories are divided into segments of n=16 points, and 15 kinematic features and 4 shape features are extracted from the real time video. The authors claim that the NN can extract more features from the data directly, and it is showed by the high performance of the model. The accuracy of the classification was over 99.5%. It was found however, that 4 shape features used may interfere with the trajectory classification.

The authors of the study [17] developed a real-time multi-class classifier based on trajectories and geographical information obtained from radar data. The features used included area, path, radar intensity, speed, normal acceleration, and temporal dynamics. Two classification algorithms (RGMM and DeltaGMM) were utilised, which are based on Gaussian mixture models and Naive Bayesian approach. A large dataset of 107722 tracks for all classes was used for training with variable class balance. The accuracy of RGMM and DeltaGMM was, respectively, 78% and 86%.

In the work [18], the authors employed a Random Forest (RF) technique coupled with an alpha beta filter to classify stationary and moving targets of 7 classes captured by radar by using their trajectory information. Simulated data is generated to test the model. Track data is extracted from Automatic Identification System (AIS), Automatic Dependent Surveillance-Broadcast (ADS-B), GPS logs and real-world radar data. The same features as in the previous work are extracted from tracks and fed into the model. The overall performance with real world data is 79.7%.

In work [19] the authors utilised a Gradient Boosting classifier that classified 7 classes of ships based on trajectory data. 7 kinematic features were extracted from an AIS receiver from which statistical indicators were obtained. The main hypothesis was that every ship had its own characteristics and those would be reflected in the statistical features, like maximum or mean. Authors claim that local variations of speed are captured in the acceleration. The approach demonstrated an average F1-score of 86%.

In study [20], the authors proposed an approach to classification of cargo ships and fishing boats using trajectory data. Trajectories are split into 3 categories: anchored-off, straight-sailing and turning. Then, 3 sets of features are extracted from segments: global features, straight-sailing features and turning features. Feature selection is performed observing the probability distribution functions. Logistic regression has been chosen for classification purpose due to its hight accuracy and good interpretability. The Area Under the Curve (AUC) value of the final model was 0.963.

The authors of work [21] proposed to use RNN-based classifier with Gated Recurrent Units (GRU) for vessel classification based on the trajectory data extracted from coordinates, timestamp, and distance to shore. The feature vector included velocity, tangential and normal acceleration, displacement angle and distance to shore. The model obtained an average accuracy of 78.3%. The authors also trained a RF which has achieved 76.4%. Although the RNN model was 1.9% higher than RF, this does not indicate that the use of RNN is the most appropriate for this type of classification.

## III. Feature Analysis and Extraction

### A. Dataset

The data used for this work was collected by Aveillant's Gamekeeper 16U radars (Fig 1), which are deployed worldwide at many different sites. The Gamekeeper radar is a L-band Holographic radar specifically designed for detecting, tracking, and classifying drones to ranges of 7.5km, its unique design facilitates hi-fidelity measurement of micro-Doppler signals and tracking of all targets within its surveillance volume at high track update rates (approximately 0.28s).

The dataset consists of 76 subsets containing the trajectories of flying objects captured by the radar. Each datapoint consists of 56 features among which are: the x, y, z


Fig 1. Aveillant Gamekeeper 16U holographic radar

positions of the object at a given instant, the time at which this object and position have been captured, instantaneous velocities and their decompositions in the x, y, z axes, instantaneous accelerations, instantaneous jolt, among others. The datapoints are acquired with a constant sample time of 0.279 seconds.

The complete dataset consists of 117294 trajectories, of which 392 belong to drones and 116902 to other flying objects. The percentage of trajectories corresponding to the positive class (drones) represents approximately 0.3% of the total trajectories. This makes the dataset very imbalanced and makes it necessary to use techniques such as under sampling or oversampling in the training of the classification models.

Due to the versatile nature of the targets of interest (mostly birds), the track lengths of the trajectories can be quite short. 32% of the trajectories contain 10 or less datapoints and only 2.4% contain more than 200 datapoints. As an initial hypothesis, all trajectories with less than 30 datapoints in length were removed as they do not contain enough data to extract reliable statistical information for classification. After pruning the tracks, the dataset used in this work consists of the following: 38069 trajectories, ranging between 30 datapoints and more than 2000. 274 trajectories are from "drone" and 37795 are from the negative class, “other”.

## B. Data Preview

Trajectories extracted from the dataset are defined as follows:

$$T_k = \{(x_0, y_0, z_0), \ldots, (x_n, y_n, z_n)\} \quad (1)$$

where $k$ indicates the trajectory number and $x_n$, $y_n$ and $z_n$ represent the coordinates in 3-D space at the instant $n$ (or datapoint) within the trajectory.

In the dataset used in this work, of the 274 trajectories associated with drones, a large part of the long trajectories (>1000 datapoints) draws on the x, y plane specific patterns such as flowers, and “Pac-mans”, among others. The unnatural pattern of these waypoints is noted as a potential source of overfitting and so this must be considered when designing trajectory features.

In this work it has been hypothesised also that information on how objects fly can be gathered through statistical values of speed, yaw angle and slope without the need to characterise specific shape of flight patterns. This work focuses more on how objects fly (statistically) rather than on what trajectory patterns are performed in the flight.

## C. Features

Flight trajectories are characterised by an arbitrary number of points ordered in time in the three-dimensional space. From these points and their time, 3 properties that characterise the shape of the trajectory have been chosen: velocity, angle of rotation and slope angle. The angle of rotation characterises the trajectory in the horizontal plane, the slope characterises the changes in the Z-axis and the velocity characterises the size or distances between the sample points of the trajectory (assuming a homogeneous sample rate along the dataset).

### 1) Velocity

Speed is one of the main characteristics of a trajectory. The physical characteristics of drones and other flying objects such as birds are different. Therefore, the velocity of these objects can reflect differences between them.

Given the x, y, z components of an object's positions in space over time, the approximate instantaneous velocity module is calculated as the Euclidean distance of two consecutive points divided by the time travelled between the two points.

$$v_t = \left(\frac{(\Delta x_t, \Delta y_t, \Delta z_t)}{\Delta t}\right) \quad (2)$$

where $t$ indicates the instant or single datapoint in time; $\Delta x_t$, $\Delta y_t$ and $\Delta z_t$ indicate the difference between the x, y, and z components of two consecutive datapoints, and $\Delta t$ is the sampling time (0.279 seconds).

By going through the trajectory, an array of instantaneous velocities is obtained along the trajectory, given by:

$$v_i = \{v_1, \ldots, v_{n-1}, v_n\} \quad (3)$$

where $i$ indicates the index of a trajectory, and $n$ is the total number of datapoints within that trajectory.

The hypothesis is that the mean and variations in this array can provide differentiating information for the classification of drones and other targets, as the behaviour patterns of the objects are different. Once the velocity array is obtained, the mean, standard deviation, maximum and minimum values are calculated. The distribution of the velocity information is shown in Fig 2.

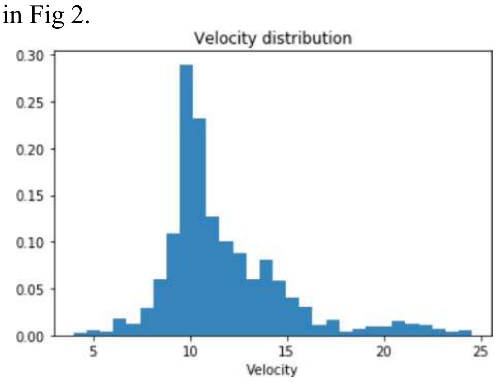

Fig 2. Example of the velocity distribution.

### 2) Rotation Angle

The angle of rotation is one of the main properties characterising the trajectory of a target. Given 3 points of a trajectory in the two-dimensional space formed by the coordinates X, Y, the angle of rotation is defined as the angle formed by the two vectors joining the first point with the second and the second with the third, respectively. Both negative and positive turning angles could be obtained, i.e. left and right turns, but the absolute value is used to not make a distinction. The angle of rotation lies between the limits of 0 and $\pi$. The rotation angle is defined by the formula:

$$\theta_t = \arccos\left(\frac{(\Delta x_t, \Delta y_t) \cdot (\Delta x_{t+1}, \Delta y_{t+1})}{\|(\Delta x_t, \Delta y_t)\| \cdot \|(\Delta x_{t+1}, \Delta y_{t+1})\|}\right) \quad (4)$$

where $t$ indicates the instant or single datapoint in time and $\Delta x_t$ and $\Delta y_t$ indicate the difference between the $x$ and $y$ components of two consecutive datapoints.

The array of rotation angles of a trajectory is given by:

$$\theta_i = \{\theta_1, \ldots, \theta_{n-1}, \theta_n\} \quad (5)$$

where $i$ indicates the number of an entire trajectory, and $n$ is the datapoint within that trajectory.

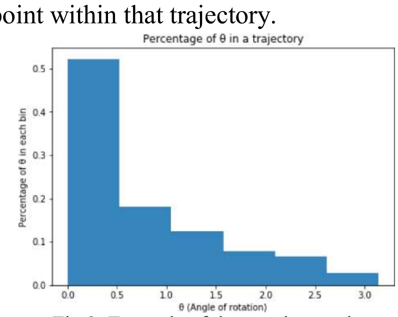

Fig 3. Example of the rotation angle

Once the array of rotation angles has been obtained, a histogram of 6 bins is made and displayed in Fig 3. The histogram is then used to encapsulate the distribution of rotation angles within the trajectory. The distribution in the histogram reflects the behaviour of the flying object's motion.

### 3) Slope Angle

The third feature is based on the height fluctuation of the target. The slope angle is defined as the angle formed by the vector joining two points with respect to the horizontal plane (ground). The values of the slope angle are in the range $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$. The slope angle is defined by the formula:

$$\varphi_t = arctan\left(\frac{\Delta z_t}{\|(\Delta x_t, \Delta y_t)\|}\right) \quad (6)$$

where $t$ indicates the instant or single datapoint in time and $\Delta x_t$, $\Delta y_t$ and $\Delta z_t$ indicate the difference between the $x$, $y$ and $z$ components of two consecutive datapoints. The array obtained is given by:

$$\varphi_i = \{\varphi_1, \dots, \varphi_{n-1}, \varphi_n\} \quad (7)$$

where $i$ indicates the trajectory index, and $n$ is the datapoint within that trajectory. Once the array of slope values has been obtained, 6 bins a histogram is made and showed in Fig 4, following the same steps as in the rotation angle.

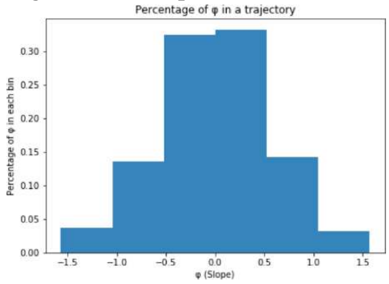


Fig 4. Example of slope angle distribution

## IV. PROPOSED METHODS FOR CLASSIFICATION

Two trajectory classification methods are proposed in this paper:

1. Full trajectory classification. The features of the full trajectory will be extracted, and the full trajectory classification will be performed.

2. Sliding window classification. A sliding window will extract features every few steps and make predictions using small windows of the track. A final classification of the complete trajectory is made from the window predictions made along the trajectory.

This section will describe the methodology behind designing each of these classification algorithms and the preparation required for training.

### A. Imbalance Parameter

One of the main issues of this work is the fact that the dataset is very imbalanced. In fact, in a real case where the aim is to detect and classify drone trajectories, the data obtained would be even more unbalanced. The number of drones that a radar can detect is minimal compared to the number of birds on the same period. The effect of this imbalance on both training sets and testing sets has been analysed.

For this purpose, an imbalance parameter has been defined as the ratio between the number of trajectories of the negative class ("other") and the number of trajectories of the positive class ("drone"). The higher the imbalance parameter, the higher the proportion of "other" trajectories. This parameter is used to indicate the relationship of the classes in the different training and testing sets that are created for the evaluation of the models and their performance.

### B. Trajectories Segmentation

Due to the low number of "drone" trajectories, in this work the trajectories are split to generate sets containing more trajectories. If each "drone" trajectory is split in two, twice as many samples of that class are obtained. This allows for more data, more variety and possibly better training and final performance of the models.

The segmentation in the sliding window classification section is carried out in the following way:

1. Sliding window size. A suitable segment size is chosen for the classification based on experimental results.

2. Segmentation step. Each trajectory is divided into segments of the chosen size. Segments of the chosen size are extracted for each step, and there may be overlapping datapoints in the segments. For example: fora window size of 50 and a segmentation step of 1, each new segment will share 49 datapoints with the previous and next segment.

3. Dataset update. A new dataset is created from the segments extracted from the original paths. The new dataset contains trajectories of a single size, making the statistical features.

### C. Classification Method: Smart Sliding Window Approach

The following method of trajectory classification using a sliding window is proposed:

1. A segmentation step is chosen that allows its use in real time. In this case, a segmentation step of 1 is used to maintain a “real-time” classification update rate.

2. For each segmentation step a prediction of the window size datapoints is made and the predictions value is added to an array. This prediction is not a binary value but a probability. The RF technique of the Scikit-learn library allows a non-discrete prediction that ranges between 0 and 1. Some of the functions in this library (and other common libraries) use the “predict_proba” class to perform this type of prediction. Although its name indicates "probability", there are some studies that show that these probabilities are not well calibrated [33], [34]. This issue is out of the scope of this work, so the non-binary prediction will be used as the probability that a trajectory is of one class or another.

3. At the end of the trajectory, the average of the predictions is taken and, if it exceeds a threshold (empirically selected), it is classified as a "drone". If it does not exceed the threshold, it is classified as "other".

It should be noted that this method is only used in the Test section, to simulate what could be its use in real time.

### D. Models and Techniques

The classification technique adopted in this work is Random Forest. It belongs to a class of ensemble learning techniques and combines multiple (hundreds) of base classifiers (Decision Trees) to perform a regression or classification task by predicting an output with every Decision Tree (DT) and providing a final value of the prediction based on those previous predictions. The model selected is based on the fact that explainability is a requirement for this work. DTs are among the most explainable models, as their decisions can be visualised and understood by analysing them after training with data.

Other explainable models such as Logistic Regression, SVM and KNN were considered for this study, but the high dimensionality of the chosen features makes them less explainable. Also, logistic regression, while being a very explainable model, performs much worse than RF. SVMs use the hyperplane to divide the features into classes, so the explainability is much lower. RF, being an ensemble model, should outperform simpler explainable models.

PCA and t-Distributed Stochastic Neighbour Embedding (t-SNE) are the techniques used in this work for data visualization only. PCA is used for dimensionality reduction, and t-SNE is used for visualisation. The results obtained in this study have been based on the workstation with Intel® Core™ i7-6700HQ CPU running at 2.60GHz with 8GB RAM and Nvidia GTX960M GPU.

## V. Results

### A. *Full Trajectory Classification*

#### 1) *Data Visualisation*

##### a) *Velocity Features*

Fig 5 shows the velocity features distributions. The average velocity of the "drone" was found to be slightly lower than that of other flying objects and, in addition, the distribution is slightly more skewed. The standard deviation and the maximum speed appear similar, but both show in the negative class ("other") a heavy tail. This may be due to the lack of more drone trajectories and not to statistical issues.

In the minimum velocity the difference between the two classes is shown. This last feature can be very useful, as multicopter drones can hover without having to move in any direction, which is not the case for birds, which must flap or glide to stay in flight (although there are special cases where a bird can take advantage of a wind current to hover or elevate its position almost without moving in the X, Y directions).

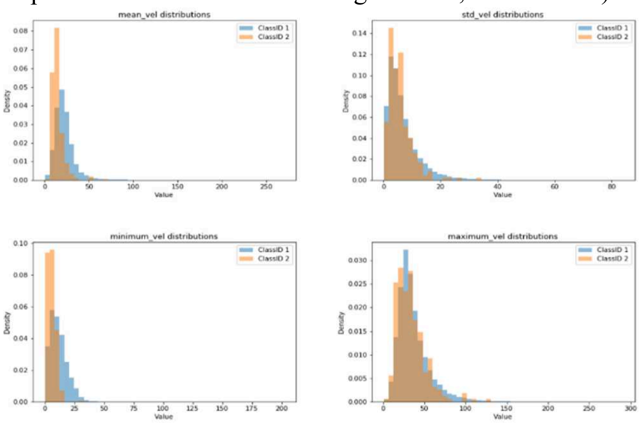

Fig 5. Distributions of velocity features.

##### b) *Rotation Angle Features*

From the data are extracted distributions to observe some behaviours of the aerial objects. The distributions are separated in 30° angles, for example, feature “angle_0_30_s0” represents the angles between 0° and 30°. And from these distributions the following observations are made.

The distribution of the feature "angle_0_30_s0" indicates that contrary to an initial hypothesis the trajectories for class "drone" targets are more curved than the flying objects of the other class, which are usually birds. Intuition would tell us that birds, being living beings with a greater capacity for reaction and much greater mobility and agility, would have to trace more curved trajectories, with all kinds of movements and pirouettes. However, the data confirms that this is not entirely true.

The distributions of the features "angle_30_60_s0", "angle_60_90_s0", "angle_90_120_s0", "angle_120_150_s0" and "angle_150_180_s0" confirm the above observation. The most repeated value of these distributions is "0", indicating that for class “other” trajectories the angular rotations are smaller than for another class. It is particularly curious to note that the distributions are similar, except for the peaks resulting from the "0" values of each distribution. The distribution of the feature "max_angle_s0" reconfirms the previous observation. The "drone" targets seem to make sharper turns than the “other” target class.

##### c) *Slope Angle Features*

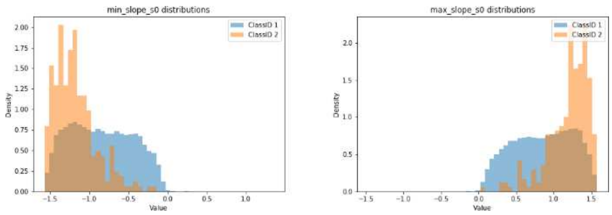

Fig 6. Distributions of max and min slope angle for classes

From the distributions "slope_0_30_s0", "slope_30_60_s0", "slope_120_150_s0" and "slope_150_180_s0", it can be obtained that most of the trajectories of "other" describe slope angles whose values are between "-30" and "30" with respect to the ground plane (it should be remembered that the feature names indicate the range of angles with respect to the vertical line Z, not with respect to the horizontal plane). The maximum and minimum slopes (min_”slope_s0” and max_”slope_s0”) show a big difference as shown in Fig 6. Drones seem to be having trajectories with steeper slopes, both positive and negative. The "other" class targets, on the other hand, show a distribution similar to a uniform one.

##### d) *t-SNE*

To check the separability presented by these selected features, t-SNE is used for visualization purposes. To considerably reduce the computational time of this statistical algorithm, PCA is applied on the extracted features dataset, reducing the number of features from 19 to 2. These 2 features captured more than 99% of the variance of the data.

The number of trajectories used from each class for these visualisations has been balanced. A random sample of 274 trajectories from the class "other" has been taken for this purpose. Even though 274 trajectories are not as representative, they can provide a general visualization of the separability of the classes. These 2-D representations allow us to check the separation between the classes. The legends of the graphs indicate the negative class ("other") with the value "0" and the positive class ("drone”) with the value "1".

In the graphs of velocity features and slope angle features a separation of the classes can be observed. There are clusters that are more defined in the positive class and others that contain more data from the negative class. Even so, it can be observed that there are many trajectories of both classes that are not differentiated from the other classes. This may indicate that there are indistinguishable trajectories or that the extracted features do not contain enough information to make this classification more accurate.

The t-SNE of angle features shows some separation of the classes, but it can be observed that most of the trajectories show similar values, making it more difficult (or impossible without overfitting) to classify them using angle features. Even then, a large cluster of the positive class can be seen in the upper left of this graph, with some outliers from the negative class.

Finally, Fig 7 using all features shows a separation of the classes. This separation is promising for the training of the model. However, as previously said, the class separation is not totally evident in any case, possibly due to the extracted features or due to the trajectories being indistinguishable between "drone" and "other".

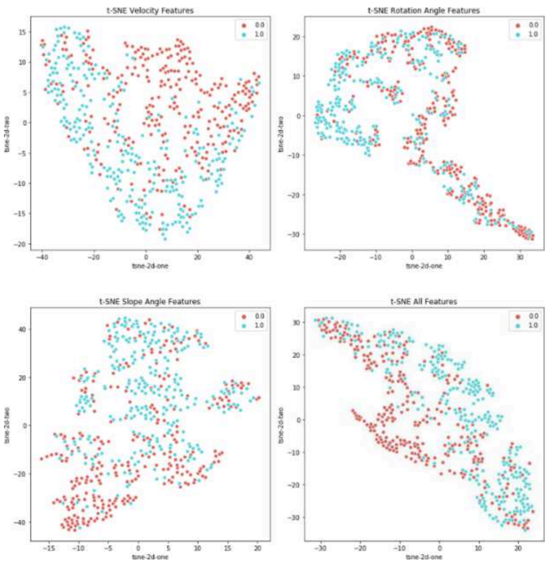


Fig 7. t-SNE of sets of features and all features.

### 2) *Features Importance*

The RF model of the Scikit-learn library allows us to know the importance of the features used for classification. For this experiment, the dataset is constructed to have a 50/50 ratio of the two classes. This is because keeping the original ratio (>0.7% of the positive class) to the models struggles to learn the differences between the classes and perform much worse. In the next section, the effect of the training imbalance on the results obtained will be studied. For this analysis, all features were used in the training of the first model. Once the importance of the features is analysed, the least important feature is removed, and the model trained again. These steps are then repeated until 1 feature remains.

After analysing the importance of each feature, it appears that the average, maximum and minimum speed are the most important features. In fact, the 3 features remain among the best until the end. The angle features are the first to be eliminated. This may indicate that these features provide little information for the classification of the trajectories. Then, slope features are the ones to be eliminated. The last 4 features

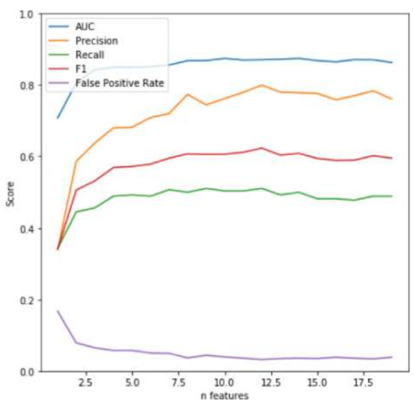


Fig 8. Performance vs number of features.

are “minimum_vel”, “max_slope_s0”, “mean_vel” and “min_slope_s0”. When analysed in Fig 8, the number of features doesn’t affect the performance of the model until half of the features are not used and decrease steeply when going under 7 features.

### 3) *Imbalance Parameter*

The class imbalance will significantly affect the classification capabilities of an RF, so it is important that the effect of this imbalance is analysed. To do so, the parameter "imbalance_parameter" is defined as the division of the number of trajectories of the positive class ("drone") and the number of trajectories of the negative class ("other"). This parameter is used to indicate the relationship of the classes in the different training sets.

In the next experiment, 7 training sets were generated as follows. For each “imbalance_parameter” (1, 2, 5, 10, 20, 50, 100):

The training set is initially empty.

Add all trajectories of the positive class of the dataset (215 trajectories) to the training set.

Add 215*imbalance_parameter trajectories (randomly chosen) of the negative class of the dataset to the training set.

Thus, the proportion of classes will be different for each training set. The proportions of each training set are as shown in Table I.

Table I. Imbalance parameter proportions

| Imbalance parameter | Proportion (%) |
|---|---|
| 1 | 50/50 |
| 2 | 33/67 |
| 5 | 17/83 |
| 10 | 9.1/90.9 |
| 20 | 4.8/94.2 |
| 50 | 1.96/98.04 |
| 100 | 1/99 |

Based on the imbalance parameter values seven different models have been trained using dedicated training sets and with cross-validation (with a validation/training ratio of 0.2). The following metrics are calculated and analysed: AUC, accuracy, recall, F1 score and False Positive Ratio (FPR). Fig 9 shows these metrics vs imbalance parameter for balancing precision and recall (left panel) and maintaining high precision (right panel).

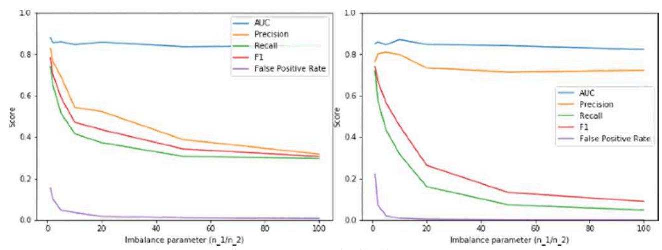


Fig 9. Performance vs imbalance parameter.

#### a) *Maintaining Precision and Recall Balanced*

The precision remains above 75% for all models. The model with “imbalance_parameter” =1 obtains a precision and recall of ~0.8, approximately. In the following models, with “imbalance_parameter” equal to 2, 5, 10, 20 and 50, recall and F1 drop drastically, falling below 40% in the last two models. The AUC remains above 80% in all models. And the FPR goes from 15.3% to 1.6%.

#### b) Maintaining High Precision

Model precision remains above 75% for all models. The recall and F1 drop drastically, lower than in the previous case, and reach below 20% in the models trained with less balanced datasets. The model with “imbalance_parameter” =1 obtains an accuracy and recall of ~75%, approximately. In the following models, with “imbalance_parameter” equal to 2, 5, 10, 20 and 50, recall and F1 drop drastically, falling below 20% in the last two models. The AUC remains above 80% in all models. And the FPR drops from 21.1% to 0.7%.

### 4) Feature Impact Analysis

Next, the study of the performance changes due to the selected features has been performed. For this purpose 4 RF-based models were created with different features: velocity features, angular features, slope features and the last model used all features. A hyperparameter optimisation of these models was carried out using a Grid Search. The scoring criteria used is the F1 score. The reason for choosing this scoring is to have a balance between precision and recall.

The hyperparameters evaluated were the following:

- 'class_weight': ['balanced'],
- 'bootstrap': [True, False],
- 'n_estimators': [100, 300, 500],
- 'max_features': [1,2,3,4],
- 'criterion': ['gini', 'entropy']

The rest of models used in this work where also optimized, but it was redundant to include the optimization in every section. The optimized models used in this work can be found in the Appendices section.

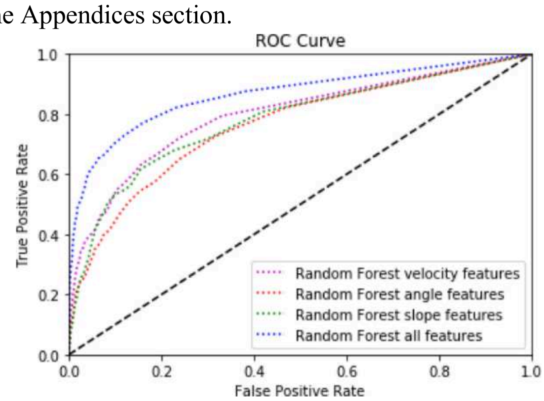


Fig 10. ROC curve of 4 models.

The Receiving Operating Characteristic (ROC) curve for the 4 models, cross-validated on training set is shown in Fig 10. The AUC values for these models are shown in Table II:

Table II. ROC curves AUC values

| | |
|---|---|
| Angle features: 0.7632 | Slope features: 0.7798 |
| Velocity features: 0.7977 | All features: 0.8666 |

The model which uses all features appears to outperform the other models significantly. The model with the worst performance is the one using only angular features, followed by the one using slope features. This may indicate that the features used in the last two models lack important information to classify with high accuracy classes “drone” and “other”. As this is an ensemble model based on the 3 that uses specific set of features would create a redundant model. An ensemble model of the 3 models was trained and the ROC curve and AUC value was exactly the same as the simple ensemble model that used all the features.

### 5) Test

Next, the final test of the effect of the imbalance parameter over the training set will be analysed. The first analysed result is the decrease in false positives. In the trained model with a ratio of 50/50 % the number of false positives is very high (~860). In contrast, in the trained model with a ratio of 1/99%, the number of false positives is 49 (0.69%), a low value.

Even though the accuracy and recall are not optimal, the accuracy of the final model (with a more realistic proportion) is 98.82%. The metrics used for the evaluation of models using imbalanced data should be analysed with care. In most papers, the data are either balanced and the precision and recall are high (~80% or more) or the metric used for evaluation is accuracy, which can be misleading, like in this case. With the low number of target trajectories, these models are very likely to have suffered from overfitting.

## B. Sliding Window Classification

In a real use case scenario, in addition to receiving trajectories from birds and other flying objects hundreds of thousands of times before a drone appears, there is the issue of detection and classification time. As described above, the sample time of the trajectory data is 0.279 seconds. If the detection needs to happen in real time, an optimal window size must be chosen to perform this type of classification.

### 1) Best Window Size

In this section the effect of segment size on model performance is analysed. 34294 trajectories are selected, 238 for class "drone" and 34056 for class "other" (randomly chosen), and segmented each trajectory is to generate a new training set.

The segmentation was performed for 10 different window sizes: 20, 25,…,60, 65. For each window size, the total number of segments of the new dataset increases as expected, as the original trajectories can be divided into more segments when the window size is reduced. 10 different training sets were generated, one for each window size. The total number of segments ranged between 2524 and 4163 for "drone" trajectories, and 24368 and 50734 for class "other" trajectories. The new training sets were much richer in quantity and in information, as the following results will show.

Eleven identical RF-based models were trained using different training sets. The confusion matrices 10-fold cross-validation have been generated and evaluated. They show high accuracy (>94%). The precision, recall, F1 and F-beta scores for each window size are shown in Fig 11. The scores increase from approximately 50% to around 70% for window sizes of 45 to 65. The effect on performance with larger window sizes has not been tested, as the number of trajectories with 70 or more datapoints is low. Furthermore, with a larger window size, the time to the first classification of a trajectory would increase, also increasing the risk of not detecting a target in a real scenario. Although a window size of 20 or 30

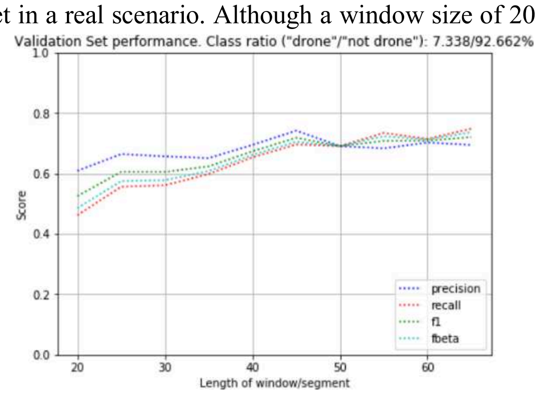


Fig 11. Performance vs window size

could be chosen, a window size of 50 is selected for the rest of the section, which translates into 13.95 seconds. In the following section, new training and testing sets are generated to analyse the effect of class unbalancing.

*2) Validation*

*a) Train Balanced, Test Balanced*

A model was trained with 31442 "drone" segments and 36269 "other" segments, an approximately balanced ratio. Through 5-fold cross-validation, the ROC curve and AUC score of 0.9674 are obtained.

The large increase in performance over models using the full trajectory for classification can be observed. By increasing the number of trajectories (new segments), the model can learn more accurately the characteristics of "drone" and "other" flights. The hypothesis that by dividing trajectories into segments, local information can be extracted more accurately from the trajectories is made, i.e. eventual behaviours unique to each class. Both the increase of samples and the extraction of local information significantly improve the performance of the model.

The model was evaluated with a validation set of 6248 "drone" segments and 7283 "other" segments of trajectories never seen before by the RF. The confusion matrix is shown in Fig 12 and the scores obtained in Table III.

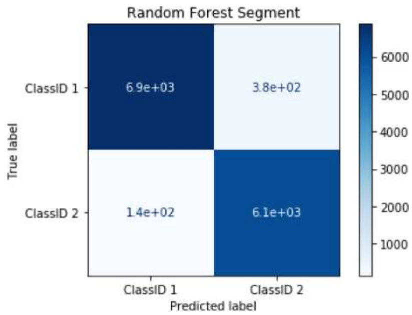


Fig 12. Confusion matrix on validation set.

Table III. Scores on validation set

| | |
|---|---|
| Precision: 0.9409 | Recall: 0.9767 |
| F1: 0.9585 | Fbeta: 0.9693 |
| Accuracy: 0.9589 | |

Here the FPR is 0.05. Comparing this model with the previous model in Section V. A. 3), where the training and validation set were balanced, the performance has increased by approximately 15% (from ~80% to ~95%).

These results, although very promising, will be challenged in the next section, where the same experiment will be performed with unbalanced training and validation sets.

*b) Train Imbalanced, Test Imbalanced*

A model was trained with 1353 "drone" segments and 36269 "other" segments, a ratio closer to that of a real case, where the number of drone trajectories is much lower than that of other flying objects such as birds. The ROC curve and the AUC of 0.8868 were obtained using 5-fold cross-validation and are shown in Fig 13.

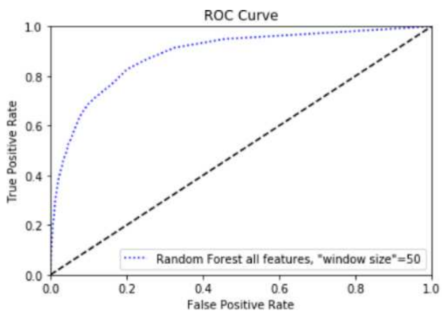


Fig 13. ROC curve of model using imbalanced training set

These results show a significant drop in model performance. One of the reasons may be the considerable reduction of data for training (30089 less "drone" trajectories). This reduction of data may cause the model to underfit the data and that could be the reason for this drop in performance. The model was evaluated with a validation set of 271 "drone" segments and 7283 "other" segments of trajectories never seen before by the RF. See the confusion matrix in Fig 14 and the scores obtained in Table IV.

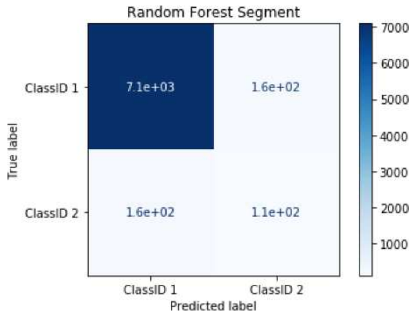


Fig 14. Confusion matrix on validation set.

Table IV. Scores on validation set.

| | |
|---|---|
| Precision: 0.4053 | Recall: 0.3948 |
| F1: 0.3999 | Fbeta: 0.3969 |
| Accuracy: 0.9573 | |

Here the FPR is 0.022. Still, comparing this model with the previous model in Section V. A. 3), where the training and validation set were imbalanced with a similar proportion, the performance has increased by about 10% (from ~30% to ~40%). These results show the importance of the choice of metrics used for the evaluation of these models on balanced and imbalanced datasets.

*3) Test*

In this section, a method for semi real-time trajectory classification is depicted. For this, a final test set is generated, using 30 complete "drone" trajectories and 2237 complete "other" trajectories. The test is imbalanced to represent the typical diversity in radar trials. The models used are the two trained in the previous section: one with a balanced training set and the second with an imbalanced training set. For the final test the method is described in Section IV. C has been used, where each trajectory is segmented to simulate a semi real-time prediction. That is, for each trajectory, the new datapoint and the 49 previous datapoints are collected to generate a segment of 50. The model then predicts the class of each segment and performs a final classification of the trajectory.

*a) Train Imbalanced, Test Imbalanced*

Fig 15 shows the "real-time" predictions of the trajectories of the test set of the model trained with an imbalanced training set. The red dashed lines represent the predictions of the "drone" trajectories, and the blue dashed lines represent the predictions of the "other" target trajectories. The x-axis represents the number of predicted segments over time. The vertical axis represents segment predictions, with "0" being the negative class and "1" the positive class.

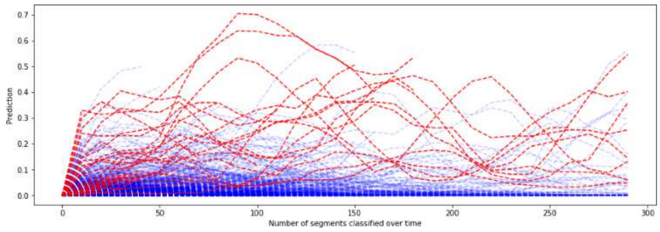


Fig 15. Predictions of classes over time on the test set.

For better visualisation, the predictions have been cut from 300 datapoints, as the lengths of the trajectories are very different and make their visualisation more difficult.

With the classification method used, it appears that trajectories corresponding to the "drone" class have higher prediction scores than those corresponding to the class "other". Towards the end, the trajectory tends to be correctly classified, even if a few false positive/negative appears. Fig 16 shows the confusion matrix of the final classification of each complete trajectory, scores are shown in table Table V.

Table V - Scores of final classification on test set.

| | |
|---|---|
| Accuracy: 0.9816 | Precision: 0.3777 |
| Recall: 0.5666 | F1: 0.4533 |

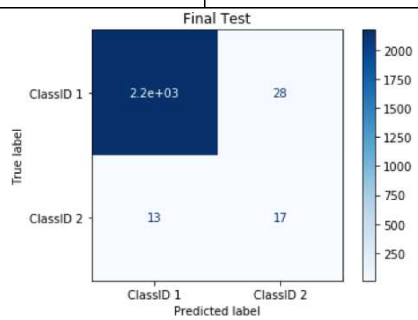


Fig 16. Confusion matrix of final classification on test set.

The accuracy in this case is above 98%. There were only 28 false positives out of 2237 trajectories classified as "other". Accuracy, recall and F1 score are low, but the model managed to correctly classify 17 of the 30 "drone" trajectories even with such a low FPR.

The thresholds of the RF and the final trajectory classification algorithm could be adjusted to achieve the required performance. Perhaps a much higher FPR could be assumed, as long as all or almost all "drone" trajectories are classified correctly. This question is outside the scope of this paper, as it is a decision to be made by experts using these classification systems for real cases.

*b) Train Balanced, Test Imbalanced*

Fig 17 represent the "real-time" predictions of the trajectories of the test set of the model trained with a training set balanced.

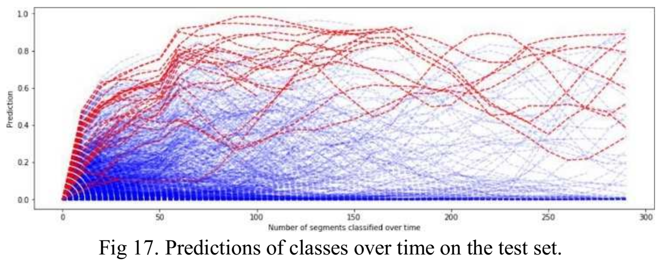


Fig 17. Predictions of classes over time on the test set.

A lot of noise can be observed in the initial predictions of "other" trajectories, however, majority of the predictions for this class stabilises around "0". As a general observation and compared to the previous case, the predictions are more confident. That is, the trajectories of class "drone" receive a prediction score closer to "1". In the previous case, very few trajectories exceeded the 0.5 threshold. Fig 18 shows the confusion matrix of the final ranking of each full trajectory whereas Table VI shows the scores.

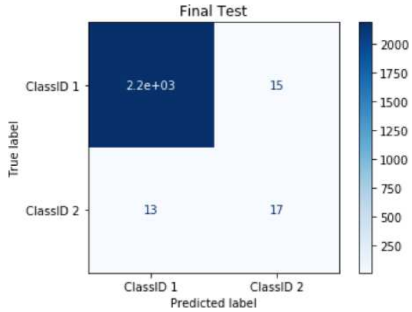


Fig 18. Confusion matrix of final classification on test set.

Table VI. Scores of final classification on the test set.

| | |
|---|---|
| Accuracy: 0.9874 | Precision: 0.5313 |
| Recall: 0.5666 | F1: 0.5484 |

The results are similar to the previous case, but the precision has increased by almost 20% and the F1 score by 10%. The accuracy is still high. Training with a balanced dataset, in addition to including more data (segments), increases the accuracy of the final classifier.

It is interesting to note that the number of correctly classified drone trajectories is exactly the same as in the previous case (17 correct classifications). In fact, a further check was performed to determine whether they were the same trajectories in the two cases. It was found that the trajectories of correctly and incorrectly classified "drones" were the same. This may indicate that there are trajectories that cannot be distinguished or that the proposed method is not efficient in distinguishing them.

The low amount of drone trajectories used in these tests may be the reason of the low results as well. 30 trajectories could be not enough for evaluation. In the context of machine learning, big numbers are the key, for training and for testing. More trajectories could be used to make this test more representative. In any case, although the metrics used show low values, the results are promising and show that there is still room for improvement.

## C. General Discussion of Results

Due to the random nature of the experiments performed (such as the choice of non-drone trajectories) and the randomness of the RFs training algorithm, the results could vary with each new result or new training. Each result of this work has been repeated dozens of times with numerous small changes in the way the dataset splits, trajectory choices, metrics used, models used, and features chosen, among others. The results shown in this study are the most significant ones found during the development work, and represent issues that are not covered in other published works known to the authors.

Among these issues is the effect of the imbalanced training and testing dataset. From the results of the sliding window classification section, it can be observed that training with a balanced training set gives a better performance, in particular, a higher precision. This may be due, as mentioned above, to the increased number of samples in the training set, rather than to the unbalanced training set itself. These types of issues should be studied in future work on similar topics to this one.

Work dealing with the development of classification systems for drones and other flying objects hardly considers the real context of this topic. Encountering a drone flying in the sky is much less common than encountering birds flying. In a real application of these systems, a 50/50 drones-birds ratio isn't possible, a ratio of 1 to 10000 or more orders of magnitude is more realistic.

Although the results in the sliding window test are not the best, only one method has been proposed for the final classification. A new method could increase precision and recall. Even though the model performs worse in the sliding window classification, the high performance in the validation test only indicates that the classification method for real time classification is the limiting factor, the one that makes the final classification perform worse. Also, as said earlier, 30 trajectories (drones) could be a very low number for testing. With 1 or 2 more orders of magnitude of drone trajectories the models could be better evaluated and give confident metrics.

In short, and as a final comment, the quantity of data in the context of machine learning is as important as its quality. By working with so few drone trajectories, the results obtained do not indicate that this system should be used in real time without risk; the number of drones that are not classified correctly is still high and could lead to very dangerous situations for people or facilities. A near-perfect recall should be the main goal, i.e. to be able to classify almost all drones correctly. But, of course, this should be coupled with a very low FPR, as it is not desirable to have so many false alarms, which can be counterproductive for the operators who ultimately control this type of drone detection and classification system. The creation of new trajectories with different types of drones will undoubtedly help the creation of models with very high recall and very low FPR.

Furthermore, the average speed of the targets of class “drone" in the dataset surely indicates that the drones used for the creation of the dataset are the rotary ones. Fixed-wing drones can reach much higher speeds. Furthermore, the trajectories and flight patterns that fixed-wing drones can generate are likely to be different, adding more data variety and complexity to this type of work.

When comparing our work with what is present in the literature, the results seem to match the performance of the most relevant literature in terms of accuracy or precision when given. The only model that clearly outperforms the ones in this work is [16]. Their model was a NN with 3 hidden layers of 64 neurons each, which is less explainable and 4 of the features input of the model were not only trajectory ones, but also included visual/shape features (number of pixels of the detected object, mean, variance and standard deviation).

## VI. Conclusion

A full trajectory classifier with 95% accuracy, precision and recall has been developed. This classifier has been produced in response to a way to differentiate drone and other trajectories from radar trajectory data only. This classifier can serve as the basis for real-time use supported by the optimized window size parameter.

## VII. References


[1] C. Bennett, M. Jahangir, F. Fioranelli, B. I. Ahmad et J. Le Kernec, «Use of Symmetrical Peak Extraction in Drone Micro-Doppler Classification for Staring Radar,» 2020.

[2] M. Jahangir, C. Baker et G. Oswald, «Doppler characteristics of micro-drones with L-band multibeam staring radar,» chez *IEEE RadarCon 2017*, Seattle, 2017.

[3] M. Jahangir, B. Ahmad et C. Baker, «Robust Drone Classification Using Two-Stage Decision Trees and Results from SESAR SAFIR Trials,» chez *2020 IEEE International Radar Conference*, Washington DC, 2020.

[4] S. Rahman et D. . A. Robertson, «Radar micro-Doppler signatures of drones and birds at K-band and W-band,» *Scientific Reports,* 2018.

[5] S. Rahman et D. A. Robertson, «Classification of drones and birds using convolutional neural networks applied to radar micro-Doppler spectrogram images,» *IET Radar, Sonar & Navigation,* 2020.

[6] A. Brewster et A. Balleri, «Extraction and analysis of micro-Doppler signatures by the Empirical Mode Decomposition,» 2015.

[7] S. Samaras, E. Diamantidou, D. Ataloglou, N. Sakellariou, A. Vafeiadis, V. Magoulianitis, A. Lalas, A. Dimou, D. Zarpalas, K. Votis, P. Daras et D. Tzovaras, «Deep Learning on Multi Sensor Data for Counter UAV Applications—A Systematic Review,» *Sensors,* 2019.

[8] B. Taha et A. Shoufan, «Machine Learning-Based Drone Detection and Classification: State-of-the-Art in Research,» *IEEE Access,* 2019.

[9] J. S. Patel, F. Fioranelli et D. Anderson, «Review of radar classification and RCS characterisation techniques for small UAVs or drones,» 2018.

[10] N. Mohajerin, J. Histon, R. Dizaji et S. L. Waslander, «Feature extraction and radar track classification for detecting UAVs in civillian airspace,» *2014 IEEE Radar Conference,* 2014.

[11] M. Messina et G. Pinelli, «Classification of Drones with a Surveillance Radar Signal,» 2019.

[12] V.-P. Thai, W. Zhong, T. Pham, S. Alam and V. Duong, “DETECTION, TRACKING AND CLASSIFICATION OF AIRCRAFT AND DRONES IN DIGITAL TOWERS USING MACHINE LEARNING ON MOTION PATTERNS.,” 2019.

[13] S. Srigrarom, K. H. Chew, D. M. Da Lee et P. Ratsamee, «Drone versus Bird Flights: Classification by Trajectories Characterization,» 2020.

[14] J. J. X. Chan, S. Srigrarom, J. Cao, P. Wang et P. Ratsamee, «Small Flying Object Classifications Based on Trajectories and Support Vector Machines,» *Journal of Robotics and Mechatronics,* 2021.

[15] X. Zhang, V. Mehta, M. Bolic et I. Mantegh, «Hybrid AI-enabled Method for UAS and Bird Detection and Classification,» *2020 IEEE International Conference on Systems, Man, and Cybernetics,* 2020.

[16] Y. Liu, L. Liao, H. Wu, J. Qin, L. He, G. Yang, H. Zhang et J. Zhang, «Trajectory and image-based detection and identification of UAV,» *The Visual Computer,* 2021.

[17] L. W. Jochumsen, M. O. Pedersen, K. Hansen et S. H. Jensen, «Recursive Bayesian classification of surveillance radar tracks based on kinematic with temporal dynamics and static features,» 2015.

[18] L. W. Jochumsen, J. Østergaard, S. H. Jensen, C. Clemente et M. Ø. Pedersen , «A recursive kinematic random forest and alpha beta filter classifier for 2D radar tracks,» *EURASIP Journal on Advances in Signal Processing,* 2016.

[19] R. Ginoulhac, F. Barbaresco, J.-Y. Schneider, J.-M. Pannier et S. Savary, «Coastal Radar Target Recognition Based On Kinematic Data (AIS) With Machine Learning,» 2019.

[20] K. Sheng, Z. Liu, D. Zhou, A. He et C. Feng, «Research on Ship Classification Based on Trajectory Features,» *Journal of Navigation,* vol. 71, 2017.

[21] S. Bækkegaard, J. Blixenkrone-Møller, J. J. Larsen et L. Jochumsen, «Target Classification Using Kinematic Data and a Recurrent Neural Network,» 2018.

2022-10-31

# Radar discrimination of small airborne targets through kinematic features and machine learning

Doumard, Timothée

IEEE

Doumard T, Gañán Riesco F, Petrunin I, et al., (2022) Radar discrimination of small airborne targets through kinematic features and machine learning. In: 2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC), 18-22 September 2022, Portsmouth, Virginia, USA

https://doi.org/10.1109/DASC55683.2022.9925778