

Deep learning for aircraft classification from VHF radar signatures

Jérémy Fix, Chengfang Ren, Arthur Costa Lopes, Guillaume Morice, Shuwa Kobayashi, Thierry Letertre, Israel Hinostroza

▶ **To cite this version:**

Jérémy Fix, Chengfang Ren, Arthur Costa Lopes, Guillaume Morice, Shuwa Kobayashi, et al.. Deep learning for aircraft classification from VHF radar signatures. IET Radar Sonar and Navigation, 2021, 10.1049/rsn2.12067 . hal-03227950

HAL Id: hal-03227950
https://hal.science/hal-03227950

Submitted on 17 May 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Received: 27 March 2020 | Revised: 13 July 2020 | Accepted: 7 August 2020

DOI: 10.1049/rsn2.12067

IET Radar, Sonar & Navigation

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

Deep learning for aircraft classification from VHF radar signatures

Jérémy Fix[1] | Chengfang Ren[1] | Arthur Costa Lopes[3] | Guillaume Morice[3] | Shuwa Kobayashi[3] | Thierry Leterte[2] | Israel D. Hinostroza Sáenz[2]

[1]Université de Lorraine, CentraleSupélec, CNRS, LORIA, Metz, France

[2]Université Paris-Saclay, CentraleSupélec, ONERA, SONDRA, Gif-sur-Yvette, France

[3]CentraleSupélec, Metz, France

Correspondence

Israel D. Hinostroza Sáenz, Université Paris-Saclay, CentraleSupélec, ONERA, SONDRA, 91190 Gif-sur-Yvette, France.
Email: Israel.Hinostroza@centralesupelec.fr

Abstract

Radio sources in the Very High Frequency (VHF) band can be seized as opportunity donors in a passive radar configuration such as FM radio stations and VHF omnidirectional range (VOR). A full-wave simulation of three size classes of aeroplanes shows that their bistatic radar cross-section (RCS) are statistically comparable, albeit perform differently in time while the plane is flying. This difference can be exploited to recognize the size of the aeroplanes with respect to these classes. Measurements confirm this possible differentiation between the aeroplanes within the same class. Encouraging initial results were obtained using convolutional or recurrent neural networks to classify aircraft classes, combining simulated bistatic RCS results and real trajectories (collected from automatic dependent surveillance-broadcastdata).

1 | INTRODUCTION

Automatic target recognition has generated great interest in aeroplane traffic control and surveillance. The increasing number of radar opportunity illuminators, for example, radio transmitters, digital audio/video broadcast transmitters, and mobile-phone base stations, can be used as natural sources of radio waves in passive radar systems. Notably, broadcast commercial transmitters of FM radio (88–108 MHz) [1, 2] and VHF omnidirectional range (VOR, 108–118 MHz) [3–6] prove attractive due to their high power transmission and wide coverage.

The diversity of opportunity illuminators offers the possibility of setting multistatic recognition systems that combine several signals scattered from a target. Acquired data in these systems is more abundant than in a bistatic configuration, without increasing the risk of being detected. In these bands mentioned above, the wavelength is between 2.5 and 3.4 m, making commercial aeroplane identification challenging since their size is on the order of tens of wavelengths. Despite this, low-frequency signals are less susceptible to weather changes, unlike optical signals. Consequently, the radar cross section (RCS) in VHF could be a useful fingerprint to identify the class of aeroplane, as shown in [6–8]. Although ground reflection interference [9] (which may also interfere with the noise at the receiver), wave polarization [8] and aeroplane heading [10] can make extremely difficult the extraction of these values in some particular cases. By exploiting the automatic dependent surveillance-broadcast (ADS-B) data broadcasted (L band) by the aeroplanes, their position and heading are then known and their BiStatic-RCS (BS-RCS), that is, RCS for separated transmitter and receiver, can be effectively estimated as in [5].

In the literature, model- and data-based classifiers [6, 7, 11–13] have been applied to similar problems showing the feasibility of aeroplane recognition from their BS-RCS signature along the flight path. However, deep learning classifiers are yet to be used, given their tremendous success in computer vision for various classification problems [14]. Unlike recognition from optical images, typically two-dimensional [2D] spatial map, BS-RCS signature can be either represented as four-dimensional data (RCS vs. 2D coordinates of incidence angle and 2D coordinates of scattered angle) or as a temporal vector series (RCS and angles coordinates vs. time). While both features appear insightful for aeroplane recognition, either convolutional neural networks (CNNs) or recurrent neural networks (RNNs) are proposed to deal with different representations of RCS. Pre-processing is an important step when dealing with a sparse map of data allowing for faster training of the neural networks when the preprocessed images exhibit more discriminant patterns.

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2021 The Authors. *IET Radar, Sonar & Navigation* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

Here, we highlight, through simulation and measurements, discriminant RCS patterns for different sizes of planes. In Section 2 we present simulation results of BS-RCS for three classes of aeroplanes (small, medium, and large) towards better understanding their similarities and differences. Then, in Section 3 we describe how the simulations results are pre-processed to facilitate the neural network training.

Finally, in Section 4, we train several neural networks to classify the aeroplane following several approaches: a 2D-CNN classifier from the sparse BS-RCS, a one-dimension (1D)-CNN classifier from the RCS time series and angular position of the plane with respect to the illuminator and receiver and finally an RNN from only the RCS time series. A discussion of the promising results and future extensions then concludes the study.

2 | MODELING AND SIMULATION OF AIRPLANES

2.1 | Aeroplane models

The dimensions and number of engines of each aeroplane model (large, medium, and small) are shown in Table 1 (see Figure 1 for a scaled comparison). A perfect electrical conductor (PEC) was considered for the models.

TABLE 1 Aeroplane models dimensions

Model size	Wing span (m)	Length (m)	Number of total engines
Small	24	29	2
Medium	33.5	40	2
Large	51	60	4

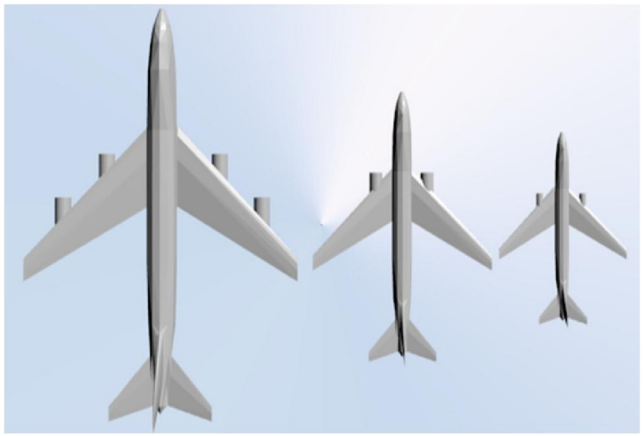


FIGURE 1 From left to right: large, medium, and small aeroplane models used in field calculations involving bodies of arbitrary shape (FEKO)

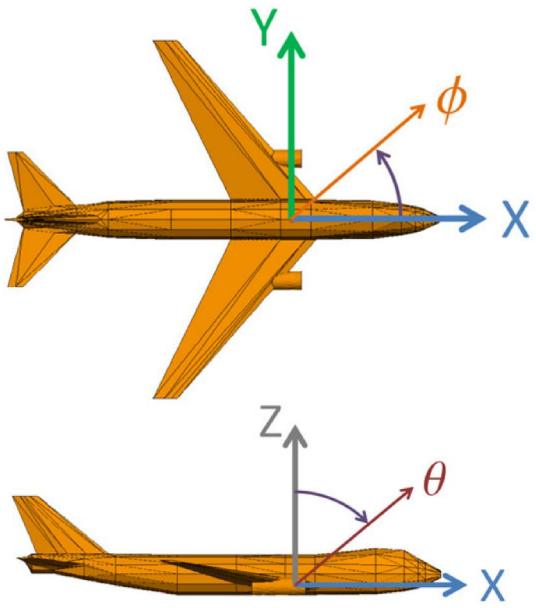


FIGURE 2 Simulation setup of aeroplane models, PEC, in FEKO and definition of spherical coordinates used in the simulations. PEC, perfect electrical conductor

2.2 | Simulation results: RCS

Since the size of the aeroplane is comparable to the working wavelength, no approximation is used, as in [15]. A full-wave simulation using the method of moments was carried out in FEKO (3D electromagnetic simulator). To increase accuracy, especially on the cross-polarization simulation results, the multilevel fast multipole method is not used. See Figure 2 for the simulation setup and spherical coordinates used for the simulation. All the simulations were carried out using a 115 MHz monochromatic linearly polarized planar wave.

Here, we have chosen to use simpler models. These models follow approximate shapes of the different planes used in civil aviation, but their main characteristics (fuselage length, wingspan, and number of engines) influencing the values of RCS correspond to the three classes.

Incident horizontally (resp. vertically) polarized (along ϕ axis) plane waves were considered with incident direction $(\theta_i, \phi_i) \in [90, 180] \times [0, 180]$ (due to the symmetry of the planes, incident angles ϕ_i larger than 180° are obtained from the original simulations when synthesizing RCS evolution in time) and scattered angles $(\theta_s, \phi_s) \in [0, 180] \times [0, 360]$, in both cases with the angular step of 1°.

Figure 3 shows the histograms of BS-RCS values of the aeroplanes (small, medium, and large) for VV $(\theta - \theta)$, HH $(\phi - \phi)$, and HV $(\phi - \theta)$ cases and for all scattered and incident angles considered lines above. We observed most of the RCS values as being concentrated between -10 dBm^2 and 40 dBm^2 (in accordance with [5], Airbus 321 is comparable to the medium size model used lines above). The histograms of RCS between the aeroplanes reveal very similar values, rendering discrimination from RCS values only difficult. Additionally, there is not much difference between RCS values for the VV, HH, and HV cases.

Although the overall BS-RCS values are similar, for a small range of incident and scattered angles, some differences are

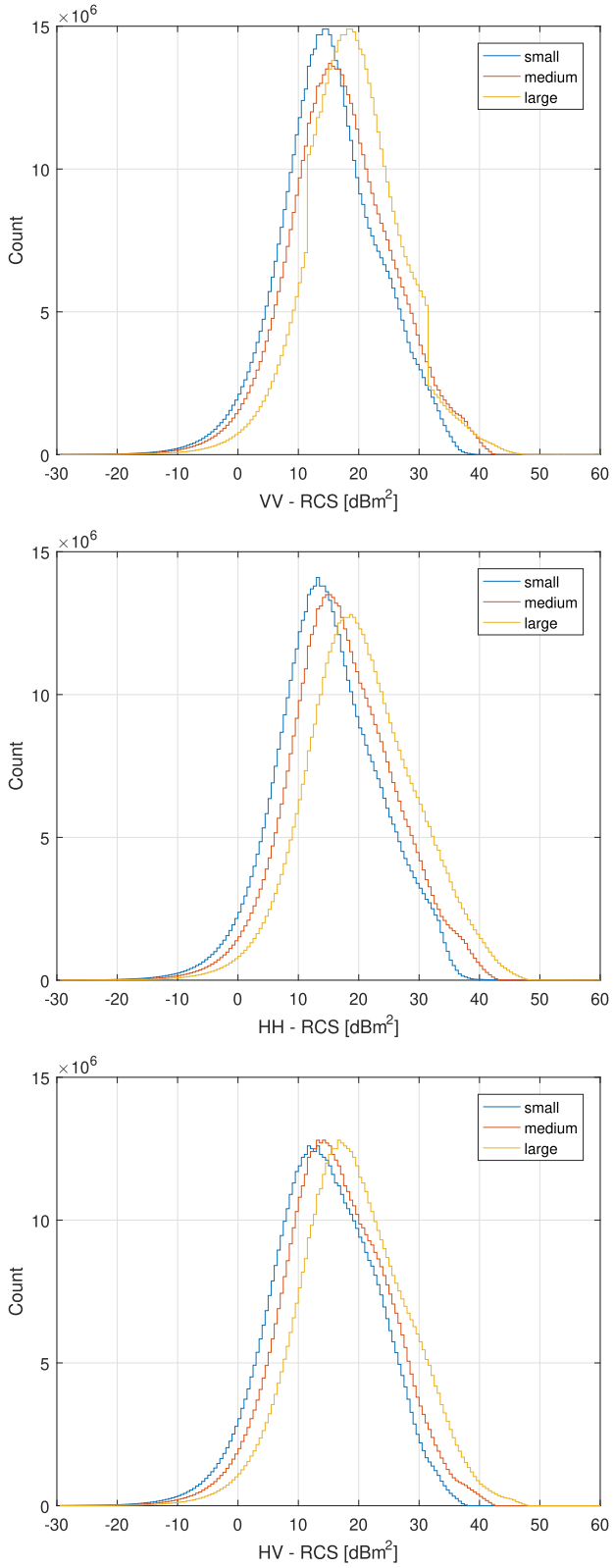


FIGURE 3 Histograms of BS-RCS (dBm^2), 115 MHz, obtained by simulation, VV ($\theta - \theta$, top), HH ($\phi - \phi$, middle), and HV ($\phi - \theta$, bottom), for small, medium, and large aeroplane models. BS-RCS, BiStatic-radar cross-section

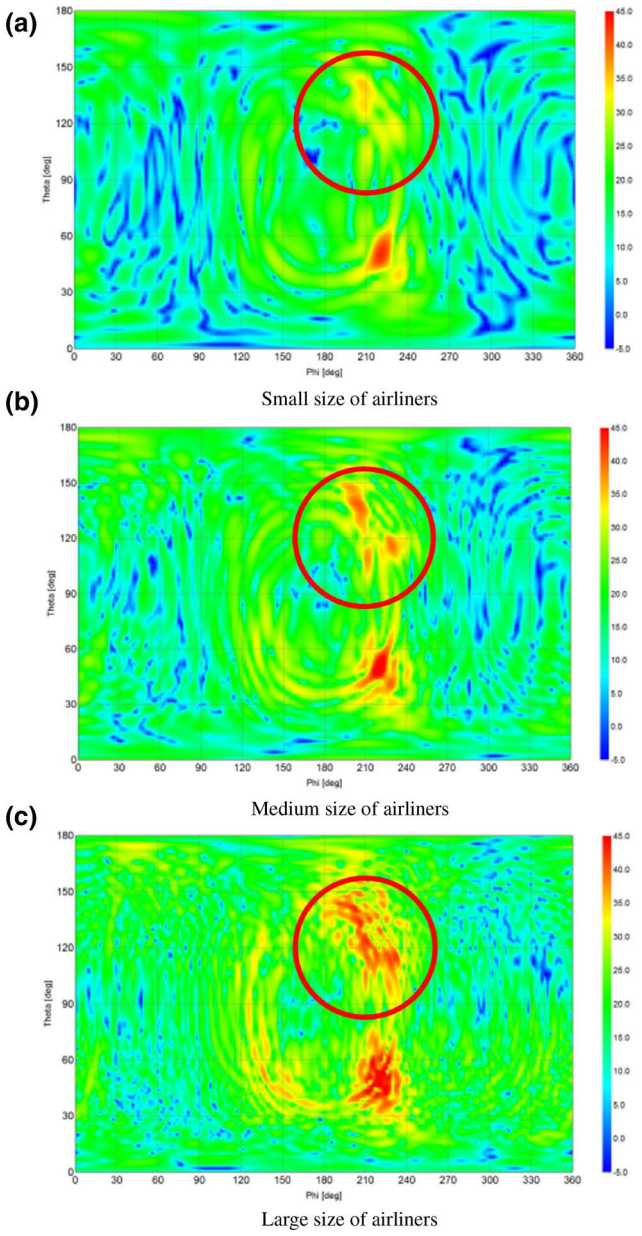


FIGURE 4 VV BS-RCS, 115 MHz, for different scattered angles (horizontal axis ϕ and vertical axis θ) considering an incident planar wave coming from $\theta = 130°$ and $\phi = 40°$. Red circle highlights the differences. BS-RCS, BiStatic-radar cross-section

easily observable. Figure 4 shows the BS-RCS values when fixing the direction ($\theta = 130°$and $\phi = 40°$) of an incident plane wave and considering all the scattered angles. From this figure, especially inside the red circle, we can discern differences related to the size of the planes.

2.3 | Simulation results: RCS time series

Using the simulation results and ADS-B data from the aeroplanes, we can also synthesisze the variation of the BS-RCS of

the aeroplanes for different trajectories. For this case, we consider a passive radar configuration where the transmitter (Tx) is at $48°39'14''N$, $1°59'39.2''E$ (close to Rambouillet, France), and the receiver (Rx) is near CentraleSupélec, Gif-sur-Yvette, France.

Figure 5 shows the position of the Tx, Rx, the plane's trajectory (obtained from real ADS-B data for about 6 min), and a 'zoom' version of the trajectory (between 3.5 and 5.5 min), all of them in Lambert 93 coordinates (spatial units in metres). The aeroplane moves from left to right (approaching landing at Orly airport). The RCS versus time (in min, 2 Hz of sampling) for this trajectory is shown for each of the three plane sizes and polarizations (VV, HH, and HV) in Figure 6.

From Figure 6, we note that the ranges of values of RCS, per polarization, are similar (as expected from the histograms, cf. Figure 3) besides a peak of 7 dB (around 3.85 min) above the remaining cases, which corresponds to the large aeroplane.

What is different is the frequency of fluctuation of the received power (see Figure 6), which varies according to the size of the aeroplane: high frequency for large plane, medium frequency for medium plane and low frequency for small plane, and all polarizations. We recall that, in our simulation, we used the same trajectories for each aeroplane class; the space of angles is swept at the same rate. The frequency of fluctuations indicates, in this case, the angular frequency due to the sweeping throughout the peaks (width and number) of the scattered wave due to the illumination of the object: large objects will induce a greater number of thinner peaks compared to smaller objects, as in Figure 4.

We can also see that the RCS of the HH and VV polarizations are about 15 dB higher than the HV polarization (although this might not always be the case, see histograms, Figure 3). We note that in all the cases, we have used a single trajectory and single movement dynamics (same speed and orientation); hence, the frequency of fluctuations of RCS in real life will depend on the speed of the planes.

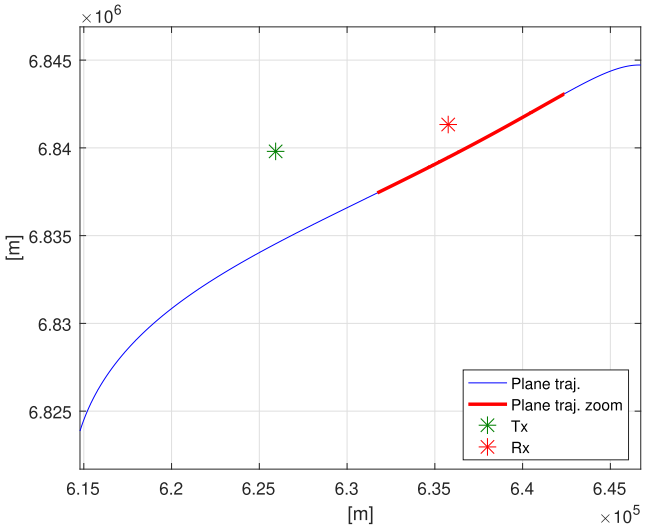


FIGURE 5 Position of Tx (near Rambouillet) in green, Rx (near CentraleSupélec-Gif) in red, plane's trajectory in blue (ADS-B data for 6.3 min), and the trajectory zoom in thick red line (between timestamps 3 and 5 min). The plane is moving from left to right. Lambert 93 coordinates (ms). ADS-B, automatic dependent surveillance-broadcast

3 | PRE-PROCESSING

To train the neural network, we synthesized the evolution of bistatic RCS values considering real flight paths. An ADS-B decoder was used to retrieve real trajectories of the aeroplanes and their type. Note that the trajectories of the planes (for this data collection) are very similar: approaching Orly airport. We collected 269 trajectories, using ADS-B data and synthesized the bistatic RCS values for each trajectory, class of plane sizes, and sources (11 VORs around Paris). Two different data representations were used: 2D images and 1D vector.

3.1 | 2D image, aspect: bistatic angles

A natural way of representing the bistatic radar data is through images of BS-RCS values for bistatic and aspect angles [4, 5]. Figure 7 shows the definition of these angles used here. A database was constructed using 2D images (181 × 361 pixels, 1 pixel per angle) of BS-RCS variations for each trajectory and 11 sources (see Figure 8) where the colours represent the relative BS-RCS values in dBsm (for VV polarization) for each coordinate.

The blue zones in Figure 8 correspond to the absence of ADS-B data (hence, no RCS is attributed to those coordinates).

Note that in this case, information of the angular position and orientation of the plane is given indirectly to the neural network.

3.2 | 1D vector

A 1D vector data of synthesized RCS values, ordered according to time, was also considered to train the neural network. Neither information on position, nor information on orientation, of the plane, was available (see Figure 6). While taking measurements, we noticed that we could easily differentiate the received power evolution of each aeroplane.

As an example, Figure 9 shows the evolution of the relative received power (after pre-processing) in dB for four aeroplanes and two classes: small and medium. The planes were approaching Orly airport and, hence, followed a similar flight path although not precisely the same path, speed, and attitude). The received power in Figure 9 contains the reflected signal on the aeroplane (from RBT VOR) and the direct path; these signals are strongly attenuated due to losses in the propagation.

From Figure 9 we can appreciate the differences in the power evolution between classes: in amplitude in general, as well as in relative level between the main peak (highest level) and secondary ones (close to the main peak). At this stage, we recall that we cannot compare the frequency of fluctuations on the received power due to the different speeds of the plane. It

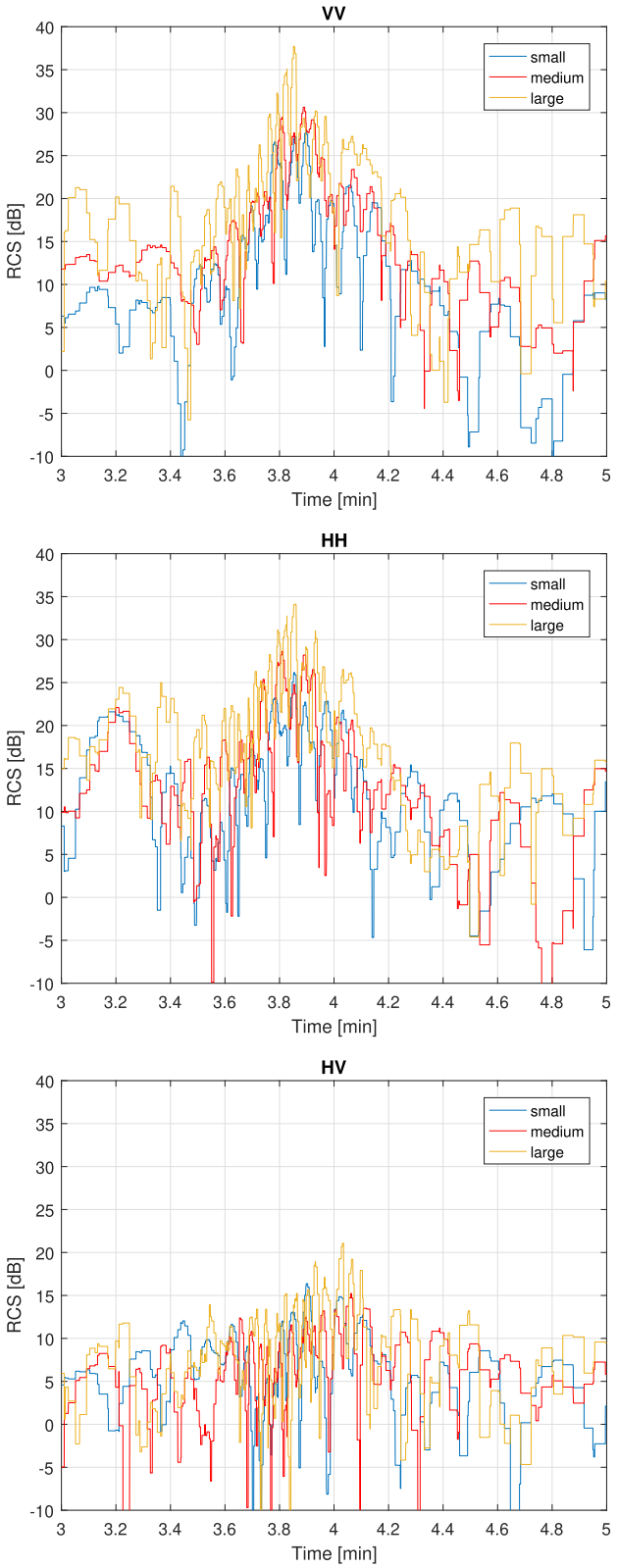


FIGURE 6 Evolution of BS-RCS of aeroplanes (VV, HH, and HV, from top to bottom, respectively), for 2 min, considering a source at 48° 39′14″*N*, 1°59′39.2″*E* and receiver near CentraleSupélec. The trajectory of the plane is the one labelled as 'Plane traj. Zoom' (thick red line) in Figure 5. BS-RCS, BiStatic-radar cross-section

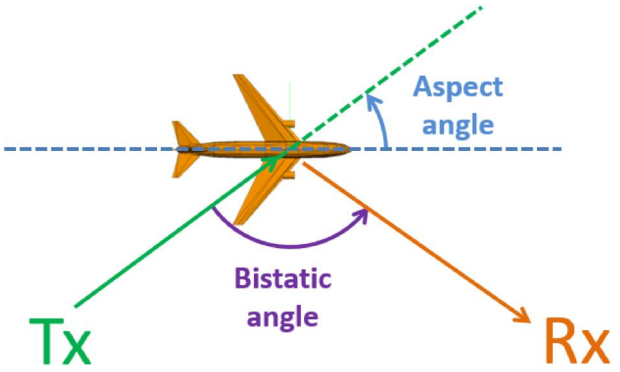


FIGURE 7 Definition of bistatic and aspect angles

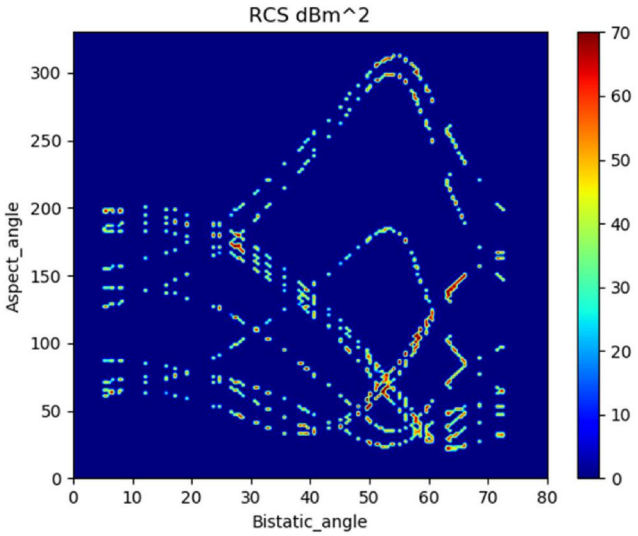


FIGURE 8 2D image of BS-RCS variations (in colour) for bistatic and aspect angles, 181 × 361 pixels (similar to [5]), for a single trajectory and 11 VORs (hence, 11 different illumination angles of the aeroplane). BS-RCS, BiStatic-radar cross-section; VOR, VHF omnidirectional range

can also be seen that there are differences between the two aeroplanes of the small class. Note that the power evolutions related to the medium class (Boeing 737) are very similar since they have similar flight path and attitude.

A software-defined radio (SDR), along with a software developed by SDR Technologies, was used to collect the aeroplane's scattered power (I-Q sampling). It was centred at VOR frequency (114.7 MHz) and connected to a PC (commanded by a Python script). The receiver was installed close to CentraleSupélec-Gif using a horizontally polarized VHF antenna.

4 | CLASSIFIERS BASED ON DEEP LEARNING

Our classifier aims to predict a label $y \in \{1, 2, 3\}$, which respectively maps to small, medium, and big classes of

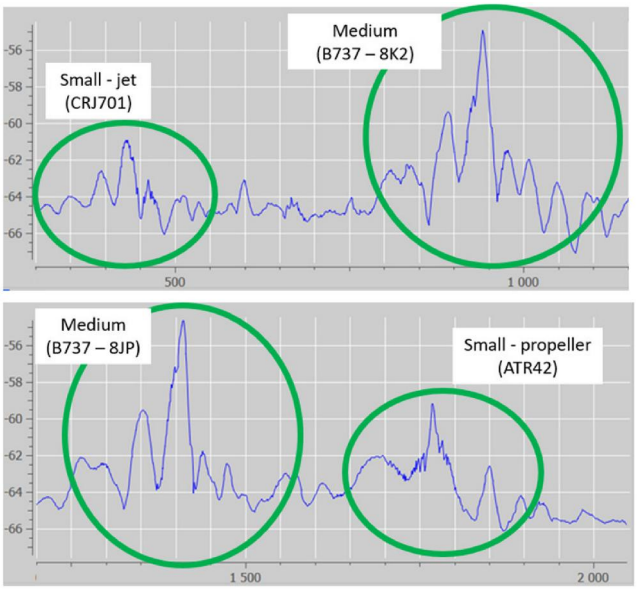


FIGURE 9 Received relative level (dB) of power, HH, after pre-processing of reflected signal on aeroplanes approaching Orly airport. Two classes are present; on the top figure, a small (jet CRJ701) and a medium size (Boeing 737) aeroplane. On the bottom figure, a medium (Boeing 737) and a small size aeroplane (aeroplane with propeller ATR-42 500)

aeroplanes, from input features $X \in \mathcal{X}$, which can be either a vector, a matrix, or a tensor. The classifier is mathematically represented by an operator:

$$f_P : \mathcal{X} \to \{1, 2, 3\}$$
$$X \mapsto \hat{y} = f_P(X)$$

where P is all the parameters of the classifier.

In a supervised learning framework, we usually distinguish from P two types of parameters for neural networks:

(i) learnable parameters W from labelled dataset $\{(X_i, y_i)\}_{i=1,\dots,N}$ generally called the weights,
(ii) hyperparameters H, for example, the number of layers, filters, choice of regularizers, optimizer, dropout, and so on.

The main difference between these parameters lies in the fact that H is only tunable in a heuristic manner [16], whereas W is learnable by minimizing the following cost function, also called the empirical risk:

$$W_{opt} = \arg\min_W \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left(y_i, f_P(X_i)\right) \quad (1)$$

where $\mathcal{L}$ is generally the cross-entropy loss function for classification problems. Solving Equation (1) is complicated since the empirical risk is non-convex w.r.t. W and the training set could be huge. To reduce this computational cost, W is iteratively updated using mini-batch gradient descent with optimizer as RMSprop, ADAM [17], and so on. Furthermore the mini-batch gradient is usually computed with the back-propagation algorithm [18].

In the following subsections, we consider two cases for $\mathcal{X}$ being either the partially observed BS-RCS or the BS-RCS time series with or without the emitter/receiver polar and azimuth angles.

For all the experiments, the models are trained with Tensorflow [19] and Keras [20] on an Nvidia GTX 1080Ti.

4.1 | Classification from the sparse BS-RCS

As shown in Section 2.2, the BS-RCS contains highly specific features pertaining to the aeroplane model and its content, had it been fully observed, could be used as a fingerprint of the plane allowing for its classification, see Figure 4. However, when a plane flies along a trajectory, only a subpart of BS-RCS is observed w.r.t. four angle coordinates (θ_i, ϕ_i, θ_s, and ϕ_s). For solving this sparsity issue, we first use multiple transmitter signals located at different positions to cover more incidence angles. Second, we reduce the dimension of the features by considering only a 2D map containing the BS-RCS w.r.t. aspect and bistatic angles. We collected 269 airliner trajectories from ADS-B data around Paris. For each trajectory, 11 BS-RCS are synthesized with a single receiver location and across 11 different positions of illuminators. In our scenario, these locations correspond to VOR donor positions, but other types of transmitters, such as radio or DVB-T stations, could be considered. For any sparse map, a clear distinction between measured and unobserved BS-RCS is needed for forcing neural network's attention to capture BS-RCS variation along the trajectory path. From the BS-RCS histogram (see Figure 3), an offset of 30 dB m^2 is added, such that 0 dB m^2 corresponds to the unobserved regions. In Figure 10, the stacked BS-RCS using all collected trajectories is plotted. Each map contains $269 \times 11 = 2959$ BS-RCS trajectories due to the diversity of transmitter positions.

In the first glance, this map appears to be an excellent candidate for discriminating each class of aeroplane. In practice, the class of aeroplane should be recognized from a single trajectory; therefore, the BS-RCS map will be incomplete. Indeed, for n transmitters and m receivers placed at different locations, we will have at most $n \times m$ trajectories in the BS-RCS map by combining the incident and reflected angles. In Figure 8, a typical BS-RCS map is plotted for one single airliner trajectory using 11 illuminator positions and one single receiver. The discrimination regions are less obvious than from the stacked BS-RCS (Figure 10). Fortunately, CNNs are, today, the state-of-the-art algorithms for image classification [14] and more sensitive to variation discrepancy than human eyes. Additionally, this architecture of deep learning is motivated by the performance provided in visual image recognition fields even though our data are slightly different from optical images and are sparse because most angular domains remain unseen. Therefore, we proposed Mercury2D, a 2D-CNN for classifying

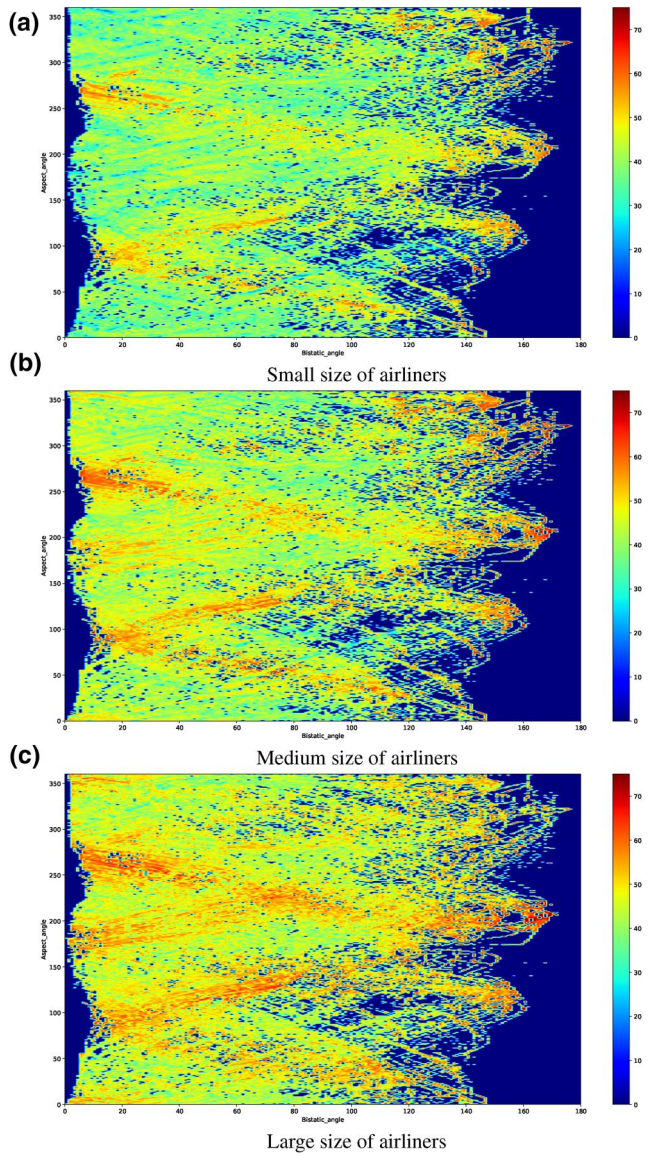


FIGURE 10 Stacked BS-RCS map with 30 dB m^2 offset versus aspect and bistatic angles for each class of airliners (colour bar dB m^2). BS-RCS, BiStatic-radar cross-section

these maps. The architecture is described in Table 2. As the network aims to classify images, the architecture follows standard design principles in deep learning applied to computer vision problems: the architecture is built from stacking blocks of a 2D convolutional layer, a non-linear ReLu (Rectified Linear Unit) activation, a pooling layer and a dropout layer followed by two fully connected layers. The number of convolutional kernels and the value of the dropout are all hyper-parameters (in H) to be found before running the optimization of W and these are specified in the table. The activation of the first fully connected layer is a ReLu while the last layer uses a Softmax for outputting a probability distribution over the three classes under consideration.

A total of 807 BS-RCS maps is used to train, validate, and test the Mercury2D network. 70% of the whole dataset is used for training ($N = 565$ in Equation (1)), 15% for the validation and 15% for testing. During the training phase, Mercury2D weights W are randomly initialized using the Glorot uniform initializer [21] and are optimized over 200 epochs with the categorical cross-entropy loss. For each epoch, a mini-batch gradient is computed over 64 samples, ADAM optimizer is used to update the weights W. The learning rate is a hyper-parameter set to $\mathrm{lr}(\mathrm{ep}) = \frac{0.001}{1+0.1\times \mathrm{ep}}$, where ep denotes the current epoch number. Mercury2D has 3, 427, 811 trainable parameters W (the values of the convolutional kernels and the weights of the fully connected layers). Loading the data and training the network last in total 135 min.

TABLE 2 Mercury2D architecure. Conv2D ($s_1 \times s_2 \times c$) is a convolution of kernel size ($s_1 \times s_2$) with c output channels, stride (1×1). MaxPool2D ($s_1 \times s_2$) is a max-pooling with kernel size ($s_1 \times s_2$) and stride ($s_1 \times s_2$)

Input size	(181 × 361)
Convolutional layers	Conv2D (6 × 12 × 32), ReLu,
	MaxPool2D (3 × 6), Dropout (0.5)
	Conv2D (6 × 6 × 64), ReLu,
	MaxPool2D (3 × 3), Dropout (0.5)
	Conv2D (3 × 3 × 128), ReLu,
	MaxPool2D (2 × 2), Dropout (0.5)
Fully connected layers	Fully connected (512), Relu
	Fully connected (3), Softmax

A fine-tuning of Mercury2D hyper-parameters H, that is, the number of layers, size of kernel, number of filters and dropout, and so on, is performed in a heuristic manner to avoid under- and over-fittings. In Figure 11, we plotted the evolution of categorical cross-entropy loss and accuracy as a function of the number of epochs. As shown, loss and accuracy metric of train, validation and test are approximately stabilized after 150 epochs. The best accuracy is around 90% with a loss of 0.38. We can notice that the validation set and test set have similar behaviour potentially indicating the correct prediction of generalization error. However, there is a gap between the train and test/validation performance, almost 10% drop in terms of accuracy. This gap shows the difficulty of Mercury2D to generalize using a sparse BS-RCS map even with 11 trajectories.

4.2 | Classifying from temporal series

In the previous section, the aeroplane model is classified from the BS-RCS, which is only sparsely observed. We now consider the problem as a temporal series classification, that is, directly classifying the aeroplane model from a 5-dimensional sequence of BS-RCS and polar/azimuth angles w.r.t. the illuminator and receiver. We will consider two typical architectures for classifying the temporal series: 1D-CNN, called Mercury1D and an RNN, called Jupyter1D.

The networks are all trained for 200 epochs, with the ADAM optimizer ($\alpha = 0.001$) and a cross-entropy loss. The

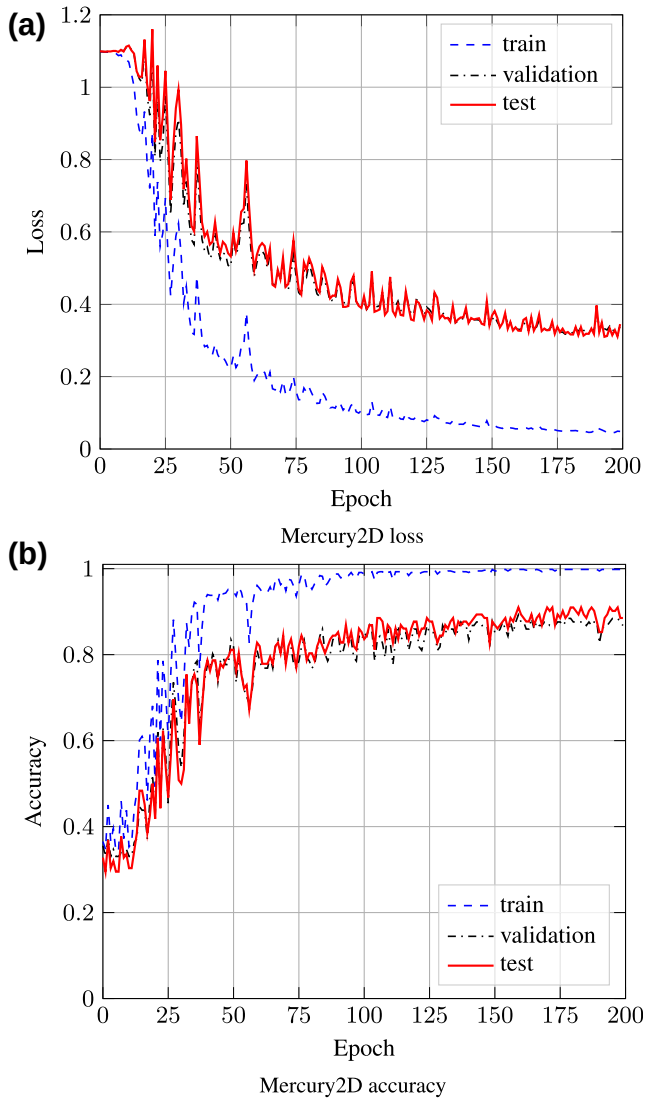


FIGURE 11 Loss and accuracy w.r.t. number of epochs for Mercury2D

learning rate is adapted with the following schedule: $\alpha = 10^{-3}$ for the first 80 epochs, $\alpha = 5.10^{-4}$ for the next 60 epochs then $\alpha = \frac{1}{3}10^{-3}$ for the next 40 epochs and $\alpha = 2.10^{-4}$ for the last 20 epochs. This adaptive learning rate was chosen empirically so as to hopefully get a better classifier.

The angular excursion of the trajectories depends on the visibility of the plane from the illuminator and receiver. When pre-processing the trajectories, they have been resampled at a variable angular step so as to contain 390 datapoints. The polar/azimuth angles with respect to the illuminator and receiver are normalized in [0, 1]. The BS-RCS is not normalized.

In the metrics reported in the next sections, the real risk is estimated with a threefold cross-validation. This means, for each fold, approximately 2/3 of the data for the training set and 1/3 for the validation and test set. The splits contain $N = 5918$ series (538 trajectories with 11 illuminators) for the training set, 1485 series for the validation set (135 trajectories with 11 illuminators), and 1474 series for the test set (134 trajectories with 11 illuminators). The validation set is used for early stopping, and the accuracy on the test sets are averaged over the folds to report the real risk estimation of the trained classifiers.

In addition to the performance metrics of the model processing a single BS-RCS time series, we consider combining the predictions from the 11 illuminators. The trained model takes as input the BS-RCS time series from a single illuminator. However, the model can be tested by either considering the prediction from a single BS-RCS time series (from a single illuminator) or the predictions from the 11 illuminators. These 11 predictions can be combined, using either a voting strategy or an averaging strategy. For the voting strategy, the assigned class is the one getting the highest number of votes from each of the 11 predictions. For the averaging strategy, the 11 class probabilities are averaged before determining the class of the plane. For computing the average probability over the aeroplane classes, each probability vector (associated with the trajectory of the plane seen from one illuminator) is weighted by the original angular length of the trajectory, owing to the hypothesis that a longer trajectory implies higher confidence in the discrimination of the aeroplane's class.

4.2.1 | Mercury1D: 1D convolutional neural network

The 1D-CNN architecture is built from three successive convolutional blocks followed by a hidden layer and the last softmax layer as given in Table 3.

In addition to the dropout layers, all the convolution layers are L2-regularized with a factor of 0.008. The amount of dropout and regularization factor belongs to the hyperparameters H found empirically. This architecture has a total of 1.775.641 trainable parameters W (convolutional kernels and weights of the fully connected layers). For training, a minibatch contains 64 samples. Loading the data and training a model (three times, one for each fold) takes around 20 min. The performances of the best (in terms of their validation loss) out of 10 runs are given in Table 4. The accuracies estimated from the threefold cross-validation are also provided: they do not differ significantly from the performances of the best single run. Combining the individual predictions of the model on all 11 illuminators, by either averaging or voting, brings a better predictor ($\approx +$ 6% accuracy) than the one with a single illuminator.

The evolution of losses and accuracies of the best single run (the one leading to the model with the lowest validation loss), out of 10 runs, are shown in Figure 12.

4.2.2 | Jupyter: GRU recurrent neural network

The RNN is built from one layer of GRU cells, followed by a dense layer and the last softmax layer as given in Table 5.

TABLE 3 Mercury1D architecure. Conv1D ($s \times c$) is a convolution of kernel size s with c output channels, stride 1. MaxPool1D(s) is a max-pooling with kernel size s and stride s

Input size	(390 × 5)
Convolutional layers	Conv1D (10 × 80), ReLu,
	MaxPool1D (5), Dropout (0.2)
	Conv1D (15 × 130), ReLu,
	MaxPool1D (5), Dropout (0.35)
	Conv1D (10 × 180), ReLu,
	MaxPool1D (4), Dropout (0.4)
Fully connected layers	Fully connected (512), Relu
	Dropout (0.5)
	Fully connected (3), Softmax

TABLE 4 Mercury1D accuracies

	Single illuminator	Voting	Averaging
3-fold cross-validation	89.16%	94.52%	95.52%
Single run (test set)	90.91%	95.52%	97.01%

Besides, to test the performances of an RNN, this model also only takes the BS-RCS time series and not the polar/azimuth angles as for the model in the previous section. The benefit of getting rid of the angles is that it allows us to get rid of the ADS-B antenna. Indeed, in a real application, ADS-B may give false information (intentionally or not).

The architecture has a total of 265.475 trainable parameters W (weights of the gates of the GRU units and weights of the fully connected layers). For training, a minibatch contains 512 samples. Loading the data and training a model (three times, one for each fold) also takes around 20 min. The performances of the best model, when it comes to validation loss, out of 10 runs are given in Table 6. The accuracies estimated from the threefold cross-validation are also provided; they do not differ significantly from the performances of the best single run. Combining the individual predictions of the model on the 11 illuminators, by either averaging or voting, by far yields a better predictor (from +10% to + 15%) than the one with a single illuminator. The benefits of combining the predictors on multiple illuminators are much higher than with Mercury1D.

The losses and accuracies of the best run out of 10 runs are shown in Figure 13. The performances of the RNN are pretty surprising. Indeed, the model does not have access to the angles and therefore has no obvious means by which to know the parts of the BS-RCS that are sampled. Also, as the BS-RCS time series is stretched to fit a length of 390 samples, the fequency content of the series gets distorted. It is, therefore, rather astonishing that the model can classify the plane from such input, and further investigations are required to fully understand why it performs so well.

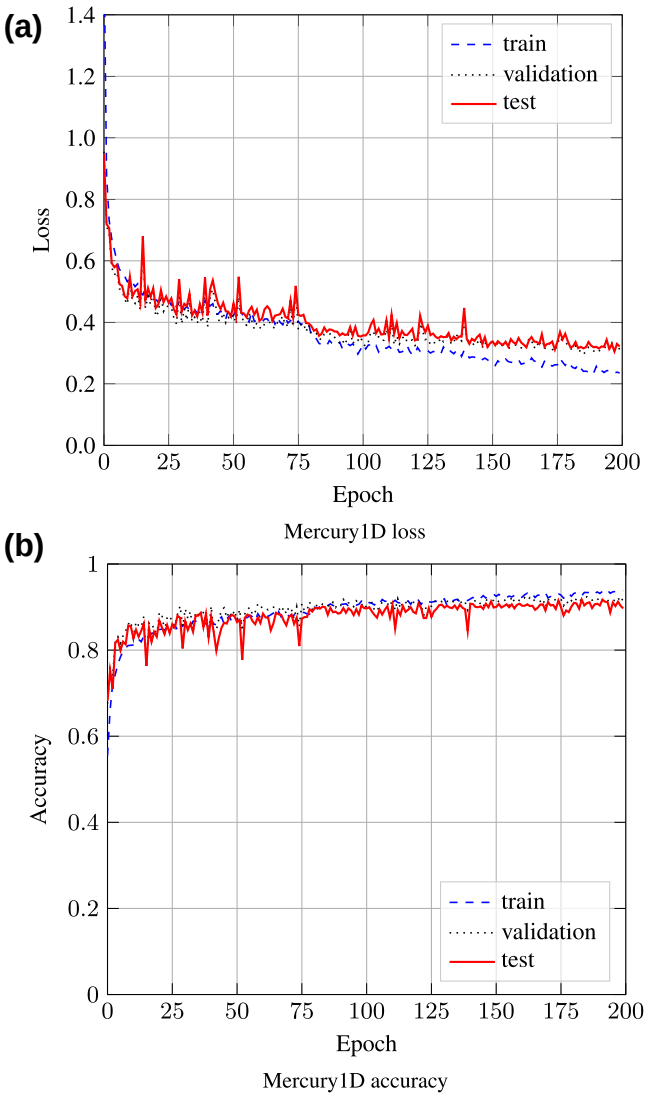


FIGURE 12 Loss and accuracy as a function of epochs for Mercury1D

TABLE 5 Jupyter architecure

Input size	(390 × 1)
Recurrent layers	GRU (256, L2 regularized 10^{-4}),
Fully connected layers	Dropout (0.5)
	Fully connected (256), Relu
	Dropout (0.5)
	Fully connected (3), Softmax

5 | CONCLUSION

Our simulations have demonstrated that, in general, the BS-RCS values of aeroplanes of different sizes (large, medium, and small) are similar, rendering their discrimination on the sole basis of BS-RCS values difficult. We have further demonstrated that it is more sensible to look for the angular frequency of the BS-RCS when the aeroplane is moving to

TABLE 6 Jupyter accuracies

	Single illuminator	Voting	Averaging
Three-fold cross-validation	82.85%	91.54%	95.27%
Single run (test set)	82.36%	92.53%	97.01%

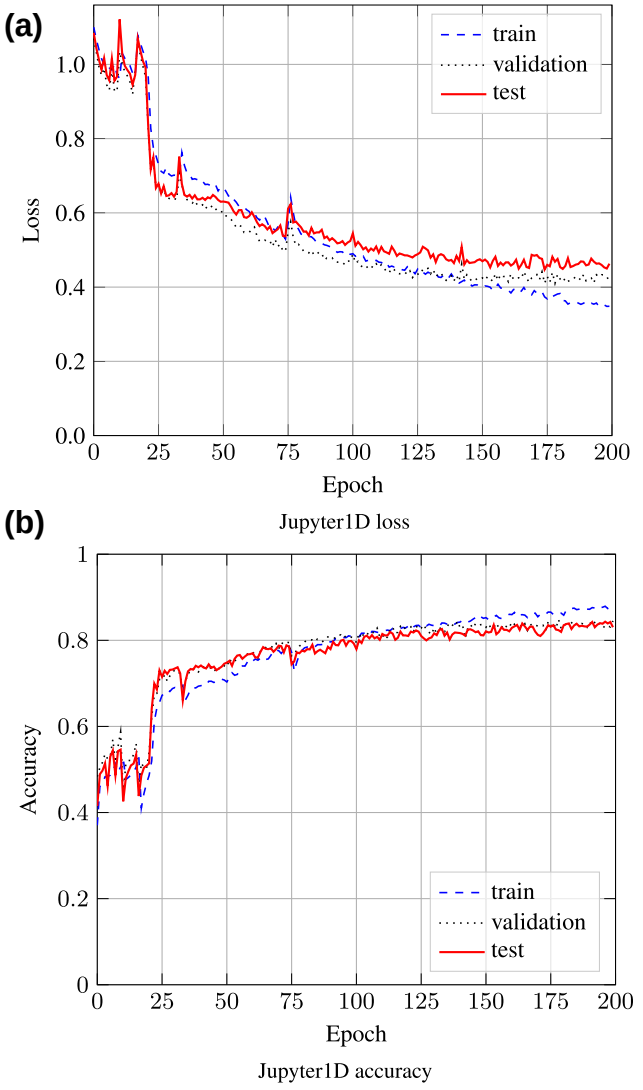


FIGURE 13 Loss and accuracy w.r.t. number of epochs for Jupyter

discriminate the planes by sizes; larger planes will scatter power with thinner and more peaks. These simulation results may be used to synthesize the BS-RCS evolution for arbitrary trajectories, which in our case is based on real trajectories, and include many radio sources and different receiver positions. Then, multiple scenarios, based on real trajectories, could be considered to train neural networks, which need a huge collection of data that are rarely available by measurement. Some measurements of the scattered power have been carried out in a passive radar configuration (Tx RBT VOR, Rx near CentraleSupélec—Gif-sur-Yvette) for aeroplanes landing at Orly airport. These experiments indeed confirm sufficient differences in the received power from the aeroplanes to discriminate between them. It might even be possible to differentiate among planes of the same class. This work only considers the RCS values but future works could investigate the impact of including the scattering-parameter phase on the classification accuracy.

We experimented three approaches: a 2D-CNN classifier of the sparse BS-RCS, a 1D-CNN classifier of the BS-RCS time series and angles, and an RNN taking as input only the BS-RCS time series. These three approaches deserve specific comments.

For the 2D-CNN, we framed the problem as a classification problem from a sparsely observed BS-RCS image, where the unobserved parts were filled with a specific value. That value was chosen to correspond to a significant attenuation of the reflected signal. It is unclear whether that specific value was the right choice. We expect to improve the performance by reconstructing the BS-RCS image with a convolutional auto-encoder before classifying it. That reconstruction is trainable because we work with simulated data so that we have both the sparse BS-RCS and the complete BS-RCS. The underlying hypothesis is that training a reconstruction network could fill the image locally by incorporating local correlations that could be learnt from the reconstruction task.

The other approach would be 1D network classifiers on the BS-RCS. In these experiments, combining the predictions from multiple illuminators yields significantly better results. This might be explained by the fact that some parts of the BS-RCS appear to be more informative than others. Further investigations are required to better understand these improvements.

The RNN works surprisingly well, considering that it does not have direct access to the angles and therefore has no obvious way of knowing which parts of the BS-RCS are observed. Further investigations are required to completely understand how the model succeeds in classifying the aeroplane from such poor and distorted input. These performances may, to some extent, rely on the regularity of the recorded trajectories approaching Orly airport. It is possible that for these recorded trajectories, the stretching of the series does not have a significant impact on the ability to classify the plane. Whether the network would still perform well on more arbitrary trajectories remains to be observed.

Finally, we still need, in the future study, to confront the network with real data, either by completely training with the real data or simply by fine-tuning a network pretrained on the simulated data.

REFERENCES

1. Lauri, A., et al.: Analysis and emulation of FM radio signals for passive radar. In: Proceedings of the IEEE aerospace conference. Big Sky, MT (2007)
2. Colone, F., et al.: Multifrequency integration in FM radio-based passive bistatic radar. Part I: target detection. IEEE Aerosp. Electron. Syst. Mag. 28(4), 28–39 (2013)
3. Azarian, S., Lesturgie, M.: Revisiting VOR transmitters: MIMO passive radar using legacy infrastructure. In: Proceedings of the international radar conference. Lille, France (2014)

4. Kayton, M., Fried, W.R.: Avionic navigation systems, 2nd ed. pp. 122–127.John Wiley & Sons Inc., USA (1997)
5. Saillant, S., et al.: Using VHF navigation aid to estimate radar cross section of airliners by inverting the radar equation. In: Proceedings of the international radar conference. Brisbane (2018)
6. Pisane, J., et al.: Automatic real-time collection of RCS of aeroplanes in a real bistatic low-frequency configuration using a software defined passive radar based on illuminators of opportunity. In: Proceedings of the IEEE radar conference. Atlanta (2012)
7. Ehrman, L.M., Lanterman, A.D.: Automated target recognition using passive radar and coordinated flight models. In: Sadjadi, F. A. (ed.) Automatic Target Recognition XIII (SPIE, 2003), vol. 5094, pp. 196–207. AeroSense 2003, Orlando, Florida (2003). https://doi.org/10.1117/12.488039
8. Bares, C., et al.: A multifrequency HF-VHF radar system for aircraft identification. In: Proceedings of the IEEE international radar conference, pp. 478–482. Arlington (2005)
9. Billingsley, J.B.: Low-angle Radar Land Clutter: Measurements and Empirical Models. IET, Norwich (2002)
10. David, A., et al.: Study of flight route effects on aircraft RCS signature at VHF frequencies by means of wire grid models. In: Proceedings of the IEEE international radar conference – radar 2000, pp. 231–235. IEEE, Washington (2000). ISBN 0-7803-5779-5. https://hal.archives-ouvertes.fr/hal-00084930
11. Pisane, J., et al.: Automatic target recognition for passive radar. IEEE Trans. Aerosp. Electron. Syst. 50(1), 371–392 (2014)
12. Pisane, J., et al.: Recognition of aeroplanes using merged subspace classifiers. In: Proceedings of the IEEE radar conference, pp. 556–560. Lille, France (2014)
13. Ehrman, L.M., Lanterman, A.D.: A robust algorithm for automatic target recognition using passive radar. In: Proceedings of the 2004 thirty-sixth southeastern symposium on system theory, pp. 102–106. Atlanta (2004)
14. LeCun, Y., et al.: Deep learning. Nature. 521(7553), 436–444 (2015). https://doi.org/10.1038/nature14539
15. David, A., et al.: Simulations and measurements of a radar cross section of a boeing 747-200 in the 20-60 MHz frequency band. Radio Sci. 38(4), 3-1–3-4 (2003)
16. Bergstra, J.S., et al.: Algorithms for hyper-parameter optimization. In: Bergstra, J.S., et al. (eds.) Advances in neural information processing systems, vol. 24, pp. 2546–2554. Curran Associates, Inc, Granada, Spain (2011). http://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-optimization.pdf
17. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Proceedings of the international conference on learning representations (ICLR). San Diego (2015)
18. Rumelhart, D.E., et al.: Learning representations by back-propagating errors. Nature. 323(6088), 533–536 (1986). https://doi.org/10.1038/323533a0
19. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous systems (2015). http://tensorflow.org/. Accessed 2 April 2021
20. Chollet, F., et al.: Keras (2015). https://keras.io. Accessed 2 April 2021
21. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, 249–256. Sardinia, Italy (2010)

How to cite this article: Fix J, Ren C, Costa Lopes A, et al. Deep learning for aircraft classification from VHF radar signatures. *IET Radar Sonar Navig*. 2021;1–11. https://doi.org/10.1049/rsn2.12067