IET Signal Processing

Research Article

IET Journals
The Institution of Engineering and Technology

Doppler processing in weather radar using deep learning

ISSN 1751-9675
Received on 4th March 2020
Revised 1st September 2020
Accepted on 2nd September 2020
doi: 10.1049/iet-spr.2020.0095
www.ietdl.org

Arturo Collado Rosell[1,2] ✉, Jorge Cogo[3], Javier Alberto Areta[3,4], Juan Pablo Pascual[1,2,4]

[1]*Instutito Balseiro, Universidad Nacional de Cuyo, Av. Bustillo 9500, Bariloche, Argentina*
[2]*Departamento de Ingeniería en Telecomunicaciones, GDTyPE, GAIyANN, CNEA, Av. Bustillo 9500, Bariloche, Argentina*
[3]*Universidad Nacional de Río Negro, Anasagasti 1463, Bariloche, Argentina*
[4]*Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Godoy Cruz 2290, Buenos Aires, Argentina*
✉ *E-mail: arturo.collado@ib.edu.ar*

Abstract: A deep learning approach to estimate the mean Doppler velocity and spectral width in weather radars is presented. It can operate in scenarios with and without the presence of ground clutter. The method uses a deep neural network with two branches, one for velocity and the other for spectral width estimation. Different network architectures are analysed and one is selected based on its validation performance, considering both serial and parallel implementations. Training is performed using synthetic data covering a wide range of possible scenarios. Monte Carlo realisations are used to evaluate the performance of the proposed method for different weather conditions. Results are compared against two standard methods, pulse-pair processing (PPP) for signals without ground clutter and Gaussian model adaptive processing (GMAP) for signals contaminated with ground clutter. Better estimates are obtained when comparing the proposed algorithm against GMAP and comparable results when compared against PPP. The performance is also validated using real weather data from the C-band radar RMA-12 located in San Carlos de Bariloche, Argentina. Once trained, the proposed method requires a moderate computational load and has the advantage of processing all the data at once, making it a good candidate for real-time implementations.

1 Introduction

Weather radar plays an important role in the society daily life. Its measurements are used to generate early meteorological alerts that help disaster prevention. It also allows the improvement of weather models, which in turn results in better forecasts [1, 2]. Signal processing is essential to extract relevant information from the measurements. In particular, Doppler processing techniques are used to obtain the first three moments of the weather Doppler spectrum, that is the power spectral density (PSD) of the received signal. The weather signal power, or zeroth moment, is related to the liquid water content and precipitation rate. The mean Doppler velocity or the first moment describes the air velocity relative to the radar. The spectral width or second moment represents the velocity dispersion due to shear or turbulence [3–5].

Several spectral moments' estimators have been proposed in the literature. However, in practice, the most common approaches are spectral processing (SP) and pulse-pair processing (PPP). SP consists of the evaluation of the spectral moments by definition, from a signal spectrum estimate. The use of the fast Fourier transform (FFT) allows a low computational load for this approach. However, the SP estimator is biased as a consequence of the finite FFT resolution [6]. In PPP the spectrum moment estimators are derived from the first two autocorrelation lag estimates. This approach also has a low computational load, but its performance degrades significantly at low signal-to-noise ratios (SNRs), <10 dB, and when meteorological target spectral width increases [7].

The derivation of the spectral moments' estimators in PPP assumes that the radar return signal is composed only of a meteorological target. However, in a general setup, the radar return signal can be modelled as the sum of contributions from meteorological targets, ground clutter – for low elevation angles – and noise. Ground clutter, also plainly called clutter, is received when the main lobe or sidelobes of the antenna illuminates static objects on the ground like mountains, buildings, trees, among others. The presence of clutter power, centred around zero velocity, often biases the meteorological target spectral moment estimates and must be taken into account to avoid this problem. It is accepted that the PSD of the weather radar signal has – on average – a well-defined structure, that can be modelled as the sum of a Gaussian-shaped weather echo spectrum, centred around the mean weather Doppler velocity, a Gaussian-shaped ground clutter spectrum around zero velocity (if clutter is present), and a white noise spectrum [8].

One of the first techniques used to mitigate ground clutter is the application of a notch filter around zero Doppler velocity [9]. While this method is simple, filtering also eliminates part of the desired signal, distorting it. Weather echoes with small radial velocity are particularly sensitive to this method. To deal with this issue, adaptive strategies that reconstruct the removed spectral components have been proposed [10–12]. The Gaussian model adaptive processing algorithm (GMAP) [10] works in the spectral domain, processing the estimated signal PSD. It uses an adaptive mask to filter the clutter and then attempts to recover the removed, – i.e. overlapped – precipitation components considering the Gaussian model for the meteorological target PSD. GMAP is computationally efficient since it works in the frequency domain, although its complexity is variable since it is an iterative algorithm. Also, as it is a spectral method it suffers the effect of time windowing which leads to spectral leakage.

GMAP has been implemented in the weather surveillance radar-1988 Doppler (WSR-88D) [12] with a ground clutter detection stage that controls its application consisting of the clutter mitigation decision (CMD) algorithm [13]. CMD combines the information from one signal phase parameter and four spatial parameters into a fuzzy logic-based algorithm to distinguish between clutter echoes and precipitation echoes. CMD presents a good performance to identify clutter in anomalous propagation and normal propagation but its performance can be improved via algorithm tuning and the integration of polarimetric variables [14].

An algorithm that follows the ideas behind GMAP but works on the time domain is the Gaussian model adaptive processing in time domain (GMAP-TD) [11]. This method applies a filter to the signal autocorrelation and recovers the precipitation signal component through an iterative procedure based on fitting a Gaussian autocorrelation precipitation model. By using signal autocorrelation instead of PSD, in the filter and reconstruction stages, GMAP-TD avoids spectral leakage. However, it is

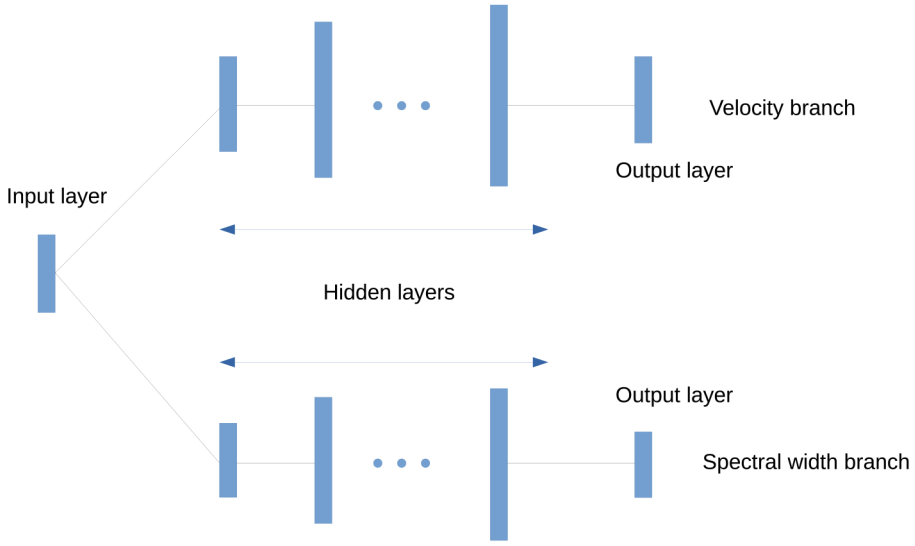

Fig. 1 *General neural network architecture*

computationally intensive because it requires the inversion of covariance matrices.

Another algorithm, clutter environment analysis using adaptive processing (CLEAN-AP) combines a detection algorithm based on the autocorrelation spectral density (ASD) [15] with a clutter removal and weather reconstruction stage [12]. The use of ASD preserves the phase information and allows detecting narrow-band interference even when not centred around zero Doppler velocity.

There are also algorithms that seek optimal estimators, like maximum likelihood or maximum a posteriori. In [16] the authors proposed a spectral moments estimation method applying the expectation–maximisation algorithm in a maximum a posteriori estimation with hyperparameters learned from the actual measurements of the ground clutter. It avoids removing the ground clutter by jointly estimating the parameters of a weather echo and the ground clutter, assuming a mixture of density functions model to the Doppler spectrum. A practical limitation of this kind of methods is the computational complexity to solve a non-convex optimisation problem with a large number of parameters to estimate.

Machine learning techniques and recently deep learning (DL) techniques have been widely used in many research fields like medical diagnosis, drug discovery, financial engineering, image classification [17–20] among others and, also, have been used in weather radar [16, 21–23]. In [21], different types of clutter (ground clutter, sea clutter and anomalous propagation) are identified using several radar measurement combinations comprising the polarimetric meteorological parameters, by means of artificial intelligence techniques including the support vector machine, the artificial neural network (NN), the decision tree, and the nearest-neighbour systems. The results show comparable performance to fuzzy logic and Bayes classifiers. Recently, a method that identifies hydrometeor types (e.g. heavy rain, wet snow, ice crystals) based on DL and the fuzzy logic algorithm has been also proposed in [22]. The model used is a convolutional NN that consists of a supervised multi-layer NN which classifies the hydrometeor type based on horizontal reflectivity, differential reflectivity, differential propagation phase constant and co-polar correlation coefficient. The algorithm was tested using a WSR-98D/XD dual-polarisation weather radar data set and it identified the main types of hydrometeors that are common in the region where the radar is located.

Finally, a ground clutter suppression method using fuzzy neural network (FNN) has been proposed in [23]. In this method, an FNN is trained with the ground clutter data acquired by dual-polarisation weather radar in clear sky conditions. Then, the ground clutter polarisation parameters are adaptively calculated and finally the ground clutter in the radar echo under the precipitation mode is identified and suppressed by the trained FNN.

In this work, a new method for ground clutter mitigation and estimation of mean Doppler velocity and spectral width of meteorological targets is proposed, based on a deep neural network (DNN). Training is done using synthetic data covering most likely combinations of situations, such as different signal-to-noise ratios, weather mean velocity, spectral width and the presence or absence of different levels of clutter. A study for selecting a good architecture among very simple to quite complex schemes for the DNN is presented.

The trained DNN is also applied to real radar data. In this case, a transfer learning approach [24] would be desirable. Unfortunately, obtaining the ground truth for meteorological data is unfeasible so this method cannot be used. Yet the results obtained with a direct application of the synthetically trained DNN are comparable to the ones of established methods like GMAP.

In terms of implementation, the use of an NN allows the parallelisation of the estimation problem, making possible to process all range cells [3] at once, thus the proposed algorithm is a good candidate for real-time implementation.

The paper is organised as follows. The method and data generation is presented in Section 2. In Section 3 different NN architectures are trained and studied to select one of them. In Section 4 the DNN performance is evaluated and compared against PPP and GMAP. Real weather radar data from RMA-12 is processed in Section 5. Finally, Section 6 presents the conclusions.

2 Method

We propose a DNN to estimate the meteorological target mean Doppler velocity, v_p, and spectrum width, σ_p; through a classification approach. The DNN is trained both in situations where clutter is present and where it is not present. Fig. 1 shows a diagram of the proposed DNN architecture; it has two branches of fully connected layers, one handling the v_p estimation and the other handling the σ_p estimation. The number of neurons in the output layer of each branch, denoted as $N_{\mathrm{o}}^{v_p}$ and $N_{\mathrm{o}}^{\sigma_p}$, is equal to the number of classes chosen for v_p and σ_p, respectively. In Section 3 a study to select the number of hidden layers and the number of neurons in each layer is carried on.

As the DNN input we use an estimate of the signal PSD – N_{i} discrete values – obtained from the in-phase and quadrature (IQ) time series corresponding to each coherent processing interval (CPI). The number of DNN input neurons, N_{i}, then depends on the PSD length. It is important to mention that we define the CPI as the number of pulses included in the antenna beamwidth [3]. The number of samples in the CPI is assumed to be constant over all the processed CPIs, and therefore so is the N_{i} value.

2.1 Data generation for training and validation

To train the DNN we generate a set of realisations of synthetic weather radar data with diverse PSD [25], considering likely situations of clutter-to-signal ratio (CSR), SNR, velocity (v_p) and spectral width (σ_p), to obtain a robust estimation model.

A very large number of realisations simulated for each situation could be used to train the model. However, there are two factors limiting this number in practice, hardware constraints, mainly RAM memory, and the time required to train the DNN, which increases with the amount of training data.

To achieve a reasonable trade-off between computation time and training quality, we divide the parameter space for data generation in a grid that accounts for the most common parameter values with reasonable separation among them. For CSR, we choose a grid of 15 uniformly spaced values from 0 to 50 dB. Also, we include the situation in which there is no clutter present. For SNR, we define a uniform grid of 15 elements from 0 to 50 dB.

In the case of σ_p we use a uniform grid of $N_o^{\sigma_p} = 8$ values, or classes, in the range from 0.04 to 0.2 of the maximum unambiguous velocity $[v_a = (\lambda/4T)]$, where T is the radar pulse repetition time (PRT). For v_p we use a grid of $N_o^{v_p} = 30$ values or classes ranging from $-1 + (1/30)$ to $1 - (1/30)$ of v_a.

In total 57,600 different meteorological cases are considered. For every case we generate 20 IQ time-series data with $N_i = 28$ samples each [25]. Every IQ time series is windowed by a Blackman window to reduce clutter sidelobes. The PSD is estimated using Welch periodogram [26] and normalised by their maximum value. The training data set (TDS) contains 1,152,000 different PSD realisations.

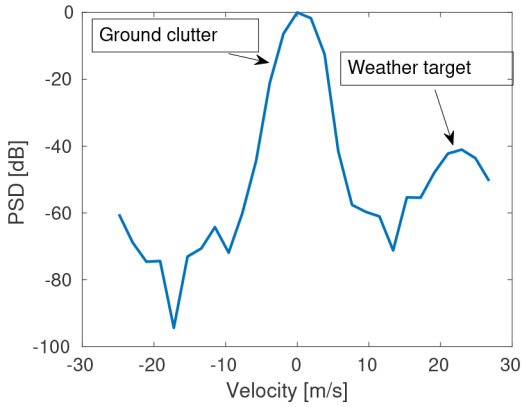

Fig. 2 *Weather radar signal PSD example*

For the validation data set (VDS) the CSR and SNR grids are divided into 20 uniformly spaced values from 0 to 50 dB (unlike the 15 of the TDS), while the σ_p and v_p grids remain the same as in the TDS. This change is made in order to have a VDS with situations not present in the TDS. With these parameter grids, a total of 100,800 different meteorological cases are selected. For every case we generate four IQ time-series data with $N_i = 28$ samples each. Every IQ time series is processed as before. The VDS contains 403,200 different PSD realisations.

To train the DNN for usage with real data, the synthetic training data is generated considering the true value of the radar asymptotic clutter width, σ_c. This parameter refers to the clutter width observed with an infinite-length data record and is a consequence of the decorrelation between azimuth pulses as the antenna beam moves [3]. The σ_c value can be calculated from the radar antenna characteristics, like the one-way azimuth half-power width, the wavelength (λ) and the rotation speed. In order to match the RMA-12 specifications, we set $\sigma_c = 0.13$ m/s.

Fig. 2 shows a typical realisation of a normalised weather radar signal PSD, composed of a weather target, ground clutter, and noise. The simulation parameters are $v_p = 0.8v_a = 21.42$ m/s, where $v_a = 26.78$ m/s, $\sigma_p = 2$ m/s and SNR = 20 dB. From Fig. 2 we can infer that the weather target is around the simulated velocity, but it is more difficult to state the spectral width from simple inspection, yet the DNN handles this estimation correctly.

3 NN architecture selection

In this section, we study different architectures of fully connected NNs in order to select the one that achieves better performance after training. We consider NN with 1–3 hidden layers and the different number of neurons. Larger NNs are not considered as they overfit the model. Table 1 shows the number of neurons in each layer for the 15 proposed architectures.

We construct the synthetic TDS to be balanced, i.e. there is the same number of training PSDs for each v_p and σ_p class. We train the 15 architectures by means of Tensorflow library, version 2.2.0 and Keras API [27, 28]. Every hidden layer uses a Relu [29] activation function and the outputs layers use a softmax function [20], which gives the probability of each output neuron in the v_p and σ_p branches. Categorical cross-entropy is used as the loss function in both branches, since this is a classification problem with more than two classes. Also, we choose Adam [30] for the gradient-based optimisation. We set the learning rate, l_r, the batch size, b_s, and epoch number, equal to 5×10^{-4}, 1024 and 100, respectively. We train the network in a PC with Intel core i3 8100 (3.6 GHz) processor, 16 GB of RAM and a Nvidia 1050 Ti GPU, 4

Table 1 Proposed neural networks architectures

Architecture	Velocity first hidden layer	Velocity second hidden layer	Velocity third hidden layer	Spectral width first hidden layer	Spectral width second hidden layer	Spectral width third hidden layer
1	50	—	—	40	—	—
2	100	—	—	80	—	—
3	200	—	—	160	—	—
4	400	—	—	320	—	—
5	800	—	—	720	—	—
6	50	100	—	40	80	—
7	100	200	—	80	160	—
8	200	400	—	160	320	—
9	400	800	—	320	640	—
10	800	1600	—	640	1280	—
11	50	100	150	40	80	120
12	100	200	300	80	160	240
13	200	400	600	160	320	480
14	400	800	1200	320	640	960
15	800	1600	2400	640	1280	1920

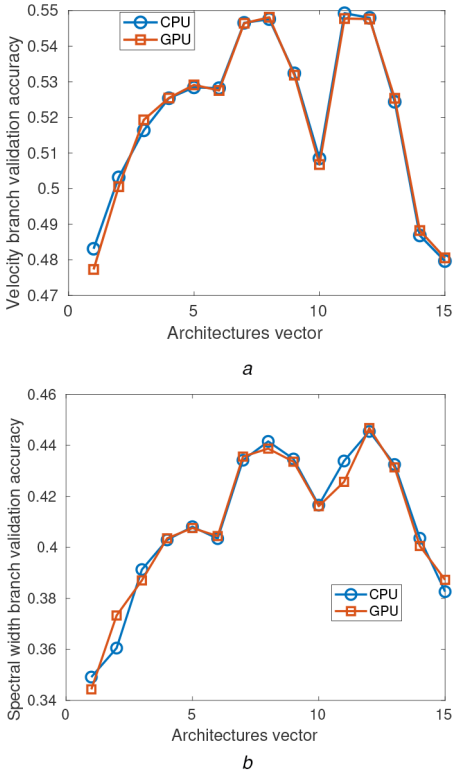

Fig. 3 *Validation accuracy after training epoch number 100*
(a) Velocity branch, ***(b)*** Spectral width branch

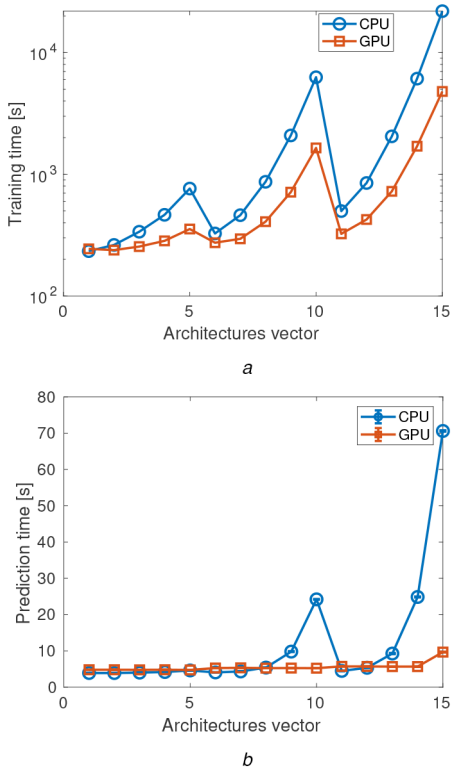

Fig. 4 *Runtime for different architectures, for both CPU and GPU*
(a) Training, ***(b)*** Prediction using validation data. The error bars are very small

GB of RAM. For every architecture we trained the NN using the CPU and GPU to compare training and prediction times.

Figs. 3*a* and *b* show the accuracy for all architectures using the VDS after training epoch 100 for v_p and σ_p branches, respectively. For both branches, the architecture 12 is the one with the highest validation accuracy. Fig. 4*a* shows the training time for all architectures. Except for the architecture number one, in all cases, the training time using the GPU is slightly less than using the CPU,

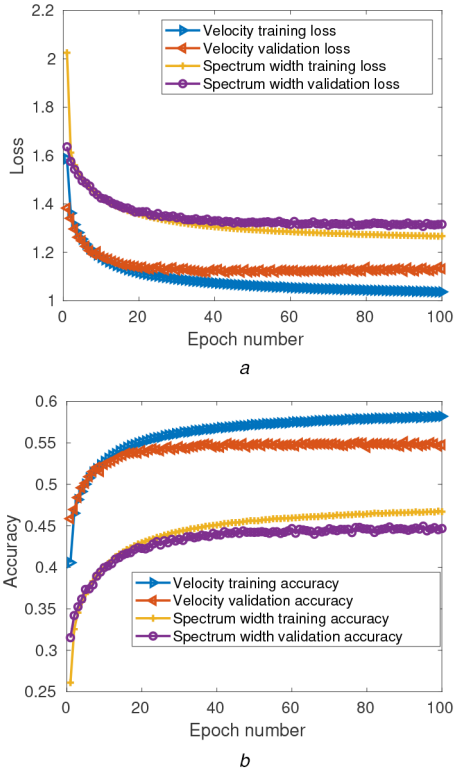

Fig. 5 *Training and validation performance for architecture 12*
(a) Training and validation loss, ***(b)*** Training and validation accuracy

not representing an advantage. Fig. 4*b* shows the mean prediction time for all architectures using the whole VDS (403,200 PSDs). For this experiment, the prediction time is measured five times for each architecture and processing unit. The standard deviation (std) of these five estimates is shown using error bars. It can be seen that these error bars are almost imperceptible, showing the precision of the estimated time values. Note that prediction time for architecture 12 for both CPU and GPU are very similar. From Figs. 3 and 4 we conclude that the architecture 12 is a good candidate to predict precipitation velocity and spectral width for the case of PSD with length, $N_\mathrm{i} = 28$.

In case that the radar acquisition strategy results in PSD with different number of samples, N_i, we propose to use a DNN with a proportional number of neurons in each hidden layer, taking $N_\mathrm{i} = 28$ and architecture 12 as a reference. The number of output neurons in each branch $N_\mathrm{o}^{v_p}$ and $N_\mathrm{o}^{\sigma_p}$, can be selected depending on the required velocity and spectral width resolution. Also, the range for σ_p can be selected depending on the different types of weather phenomena expected and the particular needs.

3.1 Training and validation performance

We further analyse the performance of the selected DNN architecture. Fig. 5*a* shows the training and validation loss, while Fig. 5*b* shows the training and validation accuracy, for this architecture, as a function of the epoch number. The accuracy is the fraction of predictions that the classification is correct. The training and validation loss decreases, and the accuracy increases with the epoch number, meaning that the DNN is learning to predict the correct classes.

From Fig. 5*b* the velocity branch accuracy is around 0.55 and the spectral width accuracy is around 0.44, which seems to be poor if good estimators are required. However, this is due to the uniform weighting function used which accounts for one error whether the estimated value is next to the true value, or if it is far from it.

To better explore the validation results, we analyse the distance between the classes. The difference between the index of the true class is subtracted from the index of the predicted class, resulting in the *velocity class error* for v_p and the *spectral width class error*

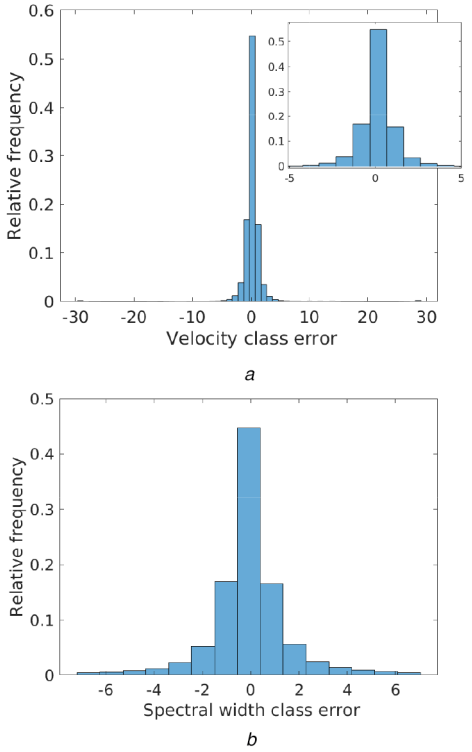

Fig. 6 *Histogram of class errors*
(a) Class error for velocity branch. The sub-figure corresponds to a zoom of the region ±5, ***(b)*** Class error for spectral width branch

Table 2 General simulation input parameters

Parameter	Value
F_c, GHz (carrier frequency)	5.6
σ_c, m/s	0.13
v_p, m/s	$(-v_a, v_a)$
N_i, samples	28
T_s (pulse repetition time), ms	0.5
v_a, m/s	26.78

for σ_p. These class distances are used to quantify the severity of the misclassification.

Figs. 6*a* and *b* show histograms of the class errors for velocity and spectral width branches, respectively. For both branches, >90% of the class errors are within two classes of the true class, showing that when errors occur the estimated value is still close to the true value in most cases.

4 Results using synthetic data

In this section, we study the performance of the trained DNN. It is worth noting that the TDS contains PDSs representing situations with and without the presence of ground clutter. Herein, we separately analyse the performance for each of these situations and compare this performance against well-known and widely used estimators: for the situation without clutter, presented in Section 4.1, we compare the DNN performance against the classic PPP; while for the situation with clutter, presented in Section 4.2, we compare the DNN performance against GMAP.

In all the cases, we use the sample bias and the sample std of the estimates as metrics; and we analyse the performance for different parameter values. Since there are three (or four) analysis dimensions: v_p, σ_p and SNR (and CSR when applicable), it is difficult to present an analysis for simultaneous variations of these parameters. For this reason, we present results for different representative scenarios where we fix one (or two) parameters, vary v_p into a fine grid, and analyse three cases of the remaining parameter.

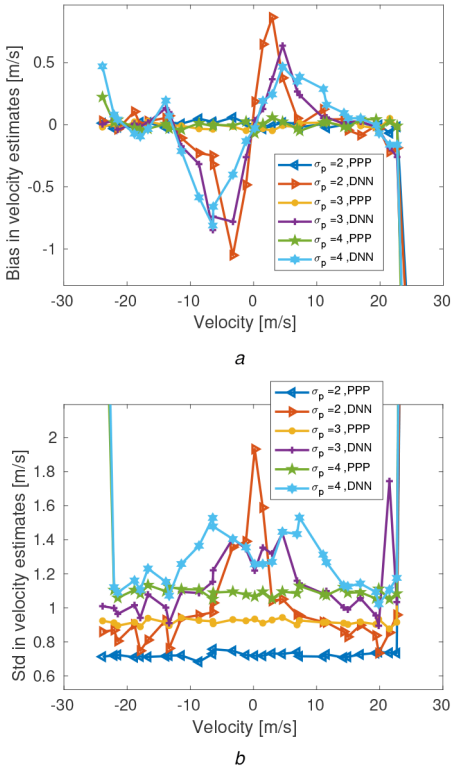

Fig. 7 *Velocity error analysis of PPP and DNN for different σ_p and SNR = 20 dB*
(a) Bias, ***(b)*** Std

The simulation parameters are presented in Table 2. All test data are generated following [25]. In order to be fair in the comparison between DNN against PPP or against GMAP, we select the true v_p values randomly, with uniform distribution in each of the velocity branch output classes. This randomisation is done again for each case analysed – i.e. a different set of true v_p values is used at each experiment.

4.1 DNN performance without ground clutter

We compare the DNN performance against PPP in the situation of no clutter present. In this situation there are three parameters to vary: v_p, σ_p and SNR. We consider two representative scenarios: for the first one, in Section 4.1.1, we fix SNR and analyse three representative values of σ_p; while on the second one, in Section 4.1.2, we fix σ_p and analyse three representative values of SNR. In each scenario, we set different values of the simulated mean Doppler velocity, v_p, ranging from $-v_a$ to v_a. For each configuration point – i.e. at each value of σ_p, SNR and v_p simulated – we conduct 1000 Monte Carlo runs.

4.1.1 Different spectral widths: We test three different values of σ_p: 2, 3 and 4 m/s; fixing SNR to 20 dB. Figs. 7*a* and *b* show the bias, $b_{\hat{v}_p}$, and the std, $\sigma_{\hat{v}_p}$, in the velocity estimation, as a function of v_p. In the same way, Figs. 8*a* and *b* show the bias, $b_{\hat{\sigma}_p}$, and the std, $\sigma_{\hat{\sigma}_p}$, in the spectral width estimation, as a function of v_p. The different curves of each plot correspond to the results for different values of σ_p using both PPP and DNN.

From Figs. 7*a* and *b* we can see that PPP performs better than DNN. The larger DNN bias around zero velocity is mostly due to using a DNN that is also trained for a signal with clutter. Indeed, almost 94% (15 out of 16) sets in the TDS correspond to cases with ground clutter. This effect also manifests in the velocity DNN bias and std around zero Doppler velocity which increases for small σ_p, since narrow signals around zero velocity are prone to be classified as clutter. From Figs. 8*a* and *b* we can observe that DNN outperforms PPP over the entire mean Doppler-velocity range, indicating that the DNN can better distinguish the different

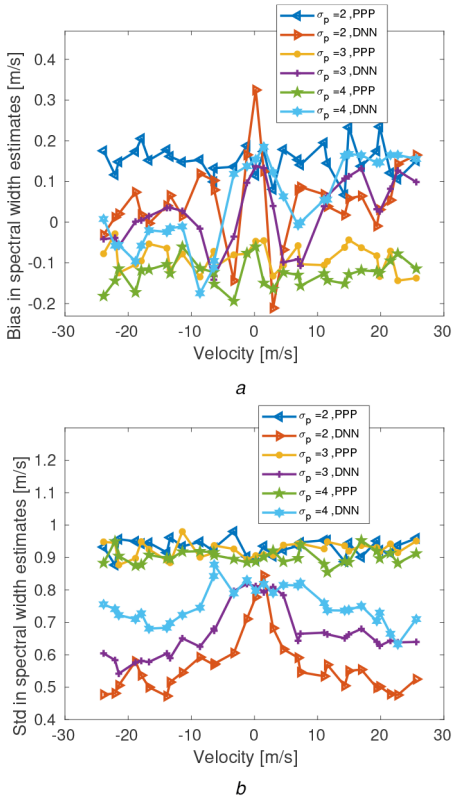

Fig. 8 *Spectral width error analysis of PPP and DNN for different σ_p and SNR = 20 dB*
(a) Bias, ***(b)*** Std

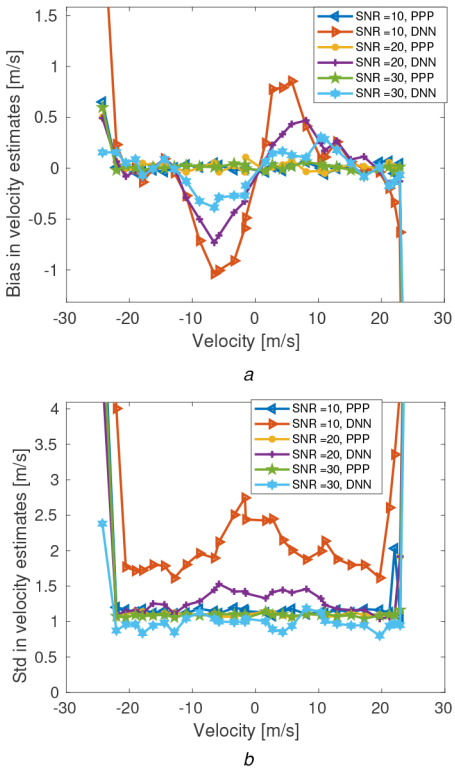

Fig. 9 *Velocity error analysis of PPP and DNN for different SNR and $\sigma_p = 4\,m/s$*
(a) Bias, ***(b)*** Std

simulated σ_p. Also, as in the velocity error analysis, the estimator worsens around zero velocity. It is important to remark that the σ_p values in use for testing (2, 3 and 4 m/s) do not exactly match any of the output class values of the spectral width branch.

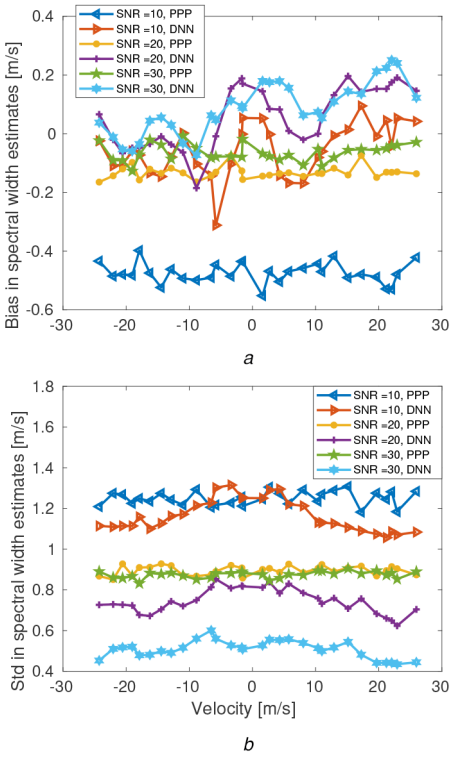

Fig. 10 *Spectral width error analysis of PPP and DNN for different SNR and $\sigma_p = 4\,m/s$*
(a) Bias, ***(b)*** Std

4.1.2 Different SNRs: We consider three different values of SNRs: 10, 20 and 30 dB; setting σ_p to 4 m/s. Figs. 9*a*, 9*b*, 10*a* and 10*b* show $b_{\hat{v}_p}$, $\sigma_{\hat{v}_p}$, $b_{\hat{\sigma}_p}$, and $\sigma_{\hat{\sigma}_p}$, as a function of v_p, respectively. The different curves correspond to different values of SNR using both PPP and DNN.

From Fig. 9*a* we can see that PPP shows almost no velocity bias, while the DNN shows bias for velocities around zero, for the same reasons as before. From Fig. 9*b* we can see that $\sigma_{\hat{v}_p}$ is approximately twice for DNN than for PPP for the particular case of SNR = 10 dB, while they are comparable for the other SNR values. From Figs. 10*a* and *b* we can observe that DNN has better performance than PPP over the entire velocity range. For SNR = 10 dB PPP shows a constant bias different from zero. This is because the noise floor, which is obtained using [31], is underestimated under these conditions.

Overall PPP performs better than DNN in terms of velocity estimation, and DNN slightly outperforms PPP for spectral width estimation. Also, DNN shows better performance for high and low SNR, whereas PPP gives better estimates for medium SNR. Note that PPP is a method that only works in the absence of clutter – otherwise gives very biased estimates, while the proposed DNN also works with clutter contaminated signals. This generality penalises performance in the no clutter case.

4.2 DNN performance when ground clutter is present

We compare the DNN performance against GMAP in the situation of clutter present. In this situation there are four parameters to vary: v_p, σ_p, CSR and SNR. We choose three representative scenarios to show: for the first one, in Section 4.2.1, we fix SNR and CSR, and analyse three values of σ_p; on the second one, in Section 4.2.2, we fix σ_p and SNR and analyse three values of CSR; while on the third one, in Section 4.2.3, we fix σ_p and CSR and analyse three values of SNR. For each configuration point – i.e. at each value of σ_p, CSR, SNR and v_p simulated – we conduct 1000 Monte Carlo runs.

4.2.1 Different spectral widths: We test three different values of σ_p: 2, 3 and 4 m/s; setting CSR to 40 dB and SNR to 20 dB.

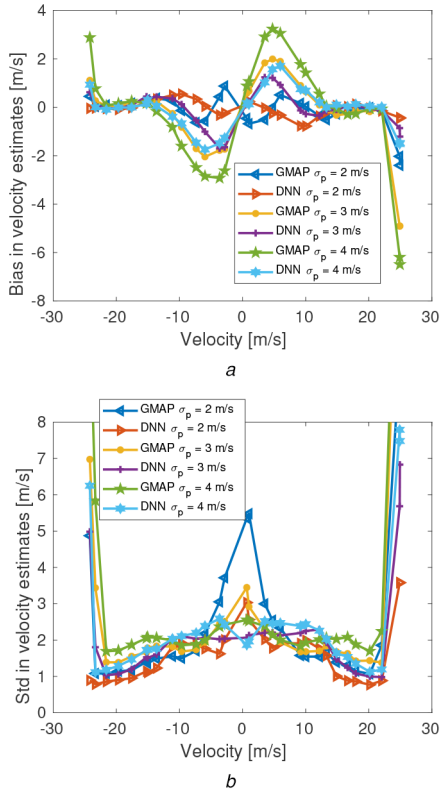

Fig. 11 *Velocity error analysis of GMAP and DNN for different σ_p, CSR = 40 dB and SNR = 20 dB*
(a) Bias, ***(b)*** Std

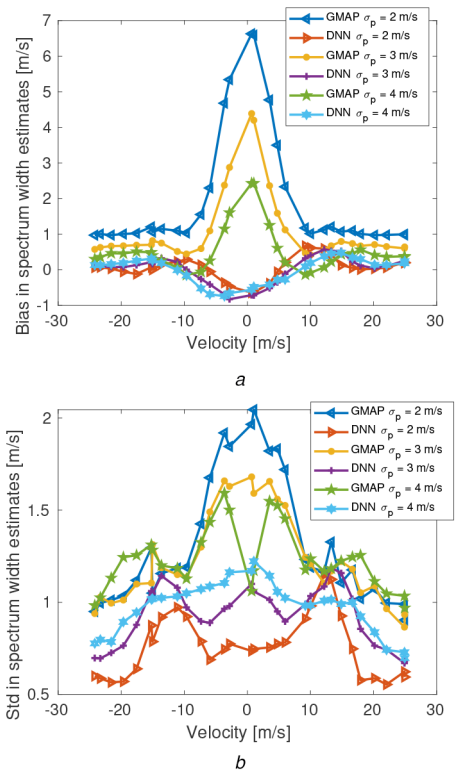

Fig. 12 *Spectral width error analysis of GMAP and DNN for different σ_p, CSR = 40 dB and SNR = 20 dB*
(a) Bias, ***(b)*** Std

Figs. 11*a* and *b* show the bias, $b_{\hat{v}_p}$, and the std, $\sigma_{\hat{v}_p}$, in the velocity estimation, as a function of v_p. In the same way, Figs. 12*a* and *b* show the bias, $b_{\hat{\sigma}_p}$, and the std, $\sigma_{\hat{\sigma}_p}$, in the spectral width estimation, as a function of v_p. The different curves of each plot correspond to the results for different values of σ_p using both GMAP and DNN.

Note that in both cases the GMAP estimates get worse when v_p is around zero. In these situations the weather target and clutter spectra overlap. Thus clutter is removed [10], significant parts of the weather target spectrum are also eliminated and cannot be properly reconstructed from the remaining data. For large v_p values, the metrics are similar since they only depend on the performance of the underlying moments estimator – PPP is already compared against DNN in the previous subsection.

Fig. 11*a* shows the bias of the mean velocity estimates, $b_{\hat{v}_p}$. GMAP bias estimates increase as σ_p increases because the centroid of a wider spectrum is more affected after truncation. The case where v_p is close to zero is an exception since the truncation of the central portion of the weather spectrum keeps symmetry and reduces the bias. The opposite is observed in the std estimation (Fig. 11*b*). When the spectrum is wider, the eliminated samples affect the different realisations quite similarly – although in a systematic way as observed in the bias.

For the case of spectral width estimate errors Fig. 12 shows that GMAP estimates present larger biases and std's when the spectrum is narrow. This is a consequence of removing clutter contaminated samples which represent a relevant portion of the weather spectrum.

Note that the proposed DNN generally outperforms GMAP over the entire v_p range. The velocity bias of DNN, Fig. 11*a*, is less affected than GMAP around zero Doppler velocity. Note that the shape of the bias curves is very similar for both methods, suggesting that the estimation for low mean velocities in the presence of clutter is hard. For the case of the velocity std $\sigma_{\hat{v}_p}$, Fig. 11*b* suggests that around ± 8 m/s, the performance of DNN is slightly worse than GMAP, but GMAP bias for this velocity is much worst, concluding that DNN performs better for all v_p.

Analysing the spectral width metrics in Fig. 12, it can be seen that the DNN performance is almost always better than GMAP, in the entire v_p range. In the sections where the std of the GMAP estimates seem to be better, around v_p values of ± 10 m/s, but as before, GMAP bias values are worst than DNN for this velocity.

4.2.2 Different CSR: We consider three cases of increasing clutter power, represented by CSR values of 30, 40 and 50 dB, fixing the spectrum width σ_p to 4 m/s and SNR to 20 dB. Figs. 13*a*, 13*b*, 14*a* and 14*b* show $b_{\hat{v}_p}$, $\sigma_{\hat{v}_p}$, $b_{\hat{\sigma}_p}$, and $\sigma_{\hat{\sigma}_p}$, respectively, as a function of the velocity v_p. Analogous to the former analysis, the different curves correspond to different values of CSR using both GMAP and DNN.

Following the arguments of the discussion in the previous subsection, GMAP estimates tend to get worse when v_p is around zero. From Fig. 13*a* it results in counter intuitive that the bias $b_{\hat{v}_p}$ is greater for lower CSR. The reason is that for higher clutter power, GMAP filters a wider low-pass spectrum region, truncating the overlapped weather PSD in a more noticeable way when v_p is small. When the truncated region is wider, the resulting PSD is noise-like, and the resulting estimate $\hat{v}_p$ gets closer to zero. On the other hand, when the truncated region is narrow, the truncated PSD keeps more points of the original PSD, but the estimates are biased towards the remaining components. Fig. 13*b* shows that the std is greater when CSR is higher – i.e. DNN performance is better than GMAP for larger CSR – as expected. From Fig. 14, the bias and std of the GMAP spectral width estimates increase when the CSR is higher, due to the greater spectrum truncation.

As in the previous analysis, it can be seen that DNN outperforms GMAP, over the entire v_p range.

4.2.3 Different SNRs: We consider three different values of SNRs: 10, 20 and 30 dB; setting σ_p to 4 m/s and CSR to 40 dB. Figs. 15*a*, 15*b*, 16*a* and 16*b* show $b_{\hat{v}_p}$, $\sigma_{\hat{v}_p}$, $b_{\hat{\sigma}_p}$, and $\sigma_{\hat{\sigma}_p}$, as a function of v_p, respectively. As before, the different curves correspond to different values of SNR using both GMAP and DNN.

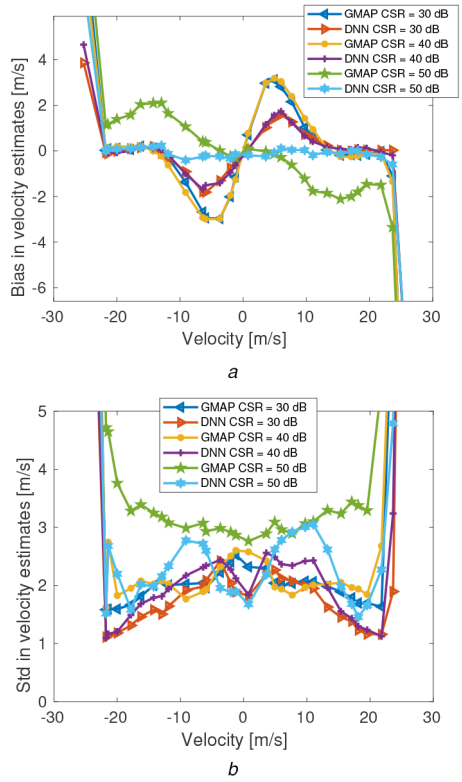

Fig. 13 *Velocity error analysis of GMAP and DNN for different CSR, SNR = 20 dB and* $\sigma_p = 4\,m/s$

(a) Bias, ***(b)*** Std

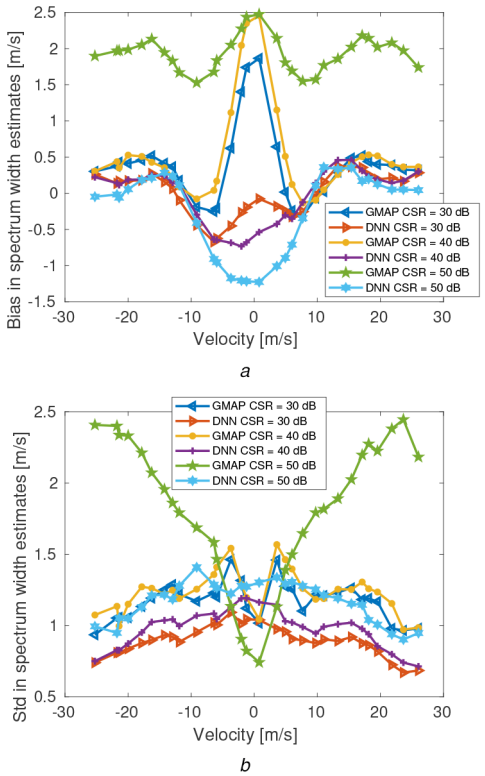

Fig. 14 *Spectral width error analysis of GMAP and DNN for different CSR, SNR = 20 dB and* $\sigma_p = 4\,m/s$

(a) Bias, ***(b)*** Std

We, again, observe the same behaviour discussed in Sections 4.2.1 and 4.2.2 in GMAP estimates for v_p around zero. From Fig. 15*a* we also see the counter intuitive behaviour of $b_{\hat{v}_p}$, becoming larger for higher SNR. The explanation is akin to the one given in the previous subsection. When the SNR is lower, the

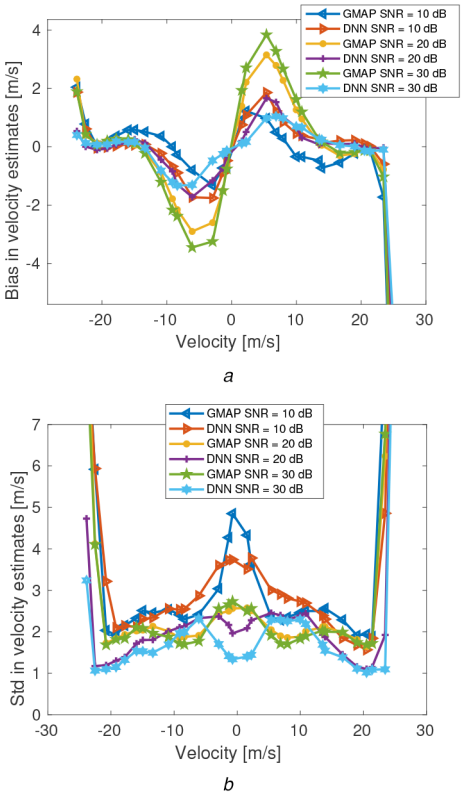

Fig. 15 *Velocity error analysis of GMAP and DNN for different SNR, CSR = 40 dB and* $\sigma_p = 4\,m/s$

(a) Bias, ***(b)*** Std

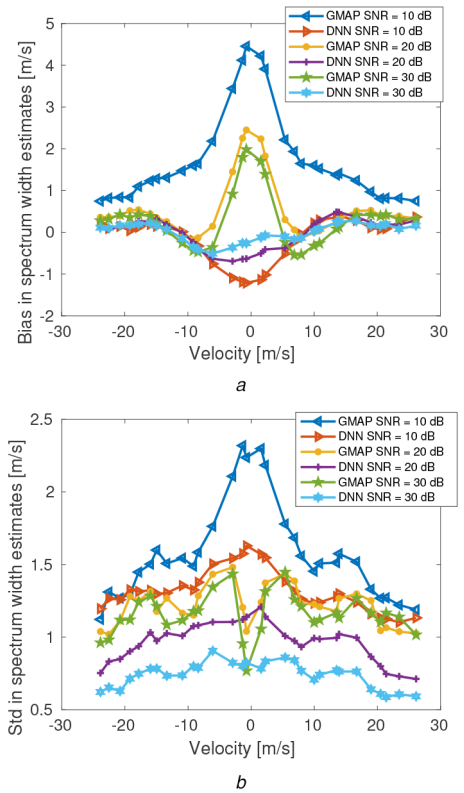

Fig. 16 *Spectral width error analysis of GMAP and DNN for different SNR, CSR = 40 dB and* $\sigma_p = 4\,m/s$

(a) Bias, ***(b)*** Std

resulting PSD after truncation is noise-like, then $\hat{v}_p$ is close to zero. On the other hand, when the SNR is higher, the truncated PSD is more distinguishable from noise, but the estimation is biased towards the remaining spectral components. Note from Fig. 15 that

the behaviour of $b_{\hat{v}_p}$ and $\sigma_{\hat{v}_p}$ is analogous to the previous section and once again DNN outperforms GMAP over the entire v_p range.

4.3 Runtime analysis of DNN

In this subsection, we present a runtime analysis of DNN and GMAP, under a well-defined simulation scenario. To perform this analysis it is important to note that the DNN and GMAP algorithms use different approaches.

Both algorithms include a first step of estimating the PSD from time-series IQ data using the FFT. The other steps of the DNN algorithm involve matrix multiplications, additions and evaluating activation functions. On the other hand, the next steps of the GMAP algorithm are iterative and their stopping conditions depend on the results of the iteration. So, the number of operations in this approach cannot be predicted. For this reason, we choose average runtime as the metric to analyse complexity. The main disadvantage of this approach lies in its dependence on the particular implementation we make of the algorithms.

The general input simulation parameters are listed in Table 2, where we use CSR = 40 dB, SNR = 20 dB and $\sigma_p = 3$ m/s. One important thing to highlight is that the implementation of GMAP in CPU requires processing one PSD at a time, but for the DNN methodology, we can arrange all PSDs in a unique matrix of size $H \times N_i$ and feed the DNN, where H is the number of all PSDs. All simulations are performed on the same platform using Python in a PC with Intel core i3 8100 (3.6 GHz) processor and 16 GB of RAM.

Fig. 17 shows average execution time for both algorithms in a logarithmic scale. We use 30 different v_p values, and for each v_p, 1000 different IQ time series are generated. In the case of GMAP, we call it for every v_p 1000 times using two *for* loops, one for v_p and the other for the 1000 Monte Carlo realisations. In the DNN case, all PSDs are arranged in a matrix of size $30{,}000 \times N_i$ to feed de DNN, no *for* loop is required, and the total execution time is divided by 30 to compare with GMAP. From Fig. 17 it is evident that GMAP needs more time to converge for velocities around zero, as expected due to the clutter and precipitation PSD overlapping. Also, the DNN execution time is more than an order of magnitude smaller than that of GMAP, making the proposed algorithm attractive for real-time implementations in operational weather radars.

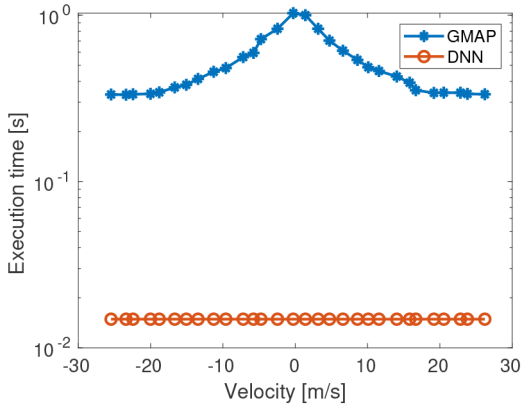

Fig. 17 *Execution time comparison between DNN and GMAP*

5 Results using real data

In this section, we present the results of the proposed algorithm performance using real weather radar data. We directly apply the RMA-12 data to the DNN trained with synthetic data. A better approach to deal with the Doppler processing estimation problem is to retrain the last hidden layer using real weather PSDs, a technique called transfer learning [24]. In practice, the use of this technique is not possible due to the lack of labelled data, i.e. exact knowledge of the weather parameters.

The measurements used were collected using the RMA-12 Argentinian weather radar, located in San Carlos de Bariloche city. RMA-12 is a C-band polarimetric radar, designed and developed by INVAP S.E. Data was recorded on 1 June 2015, under heavy rain weather conditions.

We show the results that correspond to a complete sweep of the horizontal polarisation (HH) at an elevation angle of 3.0°, ensuring that precipitation and clutter components are present. The sweep uses a uniform PRT, T, of 2 ms, resulting in an unambiguous velocity, v_a, of 6.66 m/s.

Figs. 18*a* and *b* show the plan position indicator (PPI) displays of the reflectivity and the Doppler velocity estimates, respectively, obtained by means of PPP without ground clutter filtering.

The regions of highest reflectivity observed in Fig. 18*a* correspond to ground clutter. For example, at a range of 25 km and an azimuth of 185° we identify the Cerro Catedral (2388 mamsl) mountain and at a range of 60 km and an azimuth of 180° we observe the Cerro Tronador (3554 mamsl) mountain. There is also a group of less relevant mountains at 25 km range and 100° azimuth, whose average peak is around 2000 m above the mean sea level. In Fig. 18*b* these high reflectivity regions present an almost zero mean velocity, as expected. Also, note that the velocity PPI presents stripes of Doppler velocities of 5.7–6.6 m/s next to stripes of Doppler velocities from −6.6 to −5.7 m/s, with abrupt transitions between stripes. This behaviour suggests the existence of aliasing as a consequence of the low unambiguous velocity value.

We apply the GMAP algorithm and the DNN algorithm to this weather data set, in order to filter the ground clutter and to estimate the spectral moments. Figs. 19*a* and *c* show the PPI displays of v_p

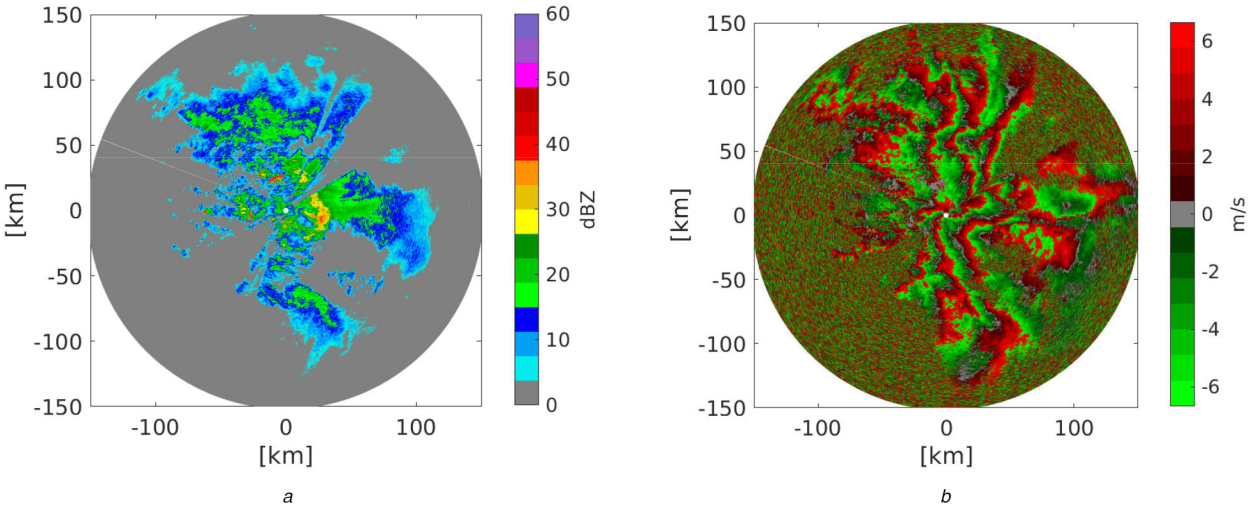

Fig. 18 *Real weather radar data*
(a) Reflectivity, ***(b)*** Velocity estimate using PPP

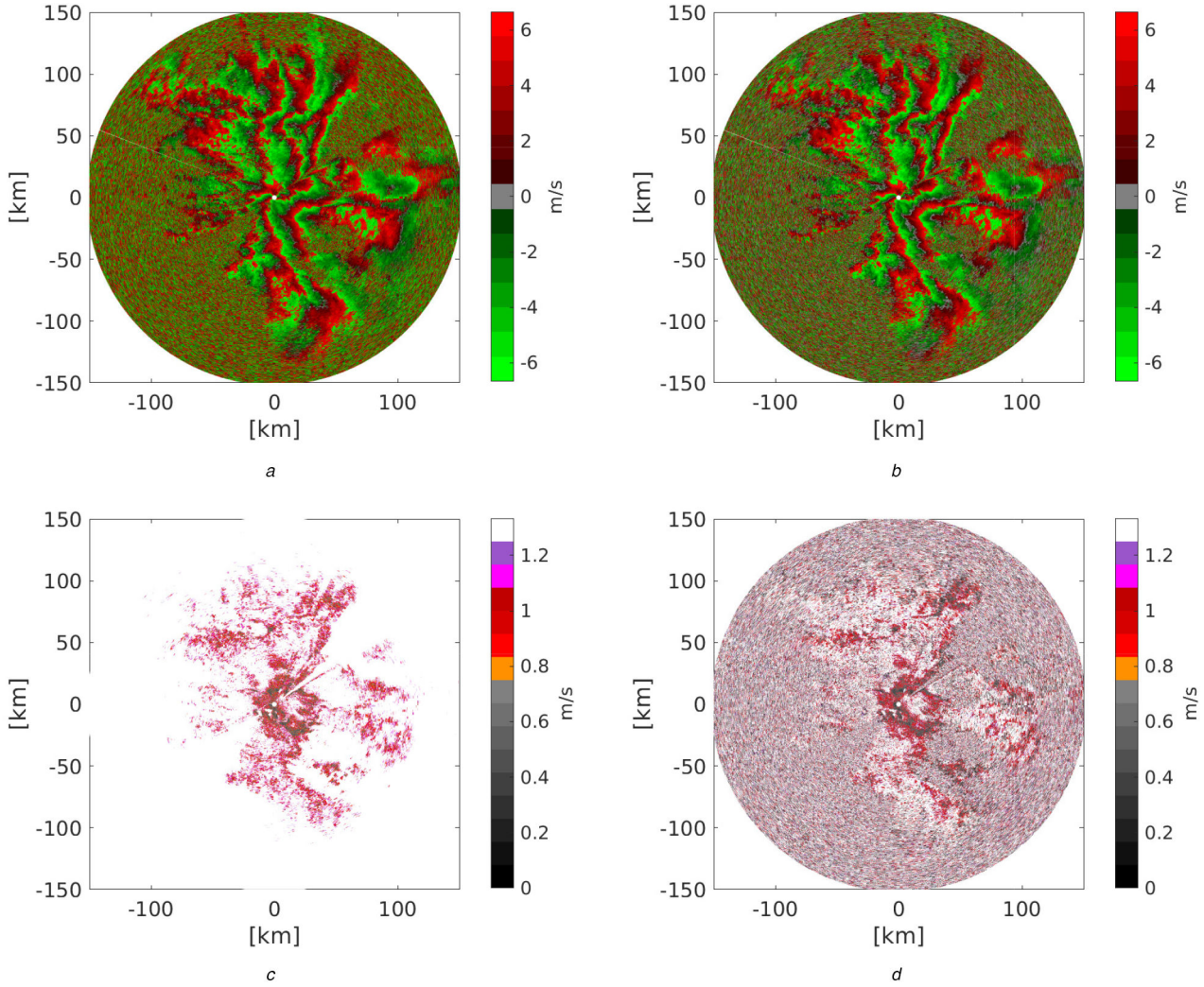

Fig. 19 *PPI plots show the performance of GMAP and DNN on RMA-12 measurements*
(a) Velocity using GMAP, ***(b)*** Velocity using DNN, ***(c)*** Spectral width using GMAP, ***(d)*** Spectral width using DNN

and σ_p estimates, respectively, after GMAP filtering. Analogously, Figs. 19*b* and *d* show the PPI displays of v_p and σ_p estimates, respectively, obtained by the DNN.

Since real data is being used, we lack knowledge of the true parameter values. Therefore, a rigorous quantitative analysis of the results cannot be performed. Only a qualitative comparison of the results obtained with the different estimators can be made.

From a qualitative analysis of Figs. 19*a* and *b* we observe in both, the GMAP velocity and the DNN velocity, that the identified ground clutter is suppressed and the weather target velocity is obtained. In addition, the zones where there is no ground clutter, the estimated Doppler velocity is similar for PPP, GMAP and DNN methods. Naturally, the aliasing effect also appears when we apply DNN or GMAP algorithms, since it is already present in the PSD. In the same way, from Figs. 19*c* and *d*, the spectral width estimates of GMAP and the DNN are similar. It is important to note that the DNN always estimates spectral width values in the range $(0.04 - 0.2)v_a$.

6 Conclusion

We present a novel method for meteorological target velocity and spectral width estimation in weather radar that operates both when ground clutter is present or not. It uses a DNN with two branches, one for the velocity and the other for the spectral width estimations. We study different NN architectures with fully connected layers using the accuracy over the VDS as a metric. The training and VDS consist of PSD estimates computed from synthetic weather radar data, covering a wide range of possible scenarios.

We divide the parameter space for the generation of training data in a grid that accounts for the most common parameter values from a trade-off between estimation accuracy and computational load of the training stage. If the grid becomes denser, the estimation error decreases. Although the ideal situation consists of a continuous parameter space that accounts for every possible meteorological situation, this is unfeasible in practice.

We define typical cases for testing the algorithm, evaluating the bias and the std of the velocity and of the spectral width estimates by means of Monte Carlo simulations using synthetic data in different signal composition situations. We compare the DNN performance against PPP and GMAP performance in these situations. The proposed method outperforms GMAP over the entire mean Doppler-velocity range. In the case of signals without clutter, we observe that the DNN obtains biased estimates around zero velocity, but the deviation is small and does not invalidate the result. This degradation in performance is due to the fact that almost 94% of the training data has ground clutter present. In situations where only weather precipitation and noise are present, a better solution to the estimation problem could be to use a DNN trained without ground clutter PSD, only having precipitation and noise PSD, but this approach requires a previous clutter detection stage.

We make a runtime comparison between DNN and a serial implementation of GMAP in a well-defined simulation scenario. We show that DNN is fast given its ability to process all data at once, making it a good candidate for real-time operation.

We also compared the DNN and GMAP performance using a uniform-PRT sweep of real weather data. Although the formulation of the proposed method is general, we tested it using data acquired with RMA-12, a C-band weather radar, located in San Carlos de

Bariloche, Argentina. We identify ground clutter regions based on the topographic knowledge of the radar placement region. The analysis shows that the trained DNN can estimate the weather target velocity and the spectral width, with and without the presence of ground clutter. The DNN presents a good performance, similar to the commonly used ground clutter filter GMAP and the PPP spectral moments estimation algorithm. It is important to note that the success of the proposed DNN algorithm is due to the fact that the statistical model used to generate the synthetic data in the training stage fits well to the weather radar measurements. As future work, it might be of interest to investigate the estimation of the weather signal power. With the current formulation, this is not possible as a consequence of the PSD normalisation. Another possible next step is to extend these ideas to estimate the spectral moments of data sequences that operate with staggered-PRT.

Furthermore, the proposed technique could be adapted to contemplate other types of clutter such as wind turbine clutter (WTC) [32, 33], sea clutter [34, 35] or biological clutter [36]. Also, since about 25% of spectral measurements deviate from the Gaussian model [8], the proposed DNN can be trained with properly modelled non-Gaussian spectra to be able to deal with these situations.

7 Acknowledgments

This work was supported by the Universidad Nacional de Cuyo C038 4142/19, the Fondo para la Investigación Científica y Tecnológica PICT-2018-01277, the Universidad Nacional de Río Negro PIDTT 40-B-693, the Consejo Nacional de Investigaciones Científicas y Técnicas and the Comisión Nacional de Energía Atómica. RMA data was provided by Secretaría de Infraestructura y Políticas Hídricas, Ministerio del Interior, Obras Públicas y Vivienda of the Argentinean National Government framed within the SINARAME Project. The National System of Weather Radars (Sistema Nacional de Radares Meteorológicos, SINARAME) project is an Argentinean effort to expand the radar network over the whole country.

8 References

[1] Meischner, P.: '*Weather radar: principles and advanced applications*' (Springer, Berlin, Heidelberg, 2013)

[2] Andrews, C.: 'The future of weather forecasting [communications met office supercomputer]', *Eng. Technol.*, 2015, **2**, (10), pp. 65–67

[3] Doviak, R.J., Zrnic, D.S.: '*Doppler radar and weather observations*' (Courier Corporation, USA, 2014)

[4] Fukao, S., Hamazu, K., Doviak, R.: '*Radar for meteorological and atmospheric observations*' (Springer Japan, Japan, 2013)

[5] Ryzhkov, A.V., Zrnic, D.S.: *'Radar polarimetry for weather observations*', Springer Atmospheric Sciences (Springer International Publishing, Switzerland, 2019)

[6] Mahapatra, P.R., Zrnić, D.S.: 'Practical algorithms for mean velocity estimation in pulse Doppler weather radars using a small number of samples', *IEEE Trans. Geosci. Remote Sens.*, 1983, **GE-21**, (4), pp. 491–501

[7] Zrnić, D.S.: 'Spectral moment estimates from correlated pulse pairs', *IEEE Trans. Aerosp. Electron. Syst.*, 1977, **AES-13**, (4), pp. 344–354

[8] Janssen, L., Van Der Spek, G.A.: 'The shape of Doppler spectra from precipitation', *IEEE Trans. Aerosp. Electron. Syst.*, 1985, **AES-21**, (2), pp. 208–219

[9] Groginsky, H.L., Glover, K.M.: 'Weather radar canceller design'. 19th Conf. on Radar Meteorology, Miami Beach, FL, USA, 1980, pp. 192–198

[10] Siggia, A., Passarelli, R.: 'Gaussian model adaptive processing (GMAP) for improved ground clutter cancellation and moment calculation'. Proc. ERAD, Visby, Sweden, 2004, pp. 421–424

[11] Nguyen, C.M., Chandrasekar, V.: 'Gaussian model adaptive processing in time domain (GMAP-TD) for weather radars', *J. Atmos. Oceanic Technol.*, 2013, **30**, pp. 2571–2584

[12] Torres, S.M., Warde, D.A.: 'Ground clutter mitigation for weather radars using the autocorrelation spectral density', *J. Atmos. Oceanic Technol.*, 2014, **31**, (10), pp. 2049–2066

[13] Hubbert, J.C., Dixon, M., Ellis, S.M.: 'Weather radar ground clutter. Part II: real-time identification and filtering', *J. Atmos. Oceanic Technol.*, 2009, **26**, pp. 1181–1197

[14] Ice, R.L., Rhoton, R.D., Krause, J.C.*, et al.*: 'Automatic clutter mitigation in the WSR-88D, design, evaluation, and implementation'. Proc. 34th Radar Meteorology, Williamsburg, VA, USA, 2009, pp. P5.3 1–12

[15] Warde, D.A., Torres, S.M.: 'The autocorrelation spectral density for Doppler-weather-radar signal analysis', *IEEE Trans. Geosci. Remote Sens.*, 2014, **52**, (1), pp. 508–518

[16] Kon, S., Tanaka, T., Mizutani, H.*, et al.*: 'A machine learning based approach to weather parameter estimation in Doppler weather radar'. 2011 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Prague, Czech Republic, 2011, pp. 2152–2155

[17] Zhang, L., Tan, J., Han, D.*, et al.*: 'From machine learning to deep learning: progress in machine intelligence for rational drug discovery', *Drug Discov. Today*, 2017, **22**, (11), pp. 1680–1685

[18] Kononenko, I.: 'Machine learning for medical diagnosis: history, state of the art and perspective', *Artif. Intell. Med.*, 2001, **23**, (1), pp. 89–109

[19] Gyorfi, L., Ottucsak, G., Walk, H.: *'Machine learning for financial engineering'*, Advances in Computer Science and Engineering: Texts (Imperial College Press, England, 2011)

[20] Goodfellow, I., Bengio, Y., Courville, A.: '*Deep learning*' (MIT Press, USA, 2016)

[21] Islam, T., Rico-Ramirez, M.A., Han, D.*, et al.*: 'Artificial intelligence techniques for clutter identification with polarimetric radar signatures', *Atmos. Res.*, 2012, **109–110**, pp. 95–113

[22] Wang, H., Ran, Y., Deng, Y.*, et al.*: 'Study on deep-learning-based identification of hydrometeors observed by dual polarization Doppler weather radars', *J. Wirel. Commun. Netw.*, 2017, **173**, pp. 1–9

[23] Li, H., Ren, J., Han, J.*, et al.*: 'Ground clutter suppression method based on FNN for dual-polarisation weather radar', *J. Eng.*, 2019, **2019**, (19), pp. 6043–6047

[24] Pan, S.J., Yang, Q.: 'A survey on transfer learning', *IEEE Trans. Knowl. Data Eng.*, 2010, **22**, (10), pp. 1345–1359

[25] Zrnic, D.S.: 'Simulation of weatherlike Doppler spectra and signals', *J. Appl. Meteorol.*, 1975, **14**, (4), pp. 619–620

[26] Welch, P.: 'The use of fast Fourier Transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms', *IEEE Trans. Audio Electroacoust.*, 1967, **15**, (2), pp. 70–73

[27] Abadi, M., Agarwal, A., Barham, P.: 'Tensorflow: large-scale machine learning on heterogeneous systems', 2015. Available at https://www.tensorflow.org/

[28] Chollet, F.: 'Keras'. GitHub, 2015. Available at https://github.com/fchollet/keras

[29] Nair, V., Hinton, G.E.: 'Rectified linear units improve restricted Boltzmann machines'. Proc. of the 27th Int. Conf. on Machine Learning (ICML-10), Haifa, Israel, 2010, pp. 807–814

[30] Kingma, D.P., Ba, J.: 'Adam: a method for stochastic optimization', arXiv preprint arXiv:14126980, 2014

[31] Hildebrand, P.H., Sekhon, R.: 'Objective determination of the noise level in Doppler spectra', *J. Appl. Meteorol.*, 1974, **13**, (7), pp. 808–811

[32] Uysal, F., Selesnick, I., Isom, B.M.: 'Mitigation of wind turbine clutter for weather radar by signal separation', *IEEE Trans. Geosci. Remote Sens.*, 2016, **54**, (5), pp. 2925–2934

[33] Hood, K., Torres, S., Palmer, R.: 'Automatic detection of wind turbine clutter for weather radars', *J. Atmos. Oceanic Technol.*, 2010, **27**, (11), pp. 1868–1880

[34] Dutta, A., Chandrasekar, V.: 'Detection, analysis and mitigation of sea clutter in polarimetric weather radar'. 2019 United States National Committee of URSI National Radio Science Meeting (USNC-URSI NRSM), Boulder, CO, USA, 2019, pp. 1–2

[35] Hailong, W., Shouyuan, D., Xu, W.*, et al.*: 'Sea clutter recognition based on dual-polarization weather radar'. 2019 Int. Conf. on Meteorology Observations (ICMO), Chengdu, China, 2019

[36] Radhakrishna, B., Fabry, F., Kilambi, A.: 'Fuzzy logic algorithms to identify birds, precipitation, and ground clutter in S-band radar data using polarimetric and nonpolarimetric variables', *J. Atmos. Oceanic Technol*, 2019, **36**, (12), pp. 2401–2414