A Convolution Neural Network Approach to Doppler Spectra Classification of 205 MHz Radar

Baazil P Thampy (✉ baazilpthampy@gmail.com)

Cochin University of Science and Technology <https://orcid.org/0000-0002-6011-2771>

Shailesh S

Cochin University of Science and Technology

Judy M.V.

Cochin University of Science and Technology

Ajil Kottayil

Cochin University of Science and Technology

Research Article

Keywords: Wind Profiler Radar, Radar Target Classification, Convolutional Neural Network

Posted Date: January 21st, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1191052/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

A Convolution Neural Network Approach to Doppler Spectra Classification of 205 MHz Radar

Baazil P Thampy^{1*}, Shailesh S^{1†}, Judy M V^{1†} and Ajil Kottayil^{2†}

^{1*}Department of Computer Applications, Cochin University of Science and Technology, Kalamassery, Cochin, 682022, Kerala, India.

²Advanced Centre for Atmospheric Radar Research, Cochin University of Science and Technology, Kalamassery, Cochin, 682022, Kerala, India.

*Corresponding author(s). E-mail(s): baazilpthampy@gmail.com;

Contributing authors: shaileshsivan@gmail.com;

judy.nair@gmail.com; ajil.acarr@cusat.ac.in;

[†]These authors contributed equally to this work.

Abstract

Wind profiler radars are capable of measuring three-dimensional wind profiles at various altitudes of the atmosphere, at very high temporal and spatial resolution. The Advanced Centre for Atmospheric Radar Research (ACARR), located at Cochin University of Science and Technology (CUSAT), operates the World's first 205 MHz stratosphere-troposphere wind profiler radar which provides three-dimensional wind profiles for an altitude range of 315 m to 20 km. During non-rainy condition, the radar Doppler power spectrum bears the signature of ambient air motion whereas during rainy conditions, it contains signatures of both ambient air motion and fall velocity of rain droplets. The classification of Doppler power spectra for rainy (Precipitation) and non-rainy (Clear) conditions is necessary as wind profile retrieval from the former needs careful separation of ambient air motion from fall velocity of droplets. A manual classification of the power spectrum is cumbersome, time-consuming, and therefore not practical due to the vast database. This work intends to automate Doppler power

spectra classification using the deep learning Convolutional Neural Network (CNN). The proposed Convolutional Neural Network model gives a k-fold validation accuracy of 99.77 % and testing accuracy of 99.60% for power spectra classification. The performance of CNN is compared against other popular machine learning classifiers such as Support Vector Machine, Decision Tree, K Nearest Neighbour and Naive Bayes. The performance comparison results show that the proposed CNN outperforms other models in radar Doppler power spectra classification.

Keywords: Wind Profiler Radar, Radar Target Classification, Convolutional Neural Network

1 Introduction

Wind profiler radars measure accurate three-dimensional wind profiles from different layers in the atmosphere with high temporal and spatial resolutions. Doppler Beam Swinging (DBS) is the most commonly used method for wind profile estimation wherein radar signals are transmitted towards the atmosphere in zenith and off-zenith directions, the azimuth angles of which are orthogonal to one another. The received echoes are Doppler shifted; the Doppler power spectra (DPS) thus obtained will have the information of the wind velocity in the atmosphere. A unique very high frequency (VHF) stratosphere-troposphere (ST) wind profiler radar (WPR) at 205 MHz has been operational in Cochin (10.04° N; 76.33° E), India, since 2017 (see Fig. 1). This facility provides accurate three-dimensional wind profiles for an altitude range of 315 m to 20 km (Kottayil et al. 2016; Mohanakumar et al. 2017; Kottayil et al. 2018; Nithya et al. 2019; Kottayil et al. 2020).

Fig. 1 Aerial view of 205 MHz ST wind profiler radar array

The main scientific objective behind the installation of this radar is to study the dynamics of the Indian summer monsoon. It has been shown in (Fukao et al. 1985) that VHF radars are sensitive to both Bragg and Rayleigh scattering, which enables the simultaneous observation of both ambient air motion as well as fall velocity of precipitation from the radar echoes during rainy conditions. One of the significant advantages of the 205 MHz radar is its ability to discern background air motion from precipitation echoes. The 205 MHz radar is operated 24/7 during the southwest monsoon season (June, July, August, and September) every year, and usually, the Doppler power spectra are generated every 7 minutes. However, during monsoon season, due to frequent rains, the Doppler power spectra will have signals of both ambient air motion and fall velocity of raindrops. In order to derive wind profiles from the Doppler power spectra obtained during rainy conditions, echoes of the ambient air motion must be separate from the precipitation echoes as the former contains the actual air motion. Since Doppler power spectra are frequently generated from radar observations, the first step is to classify the Doppler power spectra obtained during rainy (Precipitation DPS) and non-rainy (Clear DPS) conditions. A manual inspection of the power spectrum one by one for classification is cumbersome, time-consuming, and therefore not practical due to the enormous database. To circumvent this issue, the use of automated classification methods is inevitable.

Artificial intelligence/Machine learning techniques are widely in use in several areas of radar remote sensing. Marzano et al. (2007), and Nanzer et al. (2009) have proposed a machine learning model that classifies radar targets using a Bayesian approach. Roberto et al. (2017) proposed a hydrometeor classifier based on parallel support vector machines that use radial basis function kernels. Ostrovsky and Yanovsky (2006) used a multi-parametric decision technique to create a neural network classifier for classifying turbulence. Ibrahim et al. (2009) demonstrate the potential of a multi-layer perceptron neural network for ground vehicle categorization using forward scattering radar. A fuzzy logic approach for radar target classification is documented elsewhere (Andric et al. 2005; Al-Sakka et al. 2013 ; Thompson et al. 2014). Convolution neural network (CNN), a deep learning-based classification technique, has been used to classify radar targets in various applications, such as polarimetric synthetic aperture radar image classification (Zhou et al. 2016), hyperspectral image classification (Yu et al. 2017), vehicle classification (Capobianco et al. 2017), drone classification (Kim et al. 2016), etc. Lin et al. (2021) demonstrated a framework for deep learning-based radar target classification by merging data augmentation techniques and region of interest data with a residual CNN.

It is challenging to manually design suitable features for radar image target classification due to the complex scattering processes and noises in radar images. The ability to automatically learn and extract useful features from the data makes deep learning (DL) models like CNN best suited in radar image target classification. In this work, for the first time, CNN is employed for the 205 MHz wind profiler radar Doppler power spectra classification for rainy and

non-rainy conditions. The primary motive is to implement a robust automated classification algorithm based on a DL approach and explore the efficacy of CNN for radar echo identification. The proposed DPS classification algorithm based on CNN can be applied on any other WPR, which demands segregation of DPS for rainy and non-rainy conditions.

The significant contributions of this work are as follows:

- Creation of DPS image dataset from one year of WPR raw data and labeling of the image dataset
- Developed a computational framework using deep learning for the automated doppler power spectra classification of 205 MHz WPR.

2 Methodology

The high-level architecture of the proposed solution is given in Fig. 2, which includes data acquisition, preprocessing, and classification. Furthermore, each of these steps is discussed in detail below.

Fig. 2 High-level architecture of proposed CNN classifier

2.1 Data Acquisition

The 205 MHz ST radar at Cochin is an active phased array radar with 619 antennae elements and a corresponding number of Transmitter receiver modules (TRM), each 500 W. The antennae are arranged in a triangular grid with an inter-element spacing of 0.7λ where λ is 1.43 m. The effective aperture area of the radar is $536 m^2$, with a peak power aperture product of $1.6 \times 10^8 Wm^2$. The gain of the entire array is 35 dBi with a half-power beamwidth of 3.2° . Electronic beam steering helps in tilting the beam $0-30^\circ$ in the off-zenith direction and $0-360^\circ$ in the azimuth direction, both with an angular resolution of 1° . Coded and uncoded modes of operations can be performed with the ST radar. For code length, 1–64 bits can be given in steps of 2n. Baud can be set within a range of 0.3 to $4.8 \mu s$ in steps of $0.3 \mu s$. The technical specifications of the 205 MHz radar are given in Table 1. More details on radar validation and technical details can be found in Kottayil et al. (2016), and Mohanakumar et al. (2017).

Parameters	Specification
Frequency	205MHz
Bandwidth	5MHz
Type of system	Active phased array with TRM
Antenna element	Three-element Yagi-Uda antenna
Height coverage	315 m–20km
Modes of operation	DBS
Modulation	Binary phase shift keying (BPSK) coded compression
Code	Complementary code/Barker code
Peak power	$1.6 * 10^8 Wm^2$

Table 1 Key specifications of 205-MHz WPR located at ACARR, Cochin

2.2 DPS Image

The received radar echoes are used to estimate the Doppler power spectrum. As a first step, the complex time-domain signals (in-phase and quadrature-phase) after pulse decoding are averaged for n number of pulses for each range gate separately. The above process is repeated until ample fast Fourier transform (FFT) points are archived. After applying windowing operations (rectangular, Hamming, or Hanning) on time series data, these are subjected to FFT to generate the Doppler power spectra. The DPS for non-rainy and rainy conditions are shown in Fig. 3. For non-rainy cases, the DPS contains information on the ambient air motion. In contrast, in rainy cases, both ambient air motion and the fall velocity of rainfall can be seen. Unlike non-rainy conditions, the rainy condition will have multiple Doppler spectral peaks. Since our main motive is to classify DPS for rainy and non-rainy cases, we have generated the images of DPS for the whole of the year 2017. The images of DPS are segregated for rainy and non-rainy cases.

Fig. 3 Doppler power spectra for (a) non-rainy and (b) rainy conditions.

2.3 Preprocessing

Before training CNN with the Doppler power spectra images, the images are preprocessed. The images in the dataset are three-channel (RGB) color images of varying sizes, which may have been affected by different noise signals. In order to make the images into a uniform size, remove the noise, and enhance the quality, these images are processed using a preprocessing pipeline (see Fig. 4). Contrast-limited adaptive histogram equalization (CLAHE) (Zuiderveld 1994) is applied on the input image for boosting local contrast and enhancing edge definitions. The output of histogram equalization is then subjected to noise elimination using the Wiener filter (Lim 1990). The Wiener filter is a linear adaptive filter that smoothens noise while preserving edges and other high-frequency parts of an image. Then resize operation is performed on the denoised images to scale down the dimensions of images to 120x120 pixels with three channels. The output images are then normalized to a smaller interval [0,1] for making the computation easier. Once the preprocessing is over, the preprocessed images are used for training the classifier models.

Fig. 4 (a) Various image preprocessing steps applied on DPS, (b) Preprocessing of a sample DPS image.

2.4 Implementation of CNN

A deep learning-based model called the convolution neural network model (LeCun et al. 2010) is used for building the classifier. CNN-based models are widely used in various research domains as these models can automatically extract vital features and classify images with better accuracy.

2.4.1 Working of CNN

The architecture of the CNN models can be divided into two major phases. The first phase of the model is the convolutional phase that extracts the relevant features automatically, and the second phase is the neural network phase that performs the classification. The features are extracted from the images

using different kernels on the forward pass in the feature extraction phase. The primary operations associated with this phase are convolution, activation function, and pooling. The main elements of the convolution operation are the input image, feature detector (also known as Kernel or filter), and output features (Feature maps). Feature maps are obtained by an elementwise multiplication of the feature detector with the input and then summing up the results into a single output pixel. The feature detectors then slide over every input image location and repeat the above operation within the image. After the convolution operation, the activation function is applied to every element of output features to introduce nonlinearity in the output. The pooling layer reduces the dimension of the output features, keeping the essential information intact. So pooling limits the number of parameters to learn and the number of computations performed. The output obtained from the feature extraction phase is a set of feature maps, which is flattened into a vector. This vector holds all the relevant features that are needed to be extracted from the preprocessed DPS image.

The classification phase consists of several nodes or neurons distributed over various layers. The layers in this phase can be generally categorized as the input layer, hidden layer, and output layer. The input layer takes the feature vector of the previous phase as input for the next layer. The hidden layer is located between the input layer and output layer. Every node (except the input layer nodes) receives inputs from a subset of other nodes and performs nonlinear transformations from the inputs to outputs. The output layer is the last layer of the neural network phase that produces the classification results. The learning of the CNN model happens in two passes: forward pass and backward pass. In forward pass (forward propagation), the data flow is from phase one to phase two, whereas data (backpropagation) flow is reversed in the backward pass. The classification error is calculated after every training instance, and the weights and kernels are updated accordingly in the backpropagation. While designing a custom CNN model for image classification on new datasets, one can repeat the convolution, pooling, and activation function layers an arbitrary number of times to ensure the proper convergence and desired classification accuracy. The inference from the trained CNN model can be used for classifying new images during testing. Unlike the training, none of the learned parameters are updated during testing, and the data flows only in one direction from the convolutional phase to the classification phase.

2.4.2 Architecture of Proposed CNN Model

In this work, we have created a simple sequential CNN model with three convolution layers to extract features followed by three layers of the fully connected neural network for classification. The first layer of CNN is the Convolutional 2D layer which takes the preprocessed images of dimension 120x120x3 as input and produces output features of dimension 120x120x10. The first convolutional layer uses ten filters, each of shape 3x3 with stride [1, 1] and padding. The output of the first convolutional layer is fed into a Rectified Linear Unit (ReLU)

(Nair and Hinton 2010) activation function followed by a max-pooling layer (Nagi et al. 2011) which selects the maximum element from the region of the feature map covered by the filter (2x2) with stride [2,2]. ReLU is a simple activation functions that can produce excellent results in the classification problem when clubbed with layers of CNN. The mathematical definition of ReLU activation function is given in the Equation 1 and its value ranges from 0 to $+\infty$. The output shape of the Max pooling layer is 60x60x10. After extracting the high-level features, the above stages are repeated thrice to extract lower-level features.

$$ReLU(x) = \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases} \quad (1)$$

Fig. 5 Architecture of the proposed CNN model used in this study.

The second Convolutional layer takes the output of the previous Max pooling layer as input and extracts the lower level features using filters of shape 3x3 with stride [1, 1] and padding. At this layer, ten feature maps are extracted using the filters and the ReLU activation function. The second convolution layer produces output features of shape 60x60x10, and it is then transferred to the second Max pooling layer. The second Max pooling layer also uses the filters of size 2x2 with stride [2,2] and produces the output of shape 30x30x10. The output of the second Max pooling layer is fed into the third Convolutional layer. This Convolution layer uses filters of size 3x3 with stride [1, 1] and padding to extract ten feature maps. The output features of dimension

30x30x10 produced after the third convolution layer are then passed through ReLU activation followed by Max Pooling using 2x2 filters with stride [2,2] and padding. The output generated by the third Max Pooling layer is of the dimension 15x15x10. The output of dimension 15x15x10 from the third Max pooling layer is flattened and then fed to a fully connected neural network containing three layers, including the output layer. The first two layers of the fully connected NN are called the hidden layers. Layer 1 and layer 2 of fully connected NN use the ReLU activation function, and each layer contains 64 and 32 nodes, respectively. The final layer of the neural network is called the output layer, which has two nodes representing the two classes of the classification problem (clear echoes and precipitation echoes). The output layer uses the softmax function (Wang et al. 2018) for predicting the classes along with their probability value which is calculated using the mathematical equation 2.

Fig. 6 Feature maps obtained from the output of convolutional layers (a) on sample clear DPS image (b) on sample precipitation DPS image

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^N e^{x_j}} \quad (i = 1, 2, \dots, N), \quad (2)$$

where N is the number of classes, $(x_1, x_2, \dots, x_N)$ are the input values and $f(x_i)$ is the output value that gives the probability of class i which will be in the range of 0 to 1. In our case, since we are using binary classification, the value

of N is 2. The summary of the CNN model is as shown in Fig. 5. The feature maps obtained from the output of three convolutional layers by giving sample DPS images are shown in Fig. 6.

2.4.3 Training the Proposed CNN Model

For efficiently training a CNN model, a large amount of sample data is required. A balanced dataset containing 10,000 images (5,000 images each for non-rainy and rainy classes) created with the help of domain experts is used for training and testing the CNN model. After preprocessing the image dataset, it is divided randomly into three subsets: training set (70%), validation set (10%) and testing set (20%). The training set is employed for training the neural network model to learn the relationship between the inputs and the output class labels by adjusting the network weight parameters. The validation set is used for tuning the selected model during training. The testing set contains data samples unseen by the model and is used to evaluate the model performance. The CNN model is trained using a supervised approach that uses the input data and the actual class labels for training. An adaptive learning rate optimization algorithm called Adam optimizer (Kingma and Ba 2014) is used for training the model. Moreover, the most common binary classification loss function called binary cross-entropy is used to compute the loss between actual labels and predicted labels to optimize model performance. The mathematical definition of the loss function used in this work is given in the equation 3. Here y is the label and the value for class clear and precipitation is 0 and 1 respectively. Also $p(y)$ is the probability for one class, and $1 - p(y)$ is the probability for the other class.

$$Loss = \frac{1}{N} \sum_{i=1}^N - (y_i \times \log(p(y_i)) + (1 - y_i) \times \log(1 - p(y_i))) \quad (3)$$

Fig. 7 Performance of the proposed CNN model during training: (a) Model accuracy (b) Model loss

During the training process, the samples from the training sets are divided into equal size batches (40 samples each), and for each iteration, only one batch is used to train the network. In a single training epoch, all batches are fed to the model one by one in 175 iterations. The weights and biases of the proposed model are iteratively adjusted to minimize the loss. The number of epochs used for training the proposed model is set as 50. The model accuracy and loss give a more nuanced view into the model's training and validation performance during each epoch. Model loss takes the probability of a prediction and measures how much the prediction deviates from the actual value. The count of the predictions where the predicted value is equal to the actual value is used to find model accuracy. For a good learning model, as the number of epochs increases, the model loss should decrease, and model accuracy should increase. Ideally, the model loss and model accuracy should be 0 and 1, respectively. As shown in Fig. 7, after 50 epochs, the proposed model has model loss and accuracy close to 0 and 1, respectively. Also, by observing the training and validation plots, we can discard the possibility of overfitting, as the proposed model works well on both training and validation data. Once the model training and validation are over, the inference can be used to classify Doppler power spectra with less computation time.

3 Results and Performance Analysis

The skill of the proposed CNN model is estimated using the technique of k-fold cross-validation with the value of k as 5. In k-fold cross-validation, the dataset is divided randomly into 5 groups. During the first fold, one group is hold-out for testing, and the remaining groups are used to train the model. After training, the hold-out dataset is used for evaluating the model, and the evaluation scores are obtained. The model is then discarded, and the above process is repeated 4 more times by choosing a different group as hold-out data. The training accuracy and loss of each fold is shown in Fig. 8, and the validation results obtained during each fold are shown in the Table 2. Box plot shown in Fig. 9 gives a better understanding of the spread out of the accuracy and loss values obtained on each fold. The mean values of the accuracy and loss in the k-fold validation is 99.77% and 0.0184, respectively, and its corresponding median values are 99.8% and 0.0234 respectively.

Fold	Accuracy (%)	Loss
Fold 1	99.90	0.0042
Fold 2	99.70	0.0283
Fold 3	99.60	0.0234
Fold 4	99.80	0.0310
Fold 5	99.85	0.0051

Table 2 K-fold cross-validation scores of proposed CNN model (K=5)

Fig. 8 Training plots of the proposed CNN model during K-fold cross-validation: (a) Training accuracy (b) Training loss

Fig. 9 K-fold cross-validation score representation using box plots: (a) Accuracy (b) Loss

The performance of the proposed CNN model is compared with the four traditional supervised machine learning (ML) classification algorithms such as Naive Bayes (NB) (Rish et al. 2001), Support Vector Machine (SVM) (Cortes and Vapnik 1995), Decision Tree (DT) (Breiman et al. 2017), and K Nearest Neighbour (KNN) (Cover and Hart 1967). In ML approaches, the features from the image have to be extracted manually for training the model. In this study, we have extracted features directly from the Doppler images using efficient image features extraction methods such as Histogram of Oriented Gradients (HOG) (Dalal and Triggs 2005), Hu Moment (Hu 1962), Gray Level Co-occurrence Matrix (GLCM) (Haralick et al. 1973), and Local Binary Pattern (LBP) (Ahonen et al. 2004). HOG is used to extract features related to the edges, and LBP gives the surface texture information. Hu Moments and GLCM provide information about the shape of the objects in the image. The extracted features are then combined to form the feature set for training the

machine learning models. As the number of dimensions in the combined feature set is high, it may cause the dimensionality curse problem (Verleysen and François 2005). So feature selection followed by principal component analysis is used to reduce the dimensionality of the feature set. The reduced feature set is then used to train the ML classifier models such as SVM, KNN, DT and NB. The trained ML models are then evaluated on the testing dataset, and their performance is compared with the proposed CNN model.

The classification performance evaluation metrics used in this paper to compare the various experimental model's performance are confusion matrix, precision, recall, f1-score, and accuracy. The confusion matrix is a matrix that provides a simple and easy-to-understand visualization of the classifier model performance. Each row of the matrix shows instances of the actual

		Predicted Class		
		Clear	Precipitation	
Actual Class	Clear	A_cP_c	A_cP_p	A_c : Actual Clear A_p : Actual Precipitation P_c : Predicted Clear P_p : Predicted Precipitation
	Precipitation	A_pP_c	A_pP_p	

Fig. 10 Overview of confusion matrix

class, and each column shows instances of the predicted class. In the confusion matrix shown in Fig.10, the entry A_pP_c , represents the number of instances of actual class precipitation predicted as class clear. The principal diagonal cells (A_cP_c and A_pP_p) of the matrix give the number of correct classifications, and secondary diagonal cells (A_cP_p and A_pP_c) give the number of wrong classifications. A good classifier model will have higher values in principal diagonal cells and lower values in secondary diagonal cells. The confusion matrix of various experimental models on the test dataset is shown in Fig. 11. The obtained matrix results show that the proposed CNN model is better than the other experimental ML models for our DPS image dataset.

The value of performance metrics such as precision, recall and f1-score ranges from 0 to 1. The higher the ratio, the better will be the model performance. The $\text{precision}_{\text{class } x}$ gives the ratio of correctly predicted instances of class x to all instances predicted as class x (see equations 4 and 5), whereas $\text{recall}_{\text{class } x}$ gives the ratio of correctly predicted instances of class x to all instances that belong to actual class x (see equation 6 and 7). The weighted average of precision and recall is the f1-score, and is calculated using the equations 8 and 9.

$$\text{Precision}_{\text{Clear}} = \frac{A_cP_c}{A_cP_c + A_pP_c} \quad (4)$$

Fig. 11 Confusion matrix of various experimental models on test data

$$\text{Precision}_{\text{Precipitation}} = \frac{A_p P_p}{A_p P_p + A_c P_p} \quad (5)$$

$$\text{Recall}_{\text{Clear}} = \frac{A_c P_c}{A_c P_c + A_c P_p} \quad (6)$$

$$\text{Recall}_{\text{Precipitation}} = \frac{A_p P_p}{A_p P_p + A_p P_c} \quad (7)$$

$$\text{F1_score}_{\text{Clear}} = 2 * \frac{\text{Precision}_{\text{Clear}} * \text{Recall}_{\text{Clear}}}{\text{Precision}_{\text{Clear}} + \text{Recall}_{\text{Clear}}} \quad (8)$$

$$\text{F1_score}_{\text{Precipitation}} = 2 * \frac{\text{Precision}_{\text{Precipitation}} * \text{Recall}_{\text{Precipitation}}}{\text{Precision}_{\text{Precipitation}} + \text{Recall}_{\text{Precipitation}}} \quad (9)$$

From the comparison results shown in Fig. 12, it is evident that the proposed CNN model gives better evaluation scores than other models. Another important graphical representation that can validate the proposed model is the Receiver Operator Characteristic (ROC) curve and Area Under the Curve (AUC). The x-axis of the ROC curve is the TPR (True Positive Rate) which is same as the recall (see equations 6 and 7), whereas y-axis is the FPR (False Positive Rate) (see equations 10 and 11). The ROC curve of a class is plotted by calculating the TPR and FPR at different threshold points from the class's prediction probability, which ranges from 0 to 1. A good classifier model should

Fig. 12 Evaluation chart of various experimental models on test data (a) Precision (b) Recall and (c) F1-score.

have high TPR and low FPR. The information from a ROC curve can be summarized using AUC, which measures the area under the ROC curve. An ideal classifier will have an AUC value of 1. The ROC-AUC plot of the proposed CNN model is shown in Fig.13, and the AUC value obtained for both classes clear and precipitation is 1 (value approximated to 4 decimal places).

$$\text{FPR}_{\text{Clear}} = \frac{A_p P_c}{A_p P_c + A_p P_p} \quad (10)$$

$$\text{FPR}_{\text{Precipitation}} = \frac{A_c P_p}{A_c P_p + A_c P_c} \quad (11)$$

The accuracy mentioned in this paper is the overall accuracy, and it is the percentage of correct predictions on total predictions or observations (see equation 12). The accuracy results obtained from different ML classifiers and the proposed CNN model on the testing dataset are tabulated in the Table 3. The overall accuracy of the proposed CNN model is 99.60%, and it shows better skill in Doppler spectra classification as compared to other models. SVM

Fig. 13 ROC-AUC plot of the proposed CNN model

also performs well, though the classification accuracy (98.70 %) is lower than CNN. The classification accuracy of KNN, DT, and NB classifier models are 98.45%, 96.75%, and 93.30%, respectively.

$$\text{Overall Accuracy}(\%) = \frac{A_c P_c + A_p P_p}{A_c P_c + A_p P_p + A_c P_p + A_p P_c} \times 100 \quad (12)$$

	SVM	KNN	DT	NB	CNN
Overall Accuracy (%)	98.70	98.45	96.75	93.30	99.60
Precision_{clear}	0.9861	0.9821	0.9737	0.9190	0.9960
Precision_{Precipitation}	0.9879	0.9869	0.9614	0.9481	0.9960
Recall_{clear}	0.9880	0.9870	0.9611	0.9501	0.9960
Recall_{Precipitation}	0.9860	0.9819	0.9739	0.9157	0.9960
F1_{score}_{clear}	0.9871	0.9846	0.9674	0.9343	0.9960
F1_{score}_{Precipitation}	0.9870	0.9844	0.9676	0.9316	0.9960

Table 3 Performance comparison table of proposed CNN model with other experimental models

4 Summary and Conclusions

A unique stratosphere-troposphere wind profiler radar at 205 MHz has been operational at the tropical location of Cochin since 2017 January, probing an altitude range of 315 m to 20 km. The primary aim behind the setting up of 205MHz radar at this particular location is to understand the circulation features associated with the Indian summer monsoon. This WPR is the one

and only WPR operational in the World at this frequency. The Doppler power spectra obtained from radar during rainy and non-rainy conditions have different features. The Doppler power spectra obtained will have the signatures of ambient air motion and fall velocity of raindrops during rainy conditions. In contrast, DPS contains signatures of only ambient air motion for non-rainy cases. In order to classify the Doppler power spectra for rainy and non-rainy conditions, an automated approach is necessary.

In this work, we have implemented the convolution neural network to classify the Doppler power spectra from the 205 MHz wind profiler for the first time. The accuracy of CNN for DPS classification is assessed using various metrics such as precision, recall and f1-score, and all these values are close to 1. The performance of CNN is compared against other well-known image classifiers that use machine learning algorithms such as Support Vector Machine, Decision Tree, K Nearest Neighbour, and Naive Bayes. It is found that the image classification accuracy of CNN is better than the other models and is around 99.60%. The proposed CNN model can automatically extract key features from the DPS image and perform better than the combined classification efficiency of four powerful traditional feature extractors paired with standard ML classifiers. One of the challenges existing with designing a simple optimal custom CNN model is fixing the number of convolution layers, pooling layers, activation layers, and the number of hidden layers and nodes in the fully connected part. For this, we have experimented with different combinations of these layers and found out that we get better results with the proposed CNN model with three convolution layers, three max-pooling, three relu layers, and three dense layers. In this work, we have used an exhaustive sweep search to perform hyperparameter tuning.

Acknowledgments. We thank the support rendered by ACARR for carrying out the research work and sharing the DPS dataset. The financial support provided by the Ministry of Earth Sciences (MoES), Government of India for the sustenance of the ST Radar Facility is greatly acknowledged.

Declarations

- Ethics approval: Not applicable
- Consent to participate: Not applicable
- Consent for publication: Not applicable
- Conflict of interest: The authors declare no competing interests
- Availability of data : The dataset used in this research is available upon reasonable request from the ACARR facility.
- Code availability: The codes used in this research are available upon reasonable request from the corresponding author.

References

- Ahonen T, Hadid A, Pietikäinen M (2004) Face recognition with local binary patterns. In: Pajdla T, Matas J (eds) *Computer Vision - ECCV 2004*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 469–481, https://doi.org/10.1007/978-3-540-24670-1_36
- Al-Sakka H, Boumahmoud AA, Fradon B, et al (2013) A new fuzzy logic hydrometeor classification scheme applied to the french x-, c-, and s-band polarimetric radars. *Journal of Applied Meteorology and Climatology* 52(10):2328–2344. <https://doi.org/10.1175/JAMC-D-12-0236.1>
- Andric M, Durovic Z, Zrnic B (2005) Ground surveillance radar target classification based on fuzzy logic approach. In: *EUROCON 2005-The International Conference on "Computer as a Tool"*, IEEE, pp 1390–1392, <https://doi.org/10.1109/EURCON.2005.1630220>
- Breiman L, Friedman JH, Olshen RA, et al (2017) *Classification and regression trees*. Routledge, <https://doi.org/10.1201/9781315139470>
- Capobianco S, Facheris L, Cuccoli F, et al (2017) Vehicle classification based on convolutional networks applied to fmcw radar signals. In: *Italian Conference for the Traffic Police*, Springer, pp 115–128, https://doi.org/10.1007/978-3-319-75608-0_9
- Cortes C, Vapnik V (1995) Support-vector networks. *Machine learning* 20(3):273–297. <https://doi.org/10.1007/BF00994018>
- Cover T, Hart P (1967) Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* 13(1):21–27. <https://doi.org/10.1109/TIT.1967.1053964>
- Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pp 886–893 vol. 1, <https://doi.org/10.1109/CVPR.2005.177>
- Fukao S, Wakasugi K, Sato T, et al (1985) Direct measurement of air and precipitation particle motion by very high frequency doppler radar. *Nature* 316(6030):712–714. <https://doi.org/10.1038/316712a0>
- Haralick RM, Shanmugam K, Dinstein I (1973) Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics SMC-3*(6):610–621. <https://doi.org/10.1109/TSMC.1973.4309314>
- Hu MK (1962) Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory* 8(2):179–187. <https://doi.org/10.1109/TIT.1962.1057692>

- Ibrahim N, Abdullah R, Saripan M (2009) Artificial neural network approach in radar target classification. *Journal of Computer Science* 5(1):23
- Kim BK, Kang HS, Park SO (2016) Drone classification using convolutional neural networks with merged doppler images. *IEEE Geoscience and Remote Sensing Letters* 14(1):38–42. <https://doi.org/10.1109/LGRS.2016.2624820>
- Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980
- Kottayil A, Mohanakumar K, Samson T, et al (2016) Validation of 205 mhz wind profiler radar located at cochin, india, using radiosonde wind measurements. *Radio Science* 51(3):106–117. <https://doi.org/10.1002/2015RS005836>
- Kottayil A, Satheesan K, Mohankumar K, et al (2018) An investigation into the characteristics of inertia gravity waves in the upper troposphere/lower stratosphere using a 205 mhz wind profiling radar. *Remote Sensing Letters* 9(3):284–293. <https://doi.org/10.1080/2150704X.2017.1418991>
- Kottayil A, Xavier P, Satheesan K, et al (2020) Vertical structure and evolution of monsoon circulation as observed by 205-mhz wind profiler radar. *Meteorology and Atmospheric Physics* 132(4):531–545. <https://doi.org/10.1007/s00703-019-00695-4>
- LeCun Y, Kavukcuoglu K, Farabet C (2010) Convolutional networks and applications in vision. In: *Proceedings of 2010 IEEE international symposium on circuits and systems*, IEEE, pp 253–256, <https://doi.org/10.1109/ISCAS.2010.5537907>
- Lim JS (1990) Two-dimensional signal and image processing. Englewood Cliffs
- Lin T, Chen X, Tang X, et al (2021) Deep learning based classification of radar spectral maps. *International Journal of Electrical and Electronic Engineering & Telecommunications* 10:99–104. <https://doi.org/10.18178/ijeetc.10.2.99-104>
- Marzano FS, Scaranari D, Montopoli M, et al (2007) Supervised classification and estimation of hydrometeors from c-band dual-polarized radars: A bayesian approach. *IEEE transactions on Geoscience and Remote Sensing* 46(1):85–98. <https://doi.org/10.1109/TGRS.2007.906476>
- Mohanakumar K, Kottayil A, Anandan V, et al (2017) Technical details of a novel wind profiler radar at 205 mhz. *Journal of Atmospheric and Oceanic Technology* 34(12):2659–2671. <https://doi.org/10.1175/JTECH-D-17-0051.1>

- Nagi J, Ducatelle F, Di Caro GA, et al (2011) Max-pooling convolutional neural networks for vision-based hand gesture recognition. In: 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), IEEE, pp 342–347, <https://doi.org/10.1109/ICSIPA.2011.6144164>
- Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In: *Icml*
- Nanzer JA, Rogers RL (2009) Bayesian classification of humans and vehicles using micro-doppler signals from a scanning-beam radar. *IEEE Microwave and Wireless Components Letters* 19(5):338–340. <https://doi.org/10.1109/LMWC.2009.2017620>
- Nithya K, Kottayil A, Mohanakumar K (2019) Determining the tropopause height from 205 mhz stratosphere troposphere wind profiler radar and study the factors affecting its variability during monsoon. *Journal of Atmospheric and Solar-Terrestrial Physics* 182:79–84. <https://doi.org/10.1016/j.jastp.2018.10.018>
- Ostrovsky Y, Yanovsky F (2006) Use of neural network for turbulence and precipitation classification procedure. In: 2006 International Conference on Mathematical Methods in Electromagnetic Theory, IEEE, pp 161–163, <https://doi.org/10.1109/MMET.2006.1689731>
- Rish I, et al (2001) An empirical study of the naive bayes classifier. In: *IJCAI 2001 workshop on empirical methods in artificial intelligence*, pp 41–46
- Roberto N, Baldini L, Adirosi E, et al (2017) A support vector machine hydrometeor classification algorithm for dual-polarization radar. *Atmosphere* 8(8):134. <https://doi.org/10.3390/atmos8080134>
- Thompson EJ, Rutledge SA, Dolan B, et al (2014) A dual-polarization radar hydrometeor classification algorithm for winter precipitation. *Journal of Atmospheric and Oceanic Technology* 31(7):1457–1481. <https://doi.org/https://doi.org/10.1175/JTECH-D-13-00119.1>
- Verleysen M, François D (2005) The curse of dimensionality in data mining and time series prediction. In: Cabestany J, Prieto A, Sandoval F (eds) *Computational Intelligence and Bioinspired Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 758–770, https://doi.org/10.1007/11494669_93
- Wang M, Lu S, Zhu D, et al (2018) A high-speed and low-complexity architecture for softmax function in deep learning. In: 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), IEEE, pp 223–226
- Yu S, Jia S, Xu C (2017) Convolutional neural networks for hyperspectral image classification. *Neurocomputing* 219:88–98. <https://doi.org/10.1016/j.neurocom.2017.08.048>

[neucom.2016.09.010](#)

Zhou Y, Wang H, Xu F, et al (2016) Polarimetric sar image classification using deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters* 13(12):1935–1939. <https://doi.org/https://doi.org/10.1109/LGRS.2016.2618840>

Zuiderveld K (1994) Contrast limited adaptive histogram equalization. *Graphics gems* pp 474–485