

# Binary classification of rainfall time-series using machine learning algorithms

Shilpa Hudnurkar, Neela Rayavarapu

Department of Electronics and Telecommunication Engineering, Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, India

## Article Info

### Article history:

Received Mar 22, 2021

Revised Jul 19, 2021

Accepted Aug 14, 2021

### Keywords:

Artificial neural network  
Binary classification  
Summer monsoon rainfall  
Support vector machine

## ABSTRACT

Summer monsoon rainfall contributes more than 75% of the annual rainfall in India. For the state of Maharashtra, India, this is more than 80% for almost all regions of the state. The high variability of rainfall during this period necessitates the classification of rainy and non-rainy days. While there are various approaches to rainfall classification, this paper proposes rainfall classification based on weather variables. This paper explores the use of support vector machine (SVM) and artificial neural network (ANN) algorithms for the binary classification of summer monsoon rainfall using common weather variables such as relative humidity, temperature, pressure. The daily data, for the summer monsoon months, for nineteen years, was collected for the Shivajinagar station of Pune in the state of Maharashtra, India. Classification accuracy of 82.1 and 82.8%, respectively, was achieved with SVM and ANN algorithms, for an imbalanced dataset. While performance parameters such as misclassification rate, F1 score indicate that better results were achieved with ANN, model parameter selection for SVM was less involved than ANN. Domain adaptation technique was used for rainfall classification at the other two stations of Maharashtra with the network trained for the Shivajinagar station. Satisfactory results for these two stations were obtained only after changing the training method for SVM and ANN.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



## Corresponding Author:

Shilpa Hudnurkar

Department of Electronics and Telecommunication Engineering, Symbiosis Institute of Technology,

Symbiosis International (Deemed University)

Symbiosis Knowledge Village, Pune - 412115, India

Email: shilpa.hudnurkar@sitpune.edu.in

## 1. INTRODUCTION

The distribution of rainfall during the monsoon months varies both spatially and temporally. It is observed that weather parameters such as temperature and pressure undergo a gradual change in the winter and pre-monsoon seasons, however, during the summer monsoon months, namely June to September, these parameters undergo sudden changes. Depending on the weather parameters, intensity, and frequency of rainfall change region-wise [1]. Wind direction also plays an important role in rainfall events. Hence, especially during the monsoon season, planning for day-to-day activities such as commuting to work, will require accurate information regarding the day's weather. Rainfall can vary in intensity, variability, frequency [2]. However, on a day-to-day basis, information on whether or not it will rain is adequate.

Machine learning algorithms such as k-nearest neighbor (k-NN), support vector machine (SVM), artificial neural network (ANN), decision tree (DT), random forest (RF) have been used by many researchers for the purpose of rainfall classification. Kulkarni *et al.* [3] used K-means clustering and map-to-map

correlation methods for rainfall pattern classification over India. Zhang *et al.* [4] observed that the geographical characteristics of a region influence rainfall prediction accuracy and they proposed a model using K-means clustering, and a convolutional neural network for rainfall forecasting. The authors presented a multistep approach for rainfall forecasting where K-means clustering was utilized for selecting meteorological data of surrounding stations, then the high-altitude shear value, considering surrounding stations' meteorological factors, was calculated. In the third step, principal component analysis was used for dimensionality reduction of features, and finally, a convolution neural network was used for rainfall forecasting. Michaelides *et al.* [5] used ANN for rainfall variability classification. Loś *et al.* [6] employed RF for storm nowcasting using integrated water vapor (IWV) with vertical profiles of wet refractivity derived from global navigation satellite system (GNSS) as predictors. RF and SVM was used by Pour *et al.* [7] for downscaling of rainfall and prediction, respectively. For downscaling, predictors from a set of twenty-six variables, collected from the National Centre for Environmental Prediction (NCEP), reanalysis data were used. RF, deep neural network (DNN), and SVM were used by Sangiorgio *et al.* [8] for intense convective rainfall event classification. They found that DNN and RF performed better than SVM. Byakatonda *et al.* [9] used ANN to model drought severity. Raju *et al.* [10] compared the performance of RF, DTs SVM outperforming the k-NN, RF, and DT models. GNSS cloud data, along with other meteorological parameters, k-NN, and SVM for the classification of rainfall. They found, was given as input to the nonlinear autoregressive exogenous neural network model, for hourly rainfall classification by Benevides *et al.* [11]. Chai *et al.* [12] classified rainfall for flood prediction and compared backpropagation and radial basis function (RBF) ANNs. The region considered for this study was the Kuching city of Malaysia. They used the daily data of six meteorological parameters to classify rainfall, into four different classes, from light precipitation to very heavy precipitation. They found that BPN performed better than the RBF algorithm. However, the authors expressed the need for trials required for selecting the number of neurons and other network parameters in the case of BPN. Richetti *et al.* [13] used expectation-maximization (EM), K-means clustering and DT for the classification of regions having homogeneous rainfall in the Parana state of Southern Brazil. They first clustered the regions using EM and K-means clustering and then used the J48 algorithm to determine the number of regions having similar characteristics. Hussein *et al.* [14] have used SVM for the classification of large-scale precipitation maps. In a different approach, Rustam *et al.* [15] presented a method to handle an imbalanced dataset for SVM. Maldonado and Lopez also addressed the issue of the imbalanced dataset by proposing an embedded feature selection method [16]. In both [11] and [16], the authors found improved accuracy of SVM for the datasets for which they were tested.

The effect of adding an input parameter, on the extreme rainfall event multiclass classification, was inspected by Sangiorgio *et al.* [17] They compared the performance of logistic regression and DNN with weather parameters as inputs and found that with the addition of an input parameter selected by them, there was an improvement in the accuracy of the classification. Many researchers have used parameters derived from the GNSS and radar for analysis [18], classification [8], and nowcasting [6] of the rainfall, storms, thunderstorms. The parameters derived from GNSS include zenith tropospheric delay (ZTD) [8], [17], precipitable water vapor [11], [19], [20], IWV [21], and IWV with vertical profiles of wet refractivity [6], to name a few. Most of these are related to multiclass classification and utilize several different features for this purpose. In each case, the data used for classification, the region for which the classification was done, the features, and the algorithms used are all different.

The aim of this work is to build a simple, yet practical and adaptable model that can be used for any spatial region other than that for which it has been tested. If the number of inputs is large and difficult to acquire, the classification would fail. Hence, the classification model is trained and tested, with a small set of features. Here, ANN and SVM are chosen for the binary classification of rainfall on a given day as a "rainy" or "non-rainy" day. The features used for this classification are weather parameters such as temperature, humidity, and pressure. The daily values of these weather parameters are displayed on the website of the Indian Meteorological Department (IMD), and thus can be easily obtained. Results show that both the algorithms could classify rainfall days into the two classes for the selected region. In the second approach, with the trained networks for the selected region, the domain adaptation task of rainfall classification for the other two regions was undertaken [22], [23]. This was done to test the robustness of the classifiers for regions other than the one they were trained for. Due to the complexity of the weather systems and regional geographical conditions, the domain adaptation task exhibited low accuracy. Further, appended datasets were used to train the classifiers and the task of classification for other regions was achieved with fair accuracy.

The paper is arranged as follows. In section 2, details of data collection are given along with the methodology used. The SVM algorithm for classification and ANN classifier is briefly discussed in sections 3 and 4, respectively. Data cleaning and preprocessing are explained in section 5. In section 6, experimental results are discussed. Finally, the paper is concluded in section 7.

## 2. RESEARCH METHOD

The daily rainfall classification data used in this study was obtained from the National Data Center (NDC) of IMD. The datasets of various stations were obtained and the dataset from one of these namely, Shivajinagar Station (18.5314 N, 73.8446 E) of Pune, Maharashtra, India was used for training the networks. The datasets of two other stations, Nashik and Chikalthana, were also obtained. These contain the time series of daily surface parameters namely sea level pressure (SLP) in hectopascal (hPa), mean sea level pressure (MSLP) in hectopascal (hPa), relative humidity (RH) in percentage (%), maximum temperature in degree celsius (°C), minimum temperature in degree Celsius (°C), wind speed in kilometers per hour (kmph), wind direction in 16 points of compass and rainfall in millimeters (mm). The data was obtained for the years 2000 to 2018. Data preprocessing and algorithm implementation were accomplished in Python3. The methodology adopted was as follows:

- 1) Data cleaning and preprocessing
  - Daily data for the months of June through September were used after filtering out the data of the other months, for all the weather variables.
  - Data preprocessing for removing missing records was carried out.
  - Wind direction was treated as a categorical variable as it is based on the sixteen different numbers given for wind directions.
  - Weather parameters except rainfall were normalized using the min-max normalization method.
  - Rainfall data for each day was labeled. Less than 2.5 mm rainfall was labeled as “no-rain” day (label 0) and greater than or equal to 2.5mm rainfall was labeled as “rainy” day (label 1) [24].
  - The preprocessed dataset was split into 80:20 ratio for training and testing samples, respectively.
- 2) Applying SVM algorithm
  - SVM algorithm was applied to train and test data. Six weather parameters were used as features. Details of SVM are given in the subsequent sections.
  - The evaluation parameters used were accuracy, F1 score, and misclassification rate.
  - The network performance was evaluated using the test dataset.
- 3) Applying ANN algorithm
  - ANN algorithm was applied for classification, on the same dataset. The network parameter selection is discussed in the subsequent sections.
  - The network performance evaluation was done on the evaluation parameters that were used for SVM.
- 4) Testing the performance of SVM and ANN for other stations by domain adaptation
  - Station records of Nashik and Chikalthana stations, of the years 2016 and 2017, were preprocessed as per the steps in 1.
  - The dataset of the Shivajinagar station was completely used for training the networks and tested for Nashik and Chikalthana stations.
  - Steps 2 and 3 were repeated for these two stations.
- 5) Testing the performance of SVM and ANN for other stations by changing the training dataset
  - Two new datasets were prepared by adding records of Nashik and Chikalthana station to the dataset for the Shivajinagar station.
  - Steps 2 and 3 were repeated for these two stations.
- 6) Results obtained in steps 2 to 5 were compared.

## 3. CLASSIFICATION WITH SUPPORT VECTOR MACHINE

Developed in 1990, SVM, a supervised learning algorithm for classification was later extended to solve regression problems [25]. It is a machine learning algorithm that classifies  $N$  data points  $\{y_k, x_k\}$  for  $k = 1$  to  $N$  where  $y_k$  is  $k^{\text{th}}$  output and  $x_k$  is  $k^{\text{th}}$  input [26]. For this classification, it constructs a classifier given in (1),

$$y(x) = \text{sign}[\sum_{k=1}^N \alpha_k y_k \psi(x, x_k) + b] \quad (1)$$

where  $\alpha_k$  are Lagrange multipliers, and  $b$  is a real constant. It can classify linearly separable or non-separable data. Figures 1(a) and 1(b) show linearly separable and non-separable data points, respectively.

Various kernel functions such as linear, polynomial, and RBF help to map the data from one space to another, using hyperplanes for classification. Linear kernel function gives a one-dimensional plane (a straight line) for classifying data points. It uses the formula as in (2) [27].

$$\psi(x, x_k) = x_k^T x \quad (2)$$

Polynomial and RBF kernel functions use (3) and (4) respectively [27],

$$\psi(x, x_k) = x_k^T x \tag{3}$$

where  $d$  represents the degree of polynomial SVM,

$$\psi(x, x_k) = \exp\{-\|x - x_k\|^2 / 2\sigma^2\} \tag{4}$$

where  $\sigma$  is constant. These functions are commonly used for classification problems [28]. For nonlinear problems, polynomial and RBF functions are used [29].

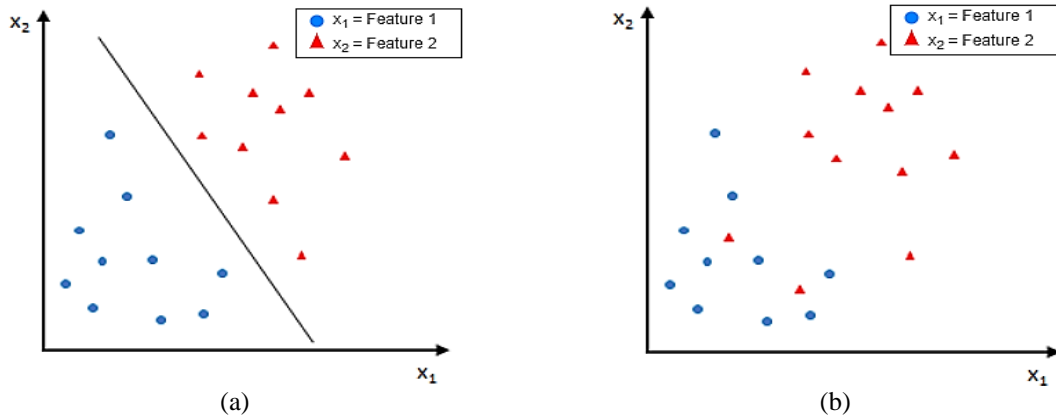


Figure 1. Data points corresponding to two features  $x_1$  and  $x_2$  respectively in (a) linearly separable and (b) linearly non-separable

#### 4. CLASSIFICATION WITH ARTIFICIAL NEURAL NETWORK

Supervised or unsupervised ANNs are widely used for classification and regression problems. The simple architecture of an ANN with one hidden layer is shown in Figure 2. Each layer consists of neurons where each neuron resembles the neuron of a human brain. Each neuron sums the input coming to it multiplied by a weight value, deciding how much significance is to be given to each input. It further uses the activation function on this multiplication result and outputs a value that connects it with the neuron of the next layer. The neurons in the output layer decide the output value. The network output is compared with the known expected output to compute training error. For each record, this error is calculated, and an algorithm is used to minimize this error by adjusting the weights and bias. Once this error reaches the predetermined minimum level, training stops. The network can then be tested for unseen data and its performance can be evaluated [30].

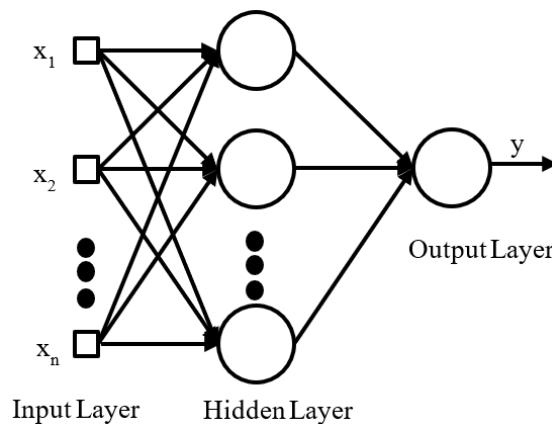


Figure 2. ANN with one hidden layer

Interconnections of the neurons in various layers decide the architecture of ANN. For solving complex problems, the number of hidden layers may be increased. During the whole process of training, appropriate hyperparameter selection is required. The selection of parameters such as the number of neurons, number of layers, learning rate, and activation function, require trials or the use of optimization algorithms. Activation function decides the nature of output of a neuron. Various activation functions such as Sigmoidal, TanH, rectified linear unit (ReLU), and the binary step function are in use [31]. For classification problems, the last layer of the network uses the sigmoidal function. Gradient descent, root mean square propagation (RMSProp), adaptive moment optimization (Adam) are the algorithms used for updating weights and bias.

## 5. DATA CLEANING AND PREPROCESSING

The Shivajinagar Station is at the heart of Pune city. The dataset represents a major area of Pune city. Hence, this station was selected for the study. This is under Madhya Maharashtra meteorological subdivision. The data obtained from NDC comprised of daily surface weather parameters, as stated earlier. For the application of machine learning algorithms, data cleaning and preprocessing were required, and this was carried out by checking the number of records available in the dataset. The dataset must be checked for any missing records, and they should be handled properly. Removing the records with missing values or interpolating those values using appropriate formulae are the two ways, among many, to handle missing values. However, to fill the missing values of a particular record, the data of the other variables of that record is required, in case of real-time assimilation of the weather database [32]. For most of the records with missing values, data pertaining to another one or two variables was also missing. Hence, for the days where records were not available for any of the weather parameters, that record was removed [33]. The data used for training has records for a period of over 15 years, hence, sufficient variation in training samples was available for the supervised network to learn. The number of records available from the year 2000 to 2018 for the summer monsoon months, for each weather parameter, before and after data cleaning, are listed in Table 1.

Table 1. Number of records before and after data cleaning stage

Weather parameter	Number of records before cleaning the dataset	Number of records after cleaning the dataset
Sea Level Pressure	2299	
Mean Sea Level Pressure	2299	
Relative Humidity	2299	
Wind Speed	2299	
Average Wind Speed	2293	2268
Wind Direction	2299	
Maximum Temperature	2303	
Minimum Temperature	2302	
Rainfall	2301	

Only 1.5% of the total records were removed during the data cleaning stage. The next stage was preprocessing for the purpose of feature selection. Any redundant features present were omitted to reduce the computational burden. SLP and MSLP are highly correlated and hence only one of them, SLP, was selected as an input feature. During the preprocessing stage, the wind direction was encoded to a categorical variable using the LabelEncoder function of Sklearn library. The standard wind directions indicated by IMD are as listed in Table 2.

The rest of the weather parameters referred to as input features hereafter were normalized using the min-max normalized method [34]. The formula for min-max normalization is given in (5),

$$x_{ni} = \frac{(x_i - X_{min})}{(X_{max} - X_{min})} \quad (5)$$

where  $x_{ni}$  is the normalized record,  $x_i$  is the record to be normalized,  $X_{min}$  is the minimum of the total records of vector  $X$  and  $X_{max}$  is the maximum of the total records of vector  $X$ .

The rainfall field was labeled in the following manner. For days where recorded daily rainfall was less than 2.5 mm, it was treated as no rain with label 0, otherwise, the field was labeled as 1. This criterion is as per the rainy-day definition by IMD [35]. After the data preprocessing stage, the dataset was divided into training and testing samples. Out of the total number of records, 80% of samples were used for training and 20% for testing. The dataset was thus prepared for binary classification.

Table 2. Wind directions as per 16-point compass

Number	Wind Direction
00	Calm
02	North north-east
05	Northeast
07	East north-east
09	East
11	East south-east
14	Southeast
16	South south-east
18	South
20	South south-west
23	Southwest
25	West south-west
27	West
29	West north-west
32	Northwest
34	North north-west
36	North
99	Variable

## 6. RESULTS AND DISCUSSION

The SVM classifier was first employed for the binary classification of the preprocessed dataset. To improve the performance of the SVM, that is to enable it to give low training and testing errors, one regularization parameter C is used. It optimizes the distance of the data point from the margin. The selection of the kernel has an impact on the model performance. High dimensional feature space could overfit the model [27]. Three different kernel functions were used, namely, linear, polynomial, and RBF.

Five-fold cross-validation was used in all the cases. All the three kernel functions were trained and tested for five-fold cross-validation, C values 0.01, 0.1 and 1. After the training, the network was tested for 454 records, already split from the original dataset. The confusion matrix was plotted, and the number of records obtained as true negative (TN), true positive (TP), false negative (FN), and false positive (FP) were substituted in the (6)-(8) for accuracy, precision, misclassification rate, and F1 score [36].

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \quad (6)$$

$$Misclassification\ Rate = (FP + FN)/(TP + TN + FP + FN) \quad (7)$$

$$F1_{score} = (2 * Recall * Precision) / (Recall + Precision) \quad (8)$$

where

$$Recall = TP/(TP + FN) \quad (9)$$

and,

$$Precision = TP/(TP + FP) \quad (10)$$

Experiments showed that linear SVM cannot provide a suitable performance. Hence, the testing performance of the polynomial and RBF kernel functions is summarized in Table 3 for the Shivajinagar station. The order of the polynomial function was changed from 2 to 7 and it was found that the performance of the third order polynomial was consistently good.

The dataset distribution, in this case, was uneven. The “non-rainy” days were almost double the “rainy” days. Hence, the F1-score is an important performance evaluator. As shown in Table 3, the polynomial kernel gave the best results for the Shivajinagar station with an 18% misclassification rate and a 69% F1 score.

Table 3. SVM classification performance for test cases of Shivajinagar station test results

SVM Kernel Function	Polynomial	RBF
Accuracy	0.821	0.79
Precision	0.769	0.705
F1_Score	0.69	0.638
Misclassification rate	0.18	0.2092

ANN was implemented for the Shivajinagar station. A three-layer ANN with four neurons and ReLU activation function in the first two layers and one neuron and sigmoidal function in the last layer was built. The RMSProp optimizer was selected and trained with 80% of the records. 20% of records were used for testing. TN, TP, FN, and FP obtained were used to calculate accuracy, misclassification rate, and F1 score. The results are summarized in Table 4.

Table 4. ANN classification performance for test cases of Shivajinagar station test results

Performance Parameter	ANN Classifier
Accuracy	0.828
Precision	0.761
F1_Score	0.711
Misclassification rate	0.1718

The number of neurons and number of layers was experimented with and the network with the best-performing parameters as mentioned above was selected. When compared with the performance of the polynomial SVM, as shown in Table 3, ANN gave a better F1-score and a slightly better misclassification rate. The accuracy and precision values of both networks are very close to each other.

India is divided into meteorological subdivisions [37] and Maharashtra has four such subdivisions based on rainfall homogeneity. However, large variability in the summer monsoon rainfall is observed across all subdivisions and within a given subdivision. The Shivajinagar station is from the Madhya Maharashtra subdivision. Although the dataset is imbalanced having 2/3rd “non-rainy” days and 1/3rd “rainy” days, the classification accuracy obtained for this station is 82%. The performance of this classification model was then tested for the other two stations.

Two stations, one from Madhya Maharashtra (Nashik) and the other from Marathwada (Chikalthana) were chosen for studying the domain adaptation technique. All the kernel functions, with their best parameters, were used to test the datasets of Nashik and Chikalthana stations. All the 2268 records of the Shivajinagar station were used for training the SVM and a separate dataset for Nashik and Chikalthana was preprocessed and used for testing. The data of the summer monsoon month of the years 2016 and 2017 was used for testing. The performance of the SVM, for this approach, is given in Table 5.

Table 5. SVM classification performance for test cases of Nashik and Chikalthana station with SVM trained with Shivajinagar data

Validation	SVM Kernel Function	Polynomial	RBF
Nashik validation results	Accuracy	0.578	0.385
	Precision	0.476	0.385
	F1_Score	0.633	0.556
	Misclassification rate	0.422	0.614
Chikalthana validation results	Accuracy	0.498	0.322
	Precision	0.374	0.322
	F1_Score	0.516	0.488
	Misclassification rate	0.502	0.677

Both the kernel functions performed poorly. However, it was observed that the polynomial kernel performed better than the RBF kernel. ANN was then implemented for testing Nashik and Chikalthana stations for the years 2016 and 2017. All the records of the Shivajinagar station were used for training the ANN. The architecture of ANN was not changed. The results obtained are summarized in Table 6.

Table 6. ANN classification performance for test cases of Nashik and Chikalthana station with ANN trained with Shivajinagar data

Validation	Performance Parameter	ANN Classifier
Nashik Validation Results	Accuracy	0.502
	Precision	0.4365
	F1_Score	0.607
	Misclassification rate	0.497
Chikalthana Validation Results	Accuracy	0.484
	Precision	0.375
	F1_Score	0.53
	Misclassification rate	0.52

For both the stations, the performances of ANN and SVM were comparable. However, the performance of both the networks was not very good when compared to their performance in the test case of the Shivajinagar station. The reasons behind this failure of the model were probed. Was the geographical distance between the stations affecting the performance of the network? Or, was it because the networks were not sufficiently trained? To search for answers to these questions the second approach was taken.

In the second approach, two separate datasets were prepared. In the first dataset, the Shivajinagar dataset was appended with Nashik records. In the second dataset, the Shivajinagar dataset was appended with Chikalthana records. Both the datasets were split in 80:20 proportion for training and testing. Results, in this case, are as shown in Table 7.

Table 7. SVM classification performance for test cases of Nashik and Chikalthana station with the network trained with an appended dataset

Validation	SVM Kernel Function	Polynomial	RBF
Nashik Validation Results	Accuracy	0.744	0.72
	Precision	0.7	0.691
	F1_Score	0.632	0.573
	Misclassification rate	0.256	0.28
Chikalthana Validation Results	Accuracy	0.779	0.767
	Precision	0.761	0.758
	F1_Score	0.65	0.618
	Misclassification rate	0.22	0.233

The results showed a significant improvement in evaluation parameters. For the new datasets prepared for Nashik and Chikalthana, ANN was used for the classification. The network architecture was kept the same and the dataset was split in an 80:20 ratio for training and testing. Results obtained for binary classification are summarized in Table 8.

The polynomial SVM performed better than the ANN for both stations. Although Nashik and Pune are geographically closer to each other and fall in the same meteorological subdivision, SVM and ANN networks gave better results for Chikalthana than Nashik. These experiments suggest that for supervised machine learning algorithms, the training dataset highly influences the performance of the algorithm. The data provided by the NDC, contained many missing years for many of the stations. This put limitations on training and testing of the network. To address this problem, the training dataset of one station appended with a few records, from recent years, of the station under test as used in this study would be helpful.

Table 8. ANN classification performance for test cases of Nashik and Chikalthana station with the network trained with an appended dataset

Validation	Performance Parameter	ANN Classifier
Nashik Validation Results	Accuracy	0.716
	Precision	0.7551
	F1_Score	0.5103
	Misclassification rate	0.284
Chikalthana Validation Results	Accuracy	0.755
	Precision	0.71
	F1_Score	0.62
	Misclassification rate	0.24

## 7. CONCLUSION

For the binary classification of rainfall during the summer monsoon months, SVM and ANN were used. Summer monsoon rainfall days were classified as “rainy” and “non-rainy” days for one station, namely, Shivajinagar. The optimal network in both the cases was used for the domain adaptation task at the other two stations: one from Madhya Maharashtra meteorological subdivision, Nashik, and the other from Marathwada subdivision, Chikalthana. Results obtained with the domain adaptation technique were less accurate. However, results obtained when the network trained with one station’s data and a few records of other stations revealed that the performance of SVM and ANN is comparable, and, successfully classified data points. The classification performance for Chikalthana was better than that for Nashik, in this case. It was observed that network performance is independent of geographical proximity when the networks are trained with the records of one station appended with a few records of the station under test.



## ACKNOWLEDGEMENTS

The authors would like to acknowledge India Meteorological Department for providing us with the data required for carrying out this research work.




## REFERENCES

- [1] S. Nandargi and S. S. Mulye, "Relationships between rainy days, mean daily intensity, and seasonal rainfall over the Koyna catchment during 1961-2005," *The Scientific World Journal*, vol. 2012, pp. 1–10, 2012, doi: 10.1100/2012/894313.
- [2] R. Vinnarasi and C. T. Dhanya, "Changing characteristics of extreme wet and dry spells of Indian monsoon rainfall," *Journal of Geophysical Research*, vol. 121, no. 5, pp. 2146–2160, Mar. 2016, doi: 10.1002/2015JD024310.
- [3] A. Kulkarni, R. H. Kripalani, and S. V. Singh, "Classification of summer monsoon rainfall patterns over India," *International Journal of Climatology*, vol. 12, no. 3, pp. 269–280, Apr. 1992, doi: 10.1002/joc.3370120304.
- [4] P. Zhang, W. Cao, and W. Li, "Surface and high-altitude combined rainfall forecasting using convolutional neural network," *Peer-to-Peer Networking and Applications*, vol. 14, no. 3, pp. 1765–1777, Jul. 2021, doi: 10.1007/s12083-020-00938-x.
- [5] S. C. Michaelides, C. S. Pattichis, and G. Kleovoulou, "Classification of rainfall variability by using artificial neural networks," *International Journal of Climatology*, vol. 21, no. 11, pp. 1401–1414, 2001, doi: 10.1002/joc.702.
- [6] M. Łoś, K. Smolak, G. Guerova, and W. Rohm, "GNSS-based machine learning storm nowcasting," *Remote Sensing*, vol. 12, no. 16, p. 2536, Aug. 2020, doi: 10.3390/RS12162536.
- [7] S. H. Pour, S. Shahid, and E. S. Chung, "A hybrid model for statistical downscaling of daily rainfall," *Procedia Engineering*, vol. 154, pp. 1424–1430, 2016, doi: 10.1016/j.proeng.2016.07.514.
- [8] M. Sangiorgio *et al.*, "A comparative study on machine learning techniques for intense convective rainfall events forecasting," in *Theory and Applications of Time Series Analysis*, 2020, pp. 305–317.
- [9] J. Byakatonda, B. P. Parida, P. K. Kenabatho, and D. B. Moalafhi, "Modeling dryness severity using artificial neural network at the Okavango Delta, Botswana," *Global Nest Journal*, vol. 18, no. 3, pp. 463–481, May 2016, doi: 10.30955/gnj.001731.
- [10] K. H. P. Raju, N. Sandhya, and R. Mehra, "Supervised SVM classification of rainfall datasets," *Indian Journal of Science and Technology*, vol. 10, no. 15, pp. 1–6, Apr. 2017, doi: 10.17485/ijst/2017/v10i15/106115.
- [11] P. Benevides, J. Catalao, and G. Nico, "Neural network approach to forecast hourly intense rainfall using GNSS precipitable water vapor and meteorological sensors," *Remote Sensing*, vol. 11, no. 8, Apr. 2019, doi: 10.3390/rs11080966.
- [12] S. S. Chai, W. K. Wong, and K. L. Goh, "Rainfall classification for flood prediction using meteorology data of Kuching, Sarawak, Malaysia: backpropagation vs radial basis function neural network," *International Journal of Environmental Science and Development*, vol. 8, no. 5, pp. 385–388, 2017, doi: 10.18178/ijesd.2017.8.5.982.
- [13] J. Richetti, J. Johann, and M. Uribe Opazo, "Data mining techniques for rainfall regionalization in parana state," *Acta Iguazu*, vol. 7, no. 1, pp. 1–8, 2018, doi: 10.48075/actaiguaz.v7i1.17777.
- [14] E. Hussein, M. Ghaziasgar, and C. Thron, "Regional rainfall prediction using support vector machine classification of large-scale precipitation maps," in *Proceedings of 2020 23rd International Conference on Information Fusion*, 2020, doi: 10.23919/FUSION45008.2020.9190285.
- [15] Z. Rustam, D. A. Utami, R. Hidayat, J. Pandelaki, and W. A. Nugroho, "Hybrid preprocessing method for support vector machine for classification of imbalanced cerebral infarction datasets," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 9, no. 2, pp. 685–691, Apr. 2019, doi: 10.18517/ijaseit.9.2.8615.
- [16] S. Maldonado and J. López, "Dealing with high-dimensional class-imbalanced datasets: embedded feature selection for SVM classification," *Applied Soft Computing Journal*, vol. 67, pp. 94–105, Jun. 2018, doi: 10.1016/j.asoc.2018.02.051.
- [17] M. Sangiorgio *et al.*, "Improved extreme rainfall events forecasting using neural networks and water vapor measures," in *6th International conference on Time Series and Forecasting (ITISE 2019)*, 2019.
- [18] M. Sangiorgio and S. Barindelli, "Spatio-temporal analysis of intense convective storms tracks in a densely urbanized Italian Basin," *ISPRS International Journal of Geo-Information*, vol. 9, no. 3, Mar. 2020, doi: 10.3390/ijgi9030183.
- [19] Q. Zhao, Y. Liu, W. Yao, and Y. Yao, "Hourly rainfall forecast model using supervised learning algorithm," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–9, 2021, doi: 10.1109/TGRS.2021.3054582.
- [20] P. Benevides, J. Catalao, and P. M. A. Miranda, "On the inclusion of GPS precipitable water vapour in the nowcasting of rainfall," *Natural Hazards and Earth System Sciences*, vol. 15, no. 12, pp. 2605–2616, 2015, doi: 10.5194/nhess-15-2605-2015.
- [21] G. Guerova, T. Dimitrova, and S. Georgiev, "Thunderstorm classification functions based on instability indices and GNSS IWV for the sofia plain," *Remote Sensing*, vol. 11, no. 24, Dec. 2019, Art. no. 2988, doi: 10.3390/rs11242988.
- [22] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: a deep learning approach," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 2011, pp. 513–520.
- [23] G. Guariso, G. Nunnari, and M. Sangiorgio, "Multi-step solar irradiance forecasting and domain adaptation of deep neural networks," *Energies*, vol. 13, no. 15, Aug. 2020, doi: 10.3390/en13153987.
- [24] K. R. Atal and A. Zende, "Wet and dry spell characteristics of semi-arid region, Western Maharashtra, India," *E-proceedings of the 36th IAHR World Congress*, no. July, 2015.
- [25] V. Vapnik, "The support vector method of function estimation," in *Nonlinear Modeling*, Springer US, 1998, pp. 55–85.
- [26] S. Gunn, "Support vector machines for classification and regression," *ISIS Tech. Rep.*, vol. 14, no. 1, pp. 5–16, 1998.
- [27] R. G. Brereton and G. R. Lloyd, "Support vector machines for classification and regression," *Analyst*, vol. 135, no. 2, pp. 230–267, 2010, doi: 10.1039/b918972f.
- [28] B. Yekkehkhany, A. Safari, S. Homayouni, and M. Hasanlou, "A comparison study of different kernel functions for SVM-based classification of multi-temporal polarimetry SAR data," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-2/W3, no. 2W3, pp. 281–285, Oct. 2014, doi: 10.5194/isprsarchives-XL-2-W3-281-2014.
- [29] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999, doi: 10.1023/A:1018628609742.
- [30] S. N. Sivanandam and S. N. Deepa, *Principles of Soft Computing, 2ND ED (With CD)*. Wiley India Pvt. Limited, 2011.
- [31] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: comparison of trends in practice and research for deep learning," *arxiv.org/abs/1811.03378v1*, Nov. 2018.
- [32] M. Sangiorgio, S. Barindelli, V. Guglieri, G. Venuti, and G. Guariso, "Reconstructing environmental variables with missing field data via end-to-end machine learning," in *Proceedings of the 21st EANN (Engineering Applications of Neural Networks) 2020 Conference*, Springer International Publishing, 2020, pp. 167–178.
- [33] A. Astorino, E. Gorgone, M. Gaudioso, and D. Pallaschke, "Data preprocessing in semi-supervised SVM classification,"




- Optimization*, vol. 60, no. 1–2, pp. 143–151, Jan. 2011, doi: 10.1080/02331931003692557.
- [34] H. Zhao, C. Yang, W. Guo, L. Zhang, and D. Zhang, “Automatic estimation of crop disease severity levels based on vegetation index normalization,” *Remote Sensing*, vol. 12, no. 12, Jun. 2020, Art. no. 1930, doi: 10.3390/rs12121930.
- [35] R. Bhatla, S. Verma, R. Pandey, and A. Tripathi, “Evolution of extreme rainfall events over Indo-Gangetic plain in changing climate during 1901–2010,” *Journal of Earth System Science*, vol. 128, no. 5, Jul. 2019, Art. no. 120, doi: 10.1007/s12040-019-1162-1.
- [36] Q. Gu, L. Zhu, and Z. Cai, “Evaluation measures of the classification performance of imbalanced data sets,” in *Communications in Computer and Information Science*, vol. 51, 2009, pp. 461–471.
- [37] N. Singh and A. Ranade, “The wet and dry spells across India during 1951–2007,” *Journal of Hydrometeorology*, vol. 11, no. 1, pp. 26–45, Feb. 2010, doi: 10.1175/2009JHM1161.1.

## BIOGRAPHIES OF AUTHORS



**Shilpa Hudnurkar**    is B.E. Instrumentation and has completed an M.Tech. in Electronics and Telecommunication. She is currently working as an Assistant Professor in the Department of Electronics and Telecommunication Engineering at Symbiosis Institute of Technology affiliated to Symbiosis International (Deemed University) and is a research scholar at Symbiosis International (Deemed University). Her research interests include artificial intelligence, machine learning, deep learning, signal processing. She is working on predicting Summer Monsoon Rainfall over a small region. Her teaching experience is over 7 years. She can be contacted at email: shilpa.hudnurkar@sitpune.edu.in.



**Neela Rayavarapu**    received her B.E. degree in Electrical Engineering from Bangalore University, Bangalore, India in 1984, M.S. Degree in Electrical and Computer Engineering from Rutgers, The State University of New Jersey, USA, in 1987, and Ph.D. degree in Electronics and Communication Engineering in 2012 from Panjab University, Chandigarh. She has been involved with teaching and research in electrical, electronics, and communication engineering since 1987. Her areas of interest are digital signal processing and its applications and control systems. She can be contacted at email: neela.raya27@gmail.com.