



# Improved Prediction Analysis with Hybrid Models for Thunderstorm Classification over the Ranchi Region

Kanchan Bala<sup>1</sup> · Sanchita Paul<sup>1</sup> · Sachi Nandan Mohanty<sup>2</sup> · Satyasundara Mahapatra<sup>3</sup>

Received: 1 February 2022 / Accepted: 6 May 2022 / Published online: 6 June 2022  
© Ohmsha, Ltd. and Springer Japan KK, part of Springer Nature 2022

## Abstract

Thunderstorms are natural disasters that impact people, animals, and the economy. Thunderstorms' detrimental repercussions can be avoided by identifying their occurrence in advance. The current work, in this respect, uses soft computing techniques such as K-Nearest Neighbour (KNN), Decision Tree (DT), Logistic Regression (LR), and Support Vector Machine (SVM) with various kernel functions to categorize the occurrence of thunderstorms over Ranchi, India. These techniques were trained and tested using two data sets: daily average and hourly meteorological datasets. The primary purpose of this study is to find which dataset-classifier combination is optimal for categorizing thunderstorm occurrence in Ranchi. No classifier was found to adequately classify either the Day Average Dataset or the Modified Day Average Dataset. On the other hand, the Hourly Dataset was found to be more balanced in terms of the number of thunderstorms that occurred than the Day Average and Modified Average datasets. The F-Score value of the incidence of thunderstorm incidents after using different classifiers was used to compare the outcomes of these datasets. The results reveal that using SVM with radial basis function. The Hourly Dataset is the best for thunderstorm day classification. For the overall and only incidence of thunderstorms classes, SVM-RBF gets 0.81 and 0.74 F-Scores, respectively. Other approaches, like grid search and Bagging, have been used to increase SVM-RBF performance. Grid search and Bagging are used on SVM-RBF to produce a hybrid Grid-Bag-SVM-RBF classifier with 82.04% accuracy and F-scores of 0.83 and 0.78 for overall and just thunderstorm occurrence, respectively.

**Keywords** Bagging · Classification · Decision tree · F-Score · K-Nearest neighbour · Logistic regression · Precision · Radial basis function · Recall · Support vector machine · Thunderstorm

---

✉ Sachi Nandan Mohanty  
Sachinandan09@gmail.com

Extended author information available on the last page of the article

## Introduction

Thunderstorms can be considered one of the most damaging but spectacular weather events that occur almost annually during the pre-monsoon season in India [1]. These are very regular meteorological phenomena occurring at the highest level of the troposphere. Thunderstorms occur when hot, humid air expands into large currents and rises rapidly towards relatively colder atmosphere areas. Humidity in the ascending stream condenses and forms cumulonimbus. The frozen air columns flow to the earth and the electrical charges associated with cloud particles cause the phenomena known as lightning. Lightning further heats the air and generates shock waves, translating into thunder [2]. Consequently, humidity, the lifting mechanism (rising current), and instability are the three most essential factors in the onset of thunderstorms.

Thunderstorms are incredibly wonderful weather events that can happen anywhere on earth. Due to deep convention, these storms are associated with tornadoes, torrential precipitation, high wind gusts, hail, lightning, and advances [3][4]. Thunderstorms have a lifetime of less than an hour or several hours and a spatial extension of a few kilometers [29]. Thunderstorms are typically classified based on their physical characteristics. These thunderstorms are continuous but categorized into single-cell thunderstorms, multicellular thunderstorms, multicellular thunderstorms, and super cellular thunderstorms. Single-celled storms are small and brief weather events that last between 20 and 30 min. They are linked to short heavy rains and low tornadoes. Multicell cluster storms are a general type of thunderstorms. They comprise a group of cells and progress as a single unit with several hours of life. However, individual cells expire within 20 min. These storms produce heavy rains and are stronger than single-celled storms. Multicell line storms are also known as squall lines. They are usually composed of a long line of storms that are well-developed continuous bursts. These thunderstorms produce heavy rain, hail, and small tornadoes. Super cellular thunderstorms are well organized and present a high risk to property and life [5].

The purpose of this work is to determine which type of data set-classifier combination would be more suitable for classifying the incidence of thunderstorms over Ranchi. The grid search and bagging technique increase the efficiency of the resulting dataset-classifier. The grid search lets you define the value of the parameters in the classifiers. It may assist meteorologists and researchers in saving the dataset for analysis. For this purpose, simple classifiers are applied to the Day Average Dataset and find that no classifier adequately classifies the incidence of thunderstorm days (TD), while all classifiers correctly classify the incidence data for combined thunderstorms and non-thunderstorms days (NTD). Therefore, we must remove specific NTD data. The NTD data have been removed in such a way that the dataset contains NTD data that is 7–8 days before the incidence of the TD and obtains the data set on the changed Day Average Dataset. So the elimination of NTD has been just like random sampling. These NTD data are removed so that the Dataset becomes more balanced. NTD days occurred between one and five days before TD's incidence. Thus, these 500 NTD days are randomly removed from Day Average Dataset. The same

classifiers are again applied to the Modified Day Average Dataset. However, they have still not obtained the appropriate outcome for TD cases, but more satisfactory result than Day average Dataset. The Average Dataset has an average value of the whole day, which does not contain any more appropriate information about parameter variation. Hourly atmospheric data were collected, showing a more appropriate hourly variation in parameters than the Day Average and Modified Day Average Datasets. However, the same classifiers are applied to the hourly data set. Now, the results of all these datasets are compared with the F-Score of the incidence of TD cases. The result shows that Hourly Dataset with records of 2 to 3 h before and after the occurrence of thunderstorms have more appropriate for analysis of the incidence of TD. Key points in this manuscript include:

1. Classifiers are applied to the hourly data set and are awarded SVM-RBF as the best classifiers.
2. The selected classifier (SVM-RBF) is optimized using the grid search and bagging technique and gets the optimized classifiers. These resulting classifiers (Grid-SVM-RBF, Bag-SVM-RBF) show better performance than other applied single classifier techniques.
3. Most authors only demonstrate the accuracy of storm forecasts or overall accuracy. They did not include distinct performance precision, recall, and F-Score values for TD, NTD, and overall cases. This manuscript contains all these performance measures.

The constraints on the statistical method [6] and the limitations of the numerical model [7] have led to the introduction of soft computing techniques in thunderstorm prediction. Several researchers have proposed Soft computing techniques to predict thunderstorms.

ANN with six learning algorithms was introduced. Six learning algorithms are Step, Momentum, Conjugate Gradient, Quick Propagation, Levenberg–Marquardt, and Delta-Bar-Delta. Three years of hourly data were taken with parameters like mean sea level pressure, relative humidity, and wind speed. ANN with Levenberg–Marquardt (LM) learning algorithm was the best nowcast the thunderstorm with ahead of 1 h to 24 h that is helpful for weather forecaster [3].

Several authors have proposed soft computing techniques in thunderstorm prediction and forecasting. The preferred type for generating violent thunderstorms has been identified. Low-level cloud patterns were recognized using the Rough Set (RS) of soft computing technique. These types of clouds were found to predict thunderstorms and non-thunderstorms. The result has been demonstrated that low-level cumulonimbus clouds are preferred for storms. The result also reveals that the formation of cumulonimbus clouds at 06 GMT is a more favourable condition for the genesis of thunderstorms in the pre-monsoon over Kolkata [6].

Tornadoes, thunderstorms, and severe thunderstorms may now be predicted using meteorological data thanks to the use of a decision tree. The decision tree has been designed to anticipate no thunderstorms, thunderstorms with local floods, thunderstorms with dry microbursts, thunderstorms with wet microbursts, or downbursts,

severe thunderstorms with downbursts or tornadoes, and many other scenarios. It employs a physical-reasoning-based design. As a result, it can be used for predicting and forecasting in a variety of countries [8]. The quantitative range of Convective Inhibition Energy (CINE) and Convective Available Potential Energy (CAPE) for severe thunderstorms was depicted using fuzzy logic and statistics methods. A statistical-fuzzy mixed-method suggests that a CINE range of 0–150 J kg<sup>-1</sup> is acceptable. The surface temperature, surface mixing ratio, and altitude of the level of free convection all have a role in determining CINE. The most crucial criterion for thunderstorms over Kolkata during the pre-monsoon season is the surface temperature, which should be between 30 and 38 degrees Celsius [9].

In Kolkata, researchers looked into using an Artificial Neural Network (ANN) using a back-propagation method to predict thunderstorms and lightning [1, 10]. Pressure, dew-point temperature, and wind speed at lifting condensation level are used to calculate the parameters. Data has been collected at one or more of the important pressure levels: roughly 500 hpa, 600 hpa, 700 hpa, 850 hpa, and 1000 hpa. An ANN model with one hidden layer and a variable learning rate back-propagation technique was used to forecast severe thunderstorms.

To forecast severe thunderstorms, researchers utilized a variety of machine learning models, including the KNN, modified KNN, and multi-layer perceptron (MLP). Predictors are dry adiabatic lapse rate and moisture difference with various geopotential heights of the atmosphere. The modified K-NN technique offers extremely favourable predictive information with excellent prediction accuracy for both no storm and squall-storm scenarios. The predictors are calculated using radiosonde data collected in the morning to compute the conditional instability and humidity of the atmosphere. The humidity measurement from the surface to the higher atmosphere is known as the vertical moisture difference. The early morning (00:00UTC) atmosphere's upper air humidity at 600 hpa and 850 hpa, as well as conditional instability from 700 to 300 hpa, are critical criteria for predicting evening squall-storms. Thus, lapse rate and humidity, two upper air morning characteristics, play a vital role in forming thunderclouds starting in the early morning [11].

Back Propagation Neural Network (BPN), PCA, and Self Organizing Map (SOM) approaches were developed for quantitative forecasting of CB (cumulonimbus) cloud. The accuracy of the outcome was greater with BPN-PCA than with just BPN. Cumulonimbus clouds may produce deadly storms, including thunder, lightning, and ice. BPN is more efficient than numerical differentiation for estimating a large class of functions. Raw sounding observation data and generated indices values were utilized as inputs for BPN. PCA has been used to conduct feature selection without sacrificing information. Clustering and initialization of neural networks have been done by SOM [12].

The Naive Bayesian (NB) model was used to estimate the likelihood of a storm turning severe. The NB model uses a glaciation rate, cloud object vertical and horizontal expansion, as well as numerical weather prediction fields (CAPE, effective shear) to estimate the likelihood of a severe storm. The NB model was trained using 864 non-severe thunderstorms and 120 severe thunderstorms. The findings demonstrate that the pace of expansion of satellite objects and the surrounding environment can assist in distinguishing between severe and non-severe [13].

Three models, RS, Support Vector Machine (SVM), and RS-SVM were used to forecast the thunderstorm on a short-term and small-scale basis. The thunderstorm was classified using SVM with a radial basis function (RBF) kernel. The SVM classifier has a greater forecast accuracy than the RS model. The RS-SVM model outperforms the SVM model with a forecast accuracy of 71%. The prediction of thunderstorms was made using a real dataset [14].

K-Nearest Neighbor (KNN) was designed to predict the thunderstorm with a 12 h lead time. Eight types of upper-air weather parameters namely, Sunshine hour (SSH), Cloud coverage (Nh), Pressure at freezing point (FRZ), adiabatic lapse rate at four different geopotential heights [19].

Genetic programming (GP) has been proposed, which was implemented in WEKA. Performance was measured in terms of precision, recall, and f-measure value for the imbalanced data set. Total 31 different attributes were taken to predict thunderstorms [12].

ANN-MLP was proposed. Moisture difference, wind shear, adiabatic lapse rate as parameters at different geopotential heights from IMD of 18 years data. ANN-MLP structure has a minimum misclassification rate for other MLP structures [20].

Various deep learning algorithms have been used to predict thunderstorm gales. Convolution neural network (CNN), a time context recurrent convolutional neural network (TRCNN), a recurrent neural network (S-RCNN), and a spatio-temporal recurrent convolutional neural network (ST-RCNN) were used to solve the thunderstorm prediction problem. The radar echo images were used to create the Dataset. The wind velocity recorded by meteorological stations was used to partially label each image. The results were compared with 10 different machine-learning techniques. There are no features selection was done [38].

A hybridization of pre-processing data ensemble empirical mode decomposition (EEMD) with two models, namely SVM and ANN, has been presented to forecast thunderstorms over Bangladesh. Thunderstorm frequency months were classified as high (March–June), moderate (July–October), and low (November–February) over the period 1981–2016. Simple classifiers such as SVM, ANN, and autoregressive integrated moving average were compared to the performance of the proposed EEMD-SVM and EEMD-ANN hybrid models. When compared to previous models, the EEMD-SVM and EEMD-ANN hybrid models showed an increase in performance accuracy of 8.02–22.48 percent. The random forest variable importance analysis approach [39] was used to choose 11 of the 21 input parameters.

A deep-learning neural network (DLNN) model was developed to forecast thunderstorms within 400 km<sup>2</sup> 15 h ahead of time (with a 2 h accuracy). The numerical weather prediction model's output parameters/variables were utilized as input features, and cloud-to-ground lightning was employed as the goal. For the development of higher-order representations of the features, the Stacked De-noising Auto Encoder (SDAE) was utilized. To train the prediction model, LR was applied to the SDAE output. Iterative strategies were used to optimize the SDAE architecture. The improved DLNN outperforms shallow neural network models in terms of performance [40].

With numerical weather prediction (NWP) data, a deep learning approach was used to predict severe convective weather (SCW) such as hail, heavy rain,

thunderstorms, and convective gusts. Five years of NCEP, final (FNL) analysis data served as the training datasets. Each sort of meteorological occurrence was practiced using the identified samples. The pressure, temperature, winds, and humidity, as well as dozens of convective physical parameters, were used as predictors. To train for prediction, a six-layer convolutional neural network (CNN) model was deployed. CNN's performance was compared to that of traditional approaches. The results demonstrate that the deep learning algorithm outperformed other standard methods in terms of classifier [41].

In a prior study, the author proposed a heuristic model based on the same Hourly Dataset. A heuristic equation was incorporated in the heuristic model, which uses the correlation coefficient of meteorological parameters with the number of hourly thunderstorm incidences. The provided heuristic equation was used to calculate the values of four indices. All of these indices' values are used to classify thunderstorms on an hourly basis. In the month-wise procedure, the first index value was produced using normalized average values of parameters of just hourly incidence of thunderstorms data. The other three indices were derived from optimization techniques, namely, simulated annealing (SA), teaching learning based optimization (TLBO) technique, and differential evaluation (DE). TLBO depicts better classifiers for hourly incidences of thunderstorms [42].

Various soft computational and data mining techniques have been summarized for thunderstorm prediction [15]. Soft computational techniques have been effectively used for prediction in another area of research [16, 17, 18]. This manuscript was used as a hybrid classifier on an hourly data set to classify thunderstorm occurrence and non-occurrence. Two data sets, namely the Day Average and the Hourly Dataset are used. Simple classifiers (LR DT KNN SVM with different kernels) are applied to these two data sets. The Day Average Dataset was not ranked higher for TD cases. The Day Average Dataset is modified using random sampling to improve accuracy. These datasets (Day Average, Modified Day Average, and Hourly dataset) and applied classifiers get the best dataset classifier and study the applicable grid search used with bagging techniques. Therefore obtaining the best hybrid classifier will improve the accuracy of thunderstorm incidence classification. Thus, the obtained hybrid classifiers achieved an accuracy of 82%. In this regard, the whole paper is organized as follows: Sect. [Introduction](#) describes the opening section with literature comments. Section [Materials and Methods](#) presents the research area. Section [Experiment and Result](#) presents methods and materials. The results and discussion are devoted to Sect. [Conclusion](#). Conclusions are discussed in Sect. 5. In addition, in the end, references are included in Sect. 6.

## Materials and Methods

### Thunderstorm Dataset Description

#### Day Average Dataset

The data used to support this study's findings is average meteorological data with station number 42701 which has station latitude (SLAT) 23.31, longitude 85.31, and elevation 652 Ranchi, India. A total of 10 years of data from April to September of 2008 to 2017 has been collected. Classification of incidence of TD has been done only on pre-monsoon data. Dataset has 1814 instances and 14 numbers of attributes, namely, average temperature (T), maximum temperature (TM), minimum temperature (Tm), atmospheric pressure at sea level (SLP), average relative humidity (H), total rainfall, and/or snowmelt (PP), maximum sustain wind speed (VM), average visibility (VV), average wind speed (V), the maximum speed of the wind (VG), the day it snowed (SN), the day it rained (RA), the day it thunderstorm (TS), the day it fogged (FG). In the Dataset, five features (FG, RA, SN, VG, and PP) have been ignored due to irrelevant and unavailability of feature values. The segregation of the Dataset into two distinct classes where class value 0 is interpreted as non- thunderstorm day (NTD) and 1 as thunderstorm day (TD) is as shown in Table 2.

Table 1 shows the 550 incidences of thunderstorms and 1264 non-incidences of thunderstorms in the average Dataset. Figure 1 depicts the month-wise distribution of thunderstorm days during pre-monsoon and monsoon from 2008 to 2017. April, May, June, July, August, and September months have 53, 73, 107, 113, 106, 98 number of incidence of thunderstorms. As clearly observed from Fig. 1, the monsoon season experiences more thunderstorm days as compared to the pre-monsoon season.

#### Modified Day Average Dataset

This dataset is obtained from Day Average Dataset. The Day Average Dataset is modified by the removal of some samples of NTD. Therefore, we must remove certain NTD data. The NTD data have been removed in such a way that the dataset contains NTD data that is 7–8 days before the incidence of the TD and obtains the data set on the changed day average. So the elimination of NTD has been just like random sampling.

**Table 1** Distribution of Dataset

Sr.no	Name data	No. of NTD	No. of TD	Total
1	Average dataset	1264	550	1814
2	Modified average data set	655	550	1205
3	Hourly dataset (pre-monsoon)	539	336	875

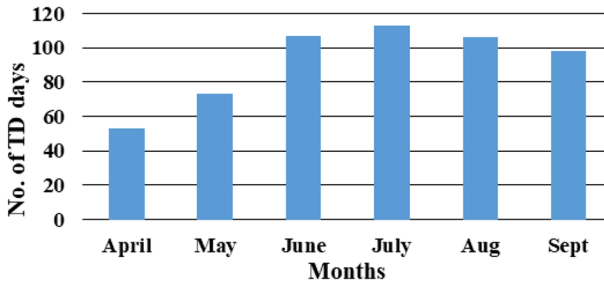


Fig. 1 Number of thunderstorm variations during pre-monsoon and monsoon

### Hourly Dataset

Three years of hourly data from April to June in the duration of 2016 to 2018 has been used to classify the incidence of thunderstorms. Data have been collected in a way that TD and NTD data include 2 to 3 h before the incidence of thunderstorms. In pre-monsoon months, data for rain has not been included. Hourly meteorological data have five sea level pressure, temperature, humidity, wind direction, and wind speed parameters.

### Data Acquisition

#### Day Average Dataset

The meteorological dataset has been obtained from <https://en.tutiempo.net/climate/ws-427010.html>. This link is for a Ranchi weather station with station number 42401. It has 14 features, and some of the values of the features are not available. Some of the features are not relevant regarding the incidence of thunderstorms. All these feature descriptions are mentioned in the data processing of Sect. [Materials and Methods](#).

#### Hourly Dataset

This study includes pre-monsoon hourly meteorological data and records of thunderstorms over Ranchi from 2016 to 2018 were collected from <http://en.tutiempo.net/records/verc>. This Hourly dataset has five relevant features. Wind direction has not been included due to non-numeric values. Some of the samples are rain data. These rain samples are not included in the classification of TD and NTD cases.

### Data Processing

#### Pre-processing Average Data

No feature selection was done but irrelevant features and missing value features of the instance were removed from the Dataset. In the Dataset five features FG, VG,



RA, SN, and PP were either irrelevant or no data was available. Tm, TM, T, H, SLP, VV, VM, and V were more relevant attributes in the occurrence of thunderstorms. These selected features have attributed value to output variable TS fall in either class value TD (1) or NTD (0).

### Pre-processing Hourly Data

In hourly meteorological data, missing values of parameters are removed from the dataset. Rain data are also removed from the dataset, and only the TD and NTD data have been considered.

### Normalization

Processed Day Average Dataset and Hourly Dataset data are normalized using the following formula.

$$\text{Parameter}_{\text{normalization}} = \frac{x_{\text{value}} - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \quad (1)$$

$x_{\text{value}}$  is the value of the parameter,  $x_{\text{max}}$  and  $x_{\text{min}}$  are the maximum and minimum value of parameters respectively.

### Classifiers

Several classification techniques have been applied on Day Average and Hourly Dataset. This section briefly describes the predictive classifiers models as below.

#### K- Nearest Neighbour

K-NN was proposed by Fix and Hodges [21] and modified by Cover and Hart [22]. The K-NN algorithm is a non-parametric technique [23] and instance-based learning used in regression and classification. K-NN is among the top 10 data mining algorithms [24]. Therefore, K-NN has been widely studied and applied in various areas [25]. It has been used in many pattern classification problems such as event recognition [26], ranking model [27], Object recognition [3], and pattern recognition [28] application.

#### Decision Tree

The Decision tree is a decision support tool that uses the model of the decision or a tree-like graph and contains conditional control statements. The decision tree is also referred to as Classification and Regression Trees (CART). Decision criteria are different for CART. The decision tree is a supervised learning algorithm. It creates a training model used to predict values or class of a target variable using the learning decision rule. The decision tree is used for classification and prediction purposes [20].

## Logistic Regression

Logistic regression was developed by David Cox in 1958. It is a supervised classification algorithm. The binary logistic regression is used to evaluate the probability of a binary response based on one or more variables (or independently) predictor (features). The logistic model is one of the most prominent machine learning algorithms for binary classification. The logistic function also called the sigmoid function is an S-shaped curve with real-valued numbers in between 0 and 1, but never exactly at those limits [30].

## Support Vector Machine

Vladimir N. Vapnik firstly proposed SVM in 1995 [31]. SVM is a supervised learning strategy that acts on the discovery of hyperplanes and uses an interclass distance or margin width to distinguish between positive and negative data. Cost a co-efficient factor of  $C_+$  and  $C_-$  denoted as ' $J$ ' which is contributed in the generation of error. Both negative and positive samples can outweigh this. Thus, the optimization of SVM is as below.

$$\text{minimize } \frac{1}{2} \|w\|^2 + C_+ + \sum_{i|y_i=1} N_i + C_- \sum_{j|y_j=-1} N_j \quad (2)$$

Which satisfied the condition  $y_k(wx_k + b) \geq 1 - N_k, N_k \geq 0$

SVM can be performed in both linear and non-linear modes, with the non-linear version or Radial bias kernel being used for non-linearly separable data with a lagrangian multiplier  $\alpha_i$ . So, the optimization problem can be stated as below.

$$\text{minimize } w(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \alpha_i \sum_{i=1, j=1}^l y_i \alpha_j y_j K(x_i, y_j) \quad (3)$$

where,

$$C \geq \alpha_i \geq \forall_i, \sum_{i=1}^l \alpha_i y_i = 0$$

SVMs are supervised machine learning models or approaches that are linked to learning algorithms that examine data, recognize patterns, and are used for classification and regression analysis. SVM creates a hyperplane or decision surface that classifies data with the greatest margin. The generalization error will be minimized by the decision surface that maximizes the margin of the training set. SVM categorizes linearly separable data. In the present study, SVM with the different kernels, namely, Polynomial Kernel, Linear kernel, RBF Kernel have been used for the classification of thunderstorms.

### Linear Kernel SVM

Linear kernel SVM is used to separate into two classes that belong to either side of the margin of the plane. The training sample is in the form of as given below.

$$T = \{(x_1, y_1), (x_2, y_2) \dots \dots \dots (x_n, y_n)\} \tag{4}$$

Where  $y_n$  is either -1 or 1 and denotes class a point  $x_n$  belongs to it,  $n$  is data sample.

Each  $x_n$  is either a  $p$ -dimensional real vector or a collection of training tuples with associated class labels  $y_n$ . The SVM classifier converts the input vectors into a decision value before classifying them using an appropriate threshold value.

To see the training data, we divide or separate the hyperplane. The hyperplane is defined in terms of the  $w$  weight vector and  $b$  scalar. which is defined as

$$\text{Mapping } w^T \cdot x + b = 0 \tag{5}$$

The vector  $w$  is perpendicular to the hyperplane separating them. The problem of determining the best hyperplane among a set of separating hyperplanes can be solved using the maximal marginal hyperplane. The offset parameter  $b$  allows the margin to be increased. Given that the training data is linearly separable, we build hyperplanes and aim to optimize the distance between them. Because the distance between two hyperplanes equals  $2/|w|$ , we must minimize  $w$  by ensuring that for any  $i$  either

$$w \cdot x_i \geq 1 \text{ or } w \cdot x_i \leq -1 \tag{6}$$

### Polynomial Kernel SVM

SVM cannot perform classification jobs when the data is non-linear. To address this constraint, these support vectors are translated into a higher dimensional feature space via kernel functions. The training points that are closest to the separation function are referred to as support vectors in this context. A kernel is a similarity function provided to a machine-learning system by a domain expert. The kernel creates linear models in nonlinear environments. Kernels are used to convert non-separable problems into separable problems and to translate data into a better representational space. A Kernel function is defined as a function in some enlarged feature space that corresponds to the dot product of two feature vectors:

The polynomial kernel is a non-linear kernel with gamma  $\gamma$ , degree  $d$  parameters. It is well suited for the problem with normalized training data. For degree  $d$ , the polynomial function is defined as below.

$$K(x_i, x_j) = (x_i^T x_j + C)^d \tag{7}$$

$$K(x_i, x_j) = (D(x_i), D(x_j)) \tag{8}$$

Equation of polynomial varies according to the value of  $d$ . As for  $d=2$ , polynomial becomes quadratic, which is given below.

$$K(x, y) = \left( \sum_{i=1}^n x_i y_i + C \right)^2 = \sum_{i=1}^n (x_i)^2 (y_i)^2 + \sum_{i=2}^n \sum_{j=1}^{i-1} (\sqrt{2} x_i x_j) (\sqrt{2} y_i y_j) + \sum_{i=1}^n \sqrt{2} C x_i (\sqrt{2} C y_i) + C \quad (9)$$

Feature mapping is derived from above as in the following equation.

$$D(x) = \left( \begin{array}{c} x_n^2 \dots, \sqrt{2} x_n x_{n-1}, \dots, \sqrt{2} x_n x_1, \sqrt{2} x_{n-1} x_{n-2}, \dots, \\ \sqrt{2} x_{n-1} x_{1}, \dots, \sqrt{2} x_2 x_1, \sqrt{2} C x_n, \sqrt{2} C x_n C \end{array} \right) \quad (10)$$

### Radial basis function

RBF kernel is also called the Gaussian Kernel. RBF kernel with SVM is non-linear and issued to analyze the data in higher dimensions. The output of the RBF kernel is Euclidean distance between two features vector  $x_m$  and  $x_n$  that is defined as below.

$$K(x_m, x_n) = \exp \left( - \frac{\|x_m - x_n\|^2}{2\sigma^2} \right) \quad (11)$$

where,

$\|x_m - x_n\|^2$  is squared Euclidean distance of two feature vectors,  $\sigma$  is denoted window width or free parameters with

$$\gamma = \frac{1}{2\sigma^2}.$$

### Grid Search SVM

SVM recognizes much better for considering numerical features and high dimensional datasets. Although SVM performs well with the default values, its performance can be enhanced significantly using parameter optimization. The grid search is applied to SVM parameters and locates near-optimal parameter combinations within the given ranges. However, grid search is very slow; therefore, it is reliable only in low-dimensional datasets with few parameters. In SVM, Only one parameter  $C$  of the linear kernel to optimize. There are two parameters  $C$  and  $\gamma$  in the RBF kernel to optimize, while the polynomial kernel has three parameters needed to optimize. Grid search takes a huge amount of time if we select many steps and parameters to optimize. The main problem with SVM parameter optimization is that it has no precise ranges of  $C$  and  $\gamma$  values. It is believed that there are more possibilities in the grid search method for finding the best combination parameter with the broader

range of parameter ranges [32]. Figure 2 shows the working of grid search. In our case, we have been taken the range of C and Y from 0.001 to 10,000.

### Bagging SVM

It is an acronym for Bootstrapping Aggregation, which is the type of ensemble method. It was proposed by Breiman [33]. It comprises a bag of classifiers that are trained. In Bagging, final results are obtained with the combined output of each of the classifiers (vote). The accuracy of obtained bag classifier is achieved to be better than the individual classifier [34]. In bagging, a classifier can be any model. The SVM is used as the base classifier to improve the accuracy. The bagged SVM is widely applied in predictive tasks [35].

### The Proposed Method

Two databases were used to rank the incidence and absence of TD using predictive models or classification techniques. An average Ranchi weather dataset has fourteen characteristics: T, TM, Tm, SLP, H, PP, VV, V, VM, VG, RA, SN, FG, and TS.

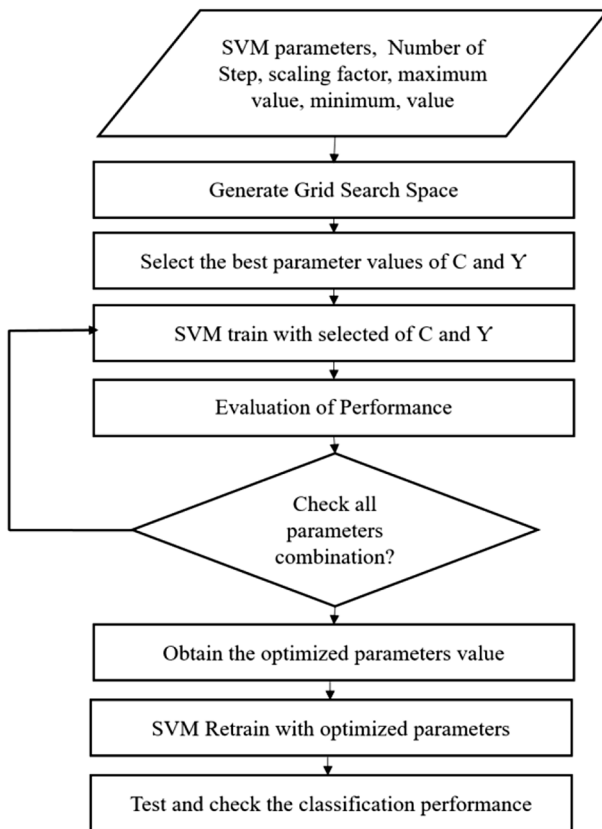


Fig. 2 Working of grid search SVM classifier

Five features RA, PP, FG, VG, and SN were omitted due to unavailable or irrelevant data. The missing data in the dataset were removed. The processed data set was normalized. The normalized data set was split at 70% and 30% into training and test datasets, respectively. Thus, the data of 7 years (2008–2014) is used for training purposes, the remaining 3 years (2015–2017) is used for testing.

Various prediction techniques such as LR, KNN, DT, and SVM with radial basis function (RBF), polynomial and linear kernels are then applied to the training and testing datasets to classify incidence TD. Then, different performance measures such as precision, recall, F-Score values were used to evaluate the classification of incidence TD and NTD with predictive models/classifiers different guesses. F-score's highest value classifier is considered to be the best classifier for classifying the incidence and absence of TD, The Day Average Dataset is obtained by the very low value of F-Score for the incidence of TD and reaches high value for NTDs. Therefore, the Day Average Dataset needs to be altered to properly classify the incidence of TD. Therefore, the data set was changed by removing some NTD days. All classification techniques were again applied to the Modified Day Average Dataset, and performance measures were verified for all classification techniques. The same predictive classification techniques are also applied to the pre-monsoon Hourly Dataset. Thus, the results of the Day Average Dataset, Modified Day Average and Hourly Dataset are compared and found the best classifier with the best Dataset. Grid search and bagging techniques were applied on the best classifier-dataset to improve the accuracy of TD incidence classification. Thus, a hybrid classifier (Grid-Bag-SVM-RBF) was obtained to classify the incidence of TD.

## Experiment and Result

All coding is done in python. This study uses two data sets: the Day Average Dataset and the Hourly Dataset. In the Day Average Dataset the parameters have the Day Average while the Hourly Dataset has the hourly value of the parameter. The Day Average Dataset does not perform well for TD incidents. The Day Average Dataset should be modified by randomly removing the sample from the majority class to improve performance. The Hourly Dataset collected data few hours before the start of the TD. Both the daily mean data set and the hourly data set were compared and which Dataset was most appropriate to classify the incidence of TD.

Since the Day Average Dataset has an imbalance in the focus class, i.e. the number of incidence of TD, is less compared to NTD. The most widely used observations for measuring accuracy irrespective of the number of correct labels of different classes.

Classifier Accuracy is not only for performance metrics but also includes precision, recall, and F-Score in performance metrics for classifiers. Precision and recall are inversely related where it is possible to increase the cost of one at the expense of reducing the other. An alternative to recall and precision is F-Score, which combines them into a single performance measure.

The dataset had to be altered because the F-Score obtained by all classifiers for the rating of thunderstorm days was unsatisfactory for the Day Average Dataset. The

recall values for thunderstorm classification are also not very good. Some non-occurrence of thunderstorms have been removed from the dataset to better classify TD and NTD. One of the important subsampling methods is the random subsampling method, which balances the distribution of the class by randomly removing the sample from the majority class sample [36, 43]. The NTD data have been removed in such a way that dataset contain NTD data that are 7-8 days before the incidence of the TD, and obtains the data set on the changed Day Average Dataset. Thus, suppressing NTD as random subsampling. All of the classification techniques listed above were re-applied to the Modified Day Average Dataset and the F-Score values of all classifiers for the verified thunderstorm day. The authors considered the classifier to have the highest F-Score value. This F-Score value must be greater than 0.60 for a meaningful classification of the TD scale. A classifier with such F-Score value is considered a good classifier. The rule-based association classifier (ARCID) was applied to five unbalanced datasets to obtain a balanced dataset. ARCID has three phases which are selecting, generating, and filtering rules. The first phase is the generation of rules from each class of the training set The second phase is filtrations of rules generated during the first phase and then selecting the rules. The F-Score value obtained by these datasets is up to 0.67 [37]. The objective of this algorithm is to extract significant knowledge from imbalanced datasets by highlighting the information removed from major classes without significantly affecting the predictive accuracy of the classifier.

### Performance Measure

The performance of classification techniques has been evaluated on the basis of the confusion matrix on a set of test data. The confusion matrix has four terms, namely, True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). According to these terms, various performance measures, accuracy, recall, precision, and F-Score are defined.

TP: Thunderstorm day is correctly predicted as thunderstorm day.

FP: Non-thunderstorm day is incorrectly predicted as thunderstorm day.

TN: Non-thunderstorm day is correctly predicted as non-thunderstorm day.

FN: Thunderstorm day is incorrectly predicted as a non-thunderstorm day.

$$\text{Classification Accuracy} = (TP + TN)/(TP + FP + TN + FN) \quad (12)$$

$$\text{Precision} = TP/(TP + FP) \quad (13)$$

$$\text{Recall} = TP/(TP + FN) \quad (14)$$

$$\text{F-Score} = (2 \times \text{Precision} \times \text{Recall})/(\text{Precision} + \text{Recall}) \quad (15)$$

F-Score reflects both values of precision and recall. Precision reflects how well accurate classifiers are predicted. As a result, precision computes the real positive as

a percentage of the total positive. Precision indicates how many of those categorized as thunderstorms are actually thunderstorms. Recall displays how many thunderstorms were properly predicted out of the total number of days with thunderstorms.

## Result and Discussion

Table 2 primarily displays the training and testing of different classifiers' accuracy for aggregate TD and NTD days categorization. Training accuracy reveals model-built accuracy, whereas testing accuracy demonstrates model testing using a dataset. SVM-RBF outperformed all other classifiers regarding training and testing accuracy, achieving 77.22 percent and 75.91 percent, respectively for Day Average Dataset. Again, SVM-RBF outperforms all classifiers in Modified Day Average and Hourly Dataset.

Table 3 displays the performance of the prediction models KNN, LR, DT, SVM-RBF, SVM-poly, and SVM-Linear on the Day Average Dataset to categorize TD and NTD. When measuring classifier performance in terms of precision, recall, and F-Score, SVM-RBF once again outperforms all other classifiers. Thus, following SVM-RBF, LR and SVM-Linear are the second and third choices in classifiers for overall classification (TD+NTD) for Day Average Dataset. All classifiers have greater than 60% overall precision, recall, and F-Score in overall performance.

Table 3 also illustrates how the classifiers perform separately in each TD and NTD day class. For NTD days, classifiers performed well in precision, recall, and F-Score. Precision, recall, and F-Score is more than 0.70 out of 1 for NTD days. The classifiers' performance for TD days, with recall and F-Score values, is insufficient. Although classifiers produce good results for NTD days, they fail to categorize correctly for TD incidences. Thus, Table 3 shows that thunderstorm class occurrences are not suitably classified for the Day Average Dataset. As a result, the same Day Average Dataset was made more balanced using the random sampling approach to increase classifier performance. The same classifiers were applied to the Modified Day Average Dataset. Thus, Table 4 shows the result for Modified Day Average Dataset.

The overall (TD+NTD), NTD, and TD incidence classifications for the Modified Day Average Dataset were satisfactory, as shown in Tables 4. The majority of the classifiers had accuracy, precision, recall, and F-Score values of more than 60%. DT, KNN, and SVM-Poly have been unable to classify TD incidents. These classifiers have an F-Score value of less than 0.55 out of 1. SVM-RBF again shows the best performer in TD classification. None of any classifiers works in TD classification for Day Average Dataset, but the same classifiers work well in TD classification for Modified Day Average Dataset. This classification performance increases due to the removal of some NTD data from Day Average Dataset and gets Modified Day Average Dataset.

Therefore, we must remove certain NTD data. The NTD data remove in such a way that these are retained for 7–8 days before the incidence of TDs and obtains the data set on the changed Day Average Dataset. Therefore, the elimination of NTD



has been just like random sampling. As a result, Modified Day Average data sets outperform typical Day Average Datasets.

For thunderstorm classification, a three-year Hourly Dataset containing data recordings from 2 to 3 h before and after the onset of the thunderstorm was used. Predictive classifier approaches are used on Hourly Datasets, and the results are reported in Table 5. Table 5 demonstrates that SVM-RBF has the greatest overall performance of all classifiers. SVM-RBF outperforms all other classifiers yet again. Table 5 also depicts the categorization performance of thunderstorm incidence.

Figure 3 shows the training and testing accuracy for the Day Average Dataset, Modified Day Average Dataset, and the Hourly Dataset, respectively. SVM-RBF (Tr) represents the training of SVM-RBF classifiers and SVM-RBF (Te) represents the testing of classifiers with three datasets. Similarly, other classifiers represent training and testing the same as SVM-RBF. In Fig. 3, line of Hourly Dataset line above the other two lines of Day average Dataset and Modified Day average Dataset for all classifiers. The SVM-RBF–Hourly Dataset combination performs best for training and testing accuracy. Therefore, Hourly Dataset is best trained and tested with classifiers.

Figures 4 illustrate the precision, recall, and F-Score values obtained by predictive classifiers on various datasets for TD classification. Precision recall and F-Score in the x-axis of Fig. 4 are performance measures for Day Average Dataset. M\_precision, M\_recall, and M\_F-score are performance measures for Modified Day Average Dataset. H\_precision, H\_recall, and H\_F-Score are performance measures for Hourly Dataset. Thus, the first, second, and third curves in Fig. 4 indicate performance measures for Day Average Dataset, Modified Day Average Dataset, and Hourly Dataset respectively. The first curve is lower than the second curve; this means that performance measures of all classifiers for Day Average Dataset are lower than Modified Day Average Dataset except precision value for SVM-Poly and SVM-RBF classifiers. The second curve is lower than the third one, which indicates

**Table 2** Training and testing accuracy for all datasets

Model	Class	Day average dataset(%)	Modified day Average Dataset(%)	Hourly dataset(%)
SVM-linear	Training accuracy	74.72%	74.43%	75.58%
	Testing accuracy	74.26%	68.50%	79.40%
SVM-RBF	Training accuracy	77.22%	74.46%	78.70%
	Testing accuracy	75.91%	70.71%	80.89%
SVM-poly	Training accuracy	73.52%	66.98%	70.70%
	Testing accuracy	71.87%	60.22%	75.93%
KNN	Training accuracy	76.04%	71.37%	78.13%
	Testing accuracy	72.42%	62.98%	78.09%
LR	Training accuracy	75.96%	73.87%	76.43%
	Testing accuracy	74.81%	69.88%	80.64%
DT	Training accuracy	73.20%	64.01%	75.04%
	Testing accuracy	70.03%	57.45%	76.10%

**Table 3** Performance analysis for day average dataset

Model	Class	Precision	Recall	F-Score
SVM-linear	NTD	0.75	0.95	0.84
	TD	0.71	0.27	0.39
	TD+NTD	0.74	0.74	0.69
SVM-RBF	NTD	0.80	0.76	0.70
	TD	0.93	0.23	0.37
	TD+NTD	0.80	0.76	0.72
SVM-poly	NTD	0.71	0.99	0.85
	TD	0.89	0.10	0.17
	TD+NTD	0.77	0.72	0.63
KNN	NTD	0.73	0.95	0.83
	TD	0.65	0.22	0.33
	TD+NTD	0.71	0.72	0.67
LR	NTD	0.76	0.92	0.84
	TD	0.67	0.35	0.46
	TD+NTD	0.73	0.75	0.70
DT	NTD	0.70	0.98	0.82
	TD	0.62	0.06	0.11
	TD+NTD	0.68	0.70	0.60

that Modified Day Average Dataset has a lower performance measure than Hourly Dataset for all classifiers in TD classification. Thus, Hourly Dataset has far better than the other two datasets for all classifiers in the classification of TD. The pre-monsoon Hourly Dataset is depicted in this figure, and SVM-RBF was determined to be the best dataset-classifier combination over Ranchi in terms of recall value and F-Score. Comparisons study are based on F-Score measure, which indicates that the pre-monsoon Hourly Dataset is the best in all datasets, and SVM-RBF is the best predictive classifier in all classifiers for three datasets. We have already observed the performance measure of different classifiers for TD incidents.

Figure 5 display the overall TD+NTD performance measure of different classifiers for three datasets. This figure also shows the performance measure for the hybrid classifier. Precision, recall, and F-score of different classifiers for Modified Day Average Dataset have lower values than Day Average Dataset as in Fig. 5. Thus all lines in Fig. 5 go down for Modified Day Average Dataset. Lines go up for the Hourly Dataset, which indicates a high-performance measure for Hourly Dataset. Thus, Hourly Dataset has a higher performance than the other two datasets (Day Average and Modified Day average dataset). Now we find out which classifier is best for TD+NTD classification. SVM-RBF has the highest precision, recall, and F-Score value in all classifiers. As a result, SVM-RBF obtains F-Score values of 0.81 and 0.74 for total (TD+NTD) and TD occurrence respectively. Thus, the performance measure recall displays SVM-RBF as the top predicted classifiers across the pre-monsoon hourly dataset, with 0.81 and 0.74 for overall and TD days' classification, respectively.

Based on the description above, SVM-RBF with hourly dataset combination is the best dataset-classifier combination for all three datasets. Grid search

**Table 4** Performance analysis for modified day average dataset

Model	Class	Precision	Recall	F-Score
SVM-linear	NTD	0.68	0.81	0.74
	TD	0.73	0.51	0.60
	TD+NTD	0.70	0.69	0.67
SVM-RBF	NTD	0.65	0.85	0.70
	TD	0.75	0.58	0.66
	TD+NTD	0.70	0.70	0.69
SVM-poly	NTD	0.57	0.95	0.71
	TD	0.79	0.22	0.35
	TD+NTD	0.68	0.6	0.54
KNN	NTD	0.61	0.85	0.71
	TD	0.70	0.39	0.50
	TD+NTD	0.65	0.63	0.61
LR	NTD	0.69	0.80	0.74
	TD	0.73	0.6	0.61
	TD+NTD	0.71	0.71	0.70
DT	NTD	0.56	0.93	0.70
	TD	0.70	0.29	0.29
	TD+NTD	0.63	0.57	0.5

and Bagging are used on obtained the best dataset-classifier combination (Hourly Dataset-SVM-RBF) to produce a hybrid classifier that enhances performance. Grid search improves performance by searching for the optimal parameter value in SVM-RBF. By bagging the SVM-RBF classifier, the bagging method enhances performance. The accuracy of the generated bag classifier is higher than that of the individual classifier [34]. How grid searches and bagging work together to increase performance on SVM-RBF is shown in Fig. 4 and Fig. 5.

In python, SVM-RBF makes use of the default parameter values in the library function SVC () with the RBF kernel. Despite achieving an excellent outcome with an F-Score of 0.74 for the incidence of TD. Grid search has been applied on SVM-RBF to get the Grid-SVM-RBF hybrid classifier. The Grid-SVM-RBF classifier optimizes SVM-RBF parameter values using grid search within a range of 0.001 to 10,000. In grid search, we get  $\gamma=1$  and  $C=10$  using GridSearchCV(). These settings are set in SVC () using an RBF kernel before being trained and tested. Bagging is applied on SVM-RBF to get Bag-SVM-RBF. Grid search and Bagging are applied together to obtain Grid-Bag-SVM-RBF. The performance of the developed hybrid classifier Grid-Bag-SVM-RBF is shown in Table 5.

Table 5 displays the TD, NTD, TD+NTD classification performance of hybrid Grid-SVM-RBF, Bag-SVM-RBF, and Grid-Bag-SVM-RBF classifiers. Grid-Bag-SVM-RBF has the greatest F-Score of 0.83, while Bag-SVM-RBF has the lowest F-Score of 0.80. Grid-Bag-SVM-RBF once again outperforms the competition in terms of testing accuracy, with a score of 82.04%. Training accuracy follows a similar performance trend. As a result, Grid-Bag-SVM-RBF is the best in terms of accuracy (training and testing) and F-Score value. Now we examine the classifier’s performance for the sole occurrence of thunderstorms separately. Table 5 displays the performance of TD and NTD days separately. Even

**Table 5** Performance analysis for hourly dataset

Model	Class	Precision	Recall	F-Score
SVM-linear	NTD	0.83	0.87	0.85
	TD	0.77	0.65	0.70
	TD + NTD	0.79	0.79	0.79
SVM-RBF	NTD	0.81	0.88	0.84
	TD	0.77	0.70	0.74
	TD + NTD	0.81	0.81	0.81
SVM-poly	NTD	0.74	0.95	0.83
	TD	0.84	0.45	0.58
	TD + NTD	0.78	0.76	0.74
KNN	NTD	0.78	0.91	0.84
	TD	0.8	0.59	0.68
	TD + NTD	0.79	0.79	0.78
LR	NTD	0.82	0.89	0.85
	TD	0.78	0.67	0.72
	TD + NTD	0.80	0.81	0.80
DT	NTD	0.78	0.86	0.83
	TD	0.73	0.6	0.66
	TD + NTD	0.76	0.76	0.76
Grid-SVM-RBF	NTD	0.83	0.91	0.87
	TD	0.79	0.70	0.75
	TD + NTD	0.82	0.82	0.82
Bag-SVM-RBF	NTD	0.89	0.79	0.83
	TD	0.70	0.81	0.76
	TD + NTD	0.81	0.79	0.80
Grid-Bag-SVM-RBF	NTD	0.93	0.77	0.84
	TD	0.76	0.89	0.78
	TD + NTD	0.84	0.81	0.83

though Bag-SVM-RBF outperforms Grid-SVM-RBF, Bag-SVM-RBF is a compilation of many SVM-RBF results, whereas Grid-SVM-RBF is the output of a single SVM-RBF with optimized parameter values. Grid-Bag-SVM-RBF outperformed the other two classifiers, Grid-SVM-RBF and Bag-SVM-RBF, in terms of precision. As a result, Grid-Bag-SVM-RBF has the highest F-Score value for the TD class, 0.78. Figures 4 and 5 also show the performance of the hybrid classifiers and compare the performance with other classifiers. These figures clearly indicate that the proposed hybrid classifiers are outperforming.

## Conclusion

The suggested approach can also be used to anticipate other meteorological events, such as rain. In the current study, thunderstorms were classified using the Day Average Dataset and the Hourly Dataset, having 1814 and 875 cases, respectively.

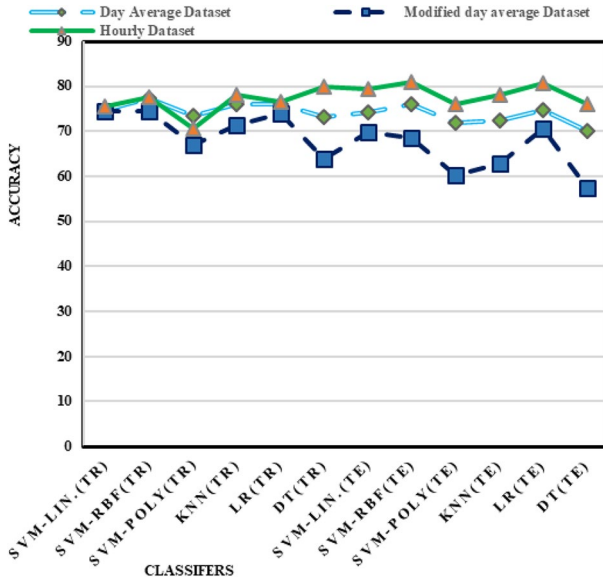


Fig. 3 Training accuracy classifiers for all datasets

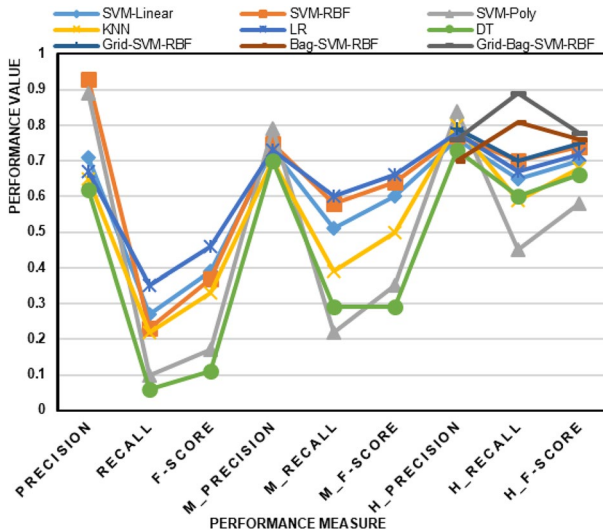


Fig. 4 Performance measure of classifiers for TD in three dataset

Classifiers such as LR, KNN, DT, SVM-Linear, SVM-Poly, and SVM-RBF were employed in this work. SVM-RBF was determined to be the best predictive model for categorizing thunderstorm occurrence days for the Day Average Dataset, Modified Day Average Dataset, and Hourly Dataset. The Hourly dataset has the best performance in all datasets in all datasets.

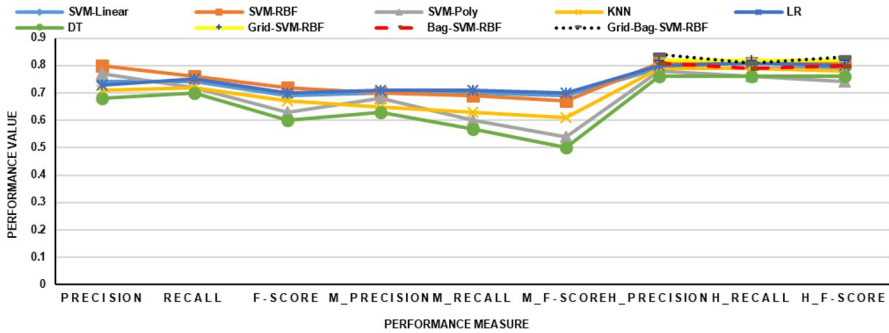


Fig. 5 Performance measure of classifiers for TD +NTD in three dataset

For all three datasets, SVM-RBF outperformed all other classifiers in terms of training and testing accuracy. For all three datasets, all classifiers have higher than 60% overall precision, recall, and F-Score. Although classifiers yield good results for NTD days in the Day Average Dataset, they fail to appropriately classify TD incidents. As a result, utilizing the random sample strategy, the same Day Average Dataset was modified more balanced to improve classifier performance. The Modified Day Average Dataset uses the same classifiers.

Table 4 displays the result for the Modified Day Average Dataset. The Modified Day Average Dataset’s overall (TD +NTD), NTD, and TD incidence classifications were good, as shown in Tables 4 and 5. The majority of classifiers have accuracy, precision, recall, and F-Score values greater than 60%. TD occurrences could not be classified by DT, KNN, or SVM-Poly. These classifiers have an F-Score of less than 0.55 out of a possible 1. SVM-RBF is the best performance in TD classification once again. As a result, Modified Day Average data sets outperform traditional Day Average data sets.

A three-year Hourly Dataset containing data recordings from 2 to 3 h before and after the commencement of the thunderstorm was used for thunderstorm classification. On Hourly Datasets, predictive classifier techniques are applied, and the results are shown in Table 5. Table 5 shows that SVM-RBF outperforms all other classifiers in terms of overall and TD performance.

Based on the description above, SVM-RBF with hourly dataset combination is the best dataset-classifier combination for all three datasets. Grid search and Bagging are used on obtained the best dataset-classifier combination (Hourly Dataset-SVM-RBF) to produce a hybrid classifier that enhances performance. Grid search improves performance by searching for the optimal parameter value in SVM-RBF.

Table 5 displays the TD, NTD, TD +NTD classification performance of hybrid Grid-SVM-RBF, Bag-SVM-RBF, and Grid-Bag-SVM-RBF classifiers. Grid-Bag-SVM-RBF has the greatest F-Score of 0.83, while Bag-SVM-RBF has the lowest F-Score of 0.80. Grid-Bag-SVM-RBF once again outperforms the competition in terms of testing accuracy, with a score of 82.04%. Training accuracy follows a similar performance trend.

for the proposed hybrid classifier may be improved using more years of data. The findings of this study might aid scholars and meteorologists in predicting or forecasting thunderstorms. Creating an expert system for the prediction of thunderstorm occurrence is recommended for future study efforts, with a good performance measure employing several classification and attribute selection strategies. The current study may also be beneficial in predicting other weather events such as rainfall.

## References

1. Litta, A.J., Idicula, S.M., Francis, C.N.: Artificial neural network for the prediction of thunderstorms over Kolkata. In: *J. Comput. Appl.* **50**(11), 50–55 (2012)
2. Wilks, D.S.: International variability and extreme value characteristics of severe stochastic daily precipitation. *Agric. For. Meteorol.* **93**, 153–169 (1999)
3. Litta, A.J., Idicula, S.M., Mohanty, U.C.: Artificial neural network model in prediction of meteorological parameters during pre-monsoon thunderstorms. *Int. J. Atmos. Sci.* **2013**, 1–14 (2013)
4. Saha, U., Maitra, A., Midya, S.K., Das, G.K.: Association of thunderstorm frequency with rainfall occurrences over an Indian urban metropolis. *Atmos. Res.* **138**, 240–252 (2014)
5. <http://www.tsgc.utexas.edu/stars/tstypes.html>
6. Chaudhuri, S.: Preferred type of cloud in the genesis of severe thunderstorms—a soft computing approach. *Atmos. Res.* **88**(2), 149–156 (2008)
7. Webb, R., King, P.: Forecasting thunderstorm and severe thunderstorm using computer models. In: 15th Annual Workshop of Bureau of Meteorology Research Center (BMRC) Modelling Workshop, (2003).
8. Colquhoun, J.R.: A decision tree method of forecasting thunderstorms, severe thunderstorms, and tornadoes. *Weather and Forecast.* **2**(4), 337–345 (1987)
9. Chaudhuri, S.: A Probe for Consistency in CAPE and CINE during the prevalence of severe thunderstorms: statistical-fuzzy coupled approach. *Atmos. Clim. Sci.* **4**(1), 197–205 (2011)
10. Basak, P., Sarkar, D., Mukhopadhyay, A.K.: Estimation of thunderstorm days from the radio-sonde observations at Kolkata (22.530 N, 88.330 E), India during pre-monsoon season: an ANN based approach. *Open Access E-J. Earth Sci. India* **5**(IV), 139–151 (2012)
11. Chakrabarty, H., Murthy, C.A., Gupta, D.A.: Application of pattern recognition techniques to predict severe thunderstorms. *Int. J. Comput. Theor. Eng.* **5**(6), 850–855 (2013)
12. Putra, A. W., Lursinsap, C.: Cumulonimbus prediction using artificial neural network backpropagation with radiosonde indices. 153–165 (2014)
13. Cintineo, J. L., Pavolonis, M. J., Sieglaff, J. M., Lindsey, D. T.: Probabilistic nowcasting of severe convection. In: National Weather Association Annual Meeting, Madison, WI, Seminar Nasional Penginderaan Jauh F18.1. (2012).
14. Ping, L., Tao-rong, Q., Yu-yuan, L.: The Study on the model of thunderstorm forecast based on RS-SVM. *J. Converg. Inf. Technol.* **8**(10), 66–74 (2013)
15. Bala, K., Choubey, D.K., Paul, S.: Soft computing and data mining techniques for thunderstorms and lightning prediction: a survey. In: International Conference on Electronics and Aerospace Technology (ICECA) Coimbatore, IEEE, pp. 42–46 (2017).
16. Choubey, D.K., Paul, S.: GA\_MLP NN: a hybrid intelligent system for diabetes disease diagnosis. *Int. J. Intell. Syst. Appl. (IJISA) MECS.* **8**, 49–59 (2016)
17. Choubey, D.K., Paul, S.: GA\_RBF NN: a classification system for diabetes. *Int. J. Biomed. Eng. Technol. (IJBET), Indersci.* **23**(1), 71–93 (2017)
18. Choubey, D.K., Paul, S.: GA\_SVM—a classification system for diagnosis of diabetes. In: Handbook of research on nature inspired soft computing and algorithms, pp. 359–397. IGI Global, Hershey (2017)
19. Chatterjee, D., Chakrabarty, H.: Application of machine learning technique to predict severe thunderstorms using upper air data. *Int. J. Sci. Eng. Res.* **6**(7), 1527–1530 (2015)
20. Chakrabarty, H., Bhattacharya, S.: Prediction of severe thunderstorms applying neural network using RSRW data. *Int. J. Comput. Appl.* **89**(16), 1–5 (2014)

21. Fix, E., Hodges, J.L., Jr.: Discriminatory analysis-nonparametric discrimination: consistency properties. In: Technical report. California University, Berkeley (1951)
22. Cover, T., Hart, P.: Nearest neighbor pattern classification. *IEEE Trans. Information. Theory* **13**(1), 21–27 (1967)
23. Kataria, A., Singh, M.D.: A review of data classification using K-nearest neighbor algorithm. *Int. J. Emerg. Technol. Adv. Eng.* **3**(6), 354–360 (2013)
24. Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H.: Top 10 algorithms in data mining. *Knowl. Inf. Syst.* **14**(1), 1–37 (2008)
25. Bhatia, N., Vandana: Survey of nearest neighbor techniques. *Int. J. Comput. Sci. Inf. Secur. (IJCSIS)* **8**(2), 302–305 (2010)
26. Yang, Y., Ault, T., Pierce, T., Lattimer, C. W.: Improving text categorization methods for event tracking. In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 65–72, (2000).
27. Xiubo, G., Tie-Yan, L., Qin, T., Andrew, A., Li, H., Shum, H. Y.: Query dependent ranking using k-nearest neighbor. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 115–122 (2008).
28. Xu, S., Wu, Y.: An algorithm for remote sensing image classification based on artificial immune B-cell network. *The Int. Arch. Photogram. Remote Sensing Spat. Inf. Sci.* **37**, 107–112 (2008)
29. Song, Y.Y., Lu, Y.: Decision tree method: application for classification and prediction. *Shanghai Arch. Psychiatry* **27**(2), 130–135 (2015)
30. Dasgupta, S., De, U.K.: A logistic regression model for prediction of pre-monsoon convective development over Kolkata. *Indian J. Radio Space Phys.* **33**, 251–255 (2004)
31. Vapnik, V.N.: The nature of statistical learning theory. Springer-Vargal New York, New York, NY (1995). <https://doi.org/10.1007/978-1-4757-2440-0>
32. Syarif, W., Prugel-Bennett, A., Wills, G.: SVM parameter optimization using grid search and genetic algorithm to improve classification performance. *TELKOMNIKA* **14**(4), 1502–1509 (2016)
33. Breiman, L.: Bagging predictors. *Mach. Learn.* **24**(2), 123–140 (1996)
34. Bajramovic, F., Mattern, F., Butko, N., Denzler, J.: A comparison of nearest neighbor search algorithms for generic object recognition. *Adv. Concepts Intell. Vision Syst. Springer, (LNCS)* **4179**, 1186–1197 (2006)
35. Ara, A., Maia, M., Louzada, F., Macêdo, S.: Random machines: a bagged-weighted support vector model with free kernel choice. *J. Data Sci.* (2021). <https://doi.org/10.6339/21-JDS1014>
36. Longadge, R., Dongre, S. S., Malik, L.: Class imbalance problem in data mining: review. *Int. J. Comput. Sci. Netw. (IJCSN)* **2**(1) (2013). <https://doi.org/10.48550/arXiv.1305.1707>
37. Abdellatif, S., Hassine, M. A. B., Yahia, S. B., Bouzeghoub, A.: ARCID: a new approach to deal with imbalanced datasets classification. In: SOFSEM 2018: theory and practice of computer science. SOFSEM 2018. Lecture Notes in Computer Science vol. 10706, (2018).
38. Li, Y., Li, H., Li, X., Xie, P.: On deep learning models for detection of thunderstorm Gale. *J. Internet Technol.* **21**(4) (2020). <https://doi.org/10.3966/160792642020072104001>
39. Azad, A. K., Reza, A., Islam, M. T., Rahman, M. S., Ayen, K.: Development of novel hybrid machine learning models for monthly thunderstorm frequency prediction over Bangladesh. *Nat. Hazards-springer* **108** (1), 1109–1135, 2021. <https://doi.org/10.1007/s11069-021-04722-9>
40. Kamangir, H., Collins, W., Tissot, P., King, S. A.: A deep-learning model to predict thunderstorms within 400 km2 South Texas domains *Meteorol. Appl.* **27**(2) 2020 <https://doi.org/10.1002/met.1905>
41. Zhou, K., Zheng, Y., Li, B., Dong, W., Zhang, X.: Forecasting different types of convective weather: a deep learning approach. *J. Meteorol. Res.* **33**, 797–809 (2019)
42. Bala, K., Paul, S., Ghosh, M.: Heuristic model to compute indices for classification of incidence of thunderstorms over ranchi with atmospheric parameter. *IEEE Access* **9**, 127086–127101 (2021)
43. Zhang, X., Mohanty, S.N., Parida, A.K., Pani, S.K.: Annual and on-monsoon rainfall prediction modeling using SVR-MLP: an empirical study from Odisha. *IEEE Access* **8**(1), 30223–30233 (2020). <https://doi.org/10.1109/ACCESS.2020.2972435>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



## Authors and Affiliations

**Kanchan Bala<sup>1</sup> · Sanchita Paul<sup>1</sup> · Sachi Nandan Mohanty<sup>2</sup> ·  
Satyasundara Mahapatra<sup>3</sup>**

Kanchan Bala  
kanchanbala237@gmail.com

Sanchita Paul  
sanchitapaul@bitmesra.ac.in

Satyasundara Mahapatra  
satyasundara123@gmail.com

- <sup>1</sup> Department of Computer Science and Engineering, BIT Mesra, Ranchi, India
- <sup>2</sup> Department of Computer Science and Engineering, Vardhaman College of Engineering (Autonomous), Hyderabad, India
- <sup>3</sup> Department of Computer Science and Engineering, Pranveer Singh Institute of Technology, Kanpur, Uttar Pradesh, India