Forecasting Thunderstorms in North East States of India during the Premonsoon Period: A Comparative Analysis of Models Using Historical Index and Surface Data (1980-2020)

Some irrelevant pages and text will be removed due to privacy but you have the most of the content , use your friend google and gpt too apart from this for bertter understaniding

Abstract

Accurate forecasting of thunderstorms is crucial for minimizing their impact on agriculture, infrastructure, and public safety. In this research, we develop thunderstorm forecasting models for the North East states of India during the pre-monsoon period (1980-2020) using machine learning techniques. We compare the performance of Random Forest, XGBoost, and Support Vector Machines (SVM) models based on historical index and surface data collected from the India Meteorological Department. Evaluation metrics such as Probability of Detection (POD), False Alarm Rate (FAR), Heidke Skill Score (HSS), and Critical Success Index (CSI) are employed to assess accuracy. Preliminary findings show that the Random Forest model outperforms the other models, offering higher accuracy scores across all states and evaluation metrics. The research contributes to enhancing thunderstorm forecasting capabilities and supporting decision-making processes in disaster management.

Keywords: Thunderstorms, forecasting, machine learning, Random Forest, XGBoost, Support Vector Machines (SVM), North East states of India, pre-monsoon period, historical data, index data, surface data, evaluation metrics

1 Introduction

Thunderstorms are fascinating natural phenomena characterized by towering cumulus or cumulonimbus clouds that produce lightning and thunder. In India, these storms are particularly prevalent during the pre-monsoon months of March to May. The North East states of India, including Patna, Guwahati, Gorakhpur, Bhubaneswar, Kolkata, Agartala, Ranchi, and Lucknow, experience a significant occurrence of thunderstorms during this period. Accurately

predicting these storms is essential to minimize their adverse impacts on agriculture, infrastructure, and public safety. However, traditional meteorological models often struggle to effectively forecast these complex weather patterns.

In this research, we utilize machine learning models, namely Random Forest, XGBoost, and Support Vector Machines (SVM), to develop thunderstorm forecasting models for the North East states of India during the pre-monsoon period from 1980 to 2020. Data is collected from the India Meteorological Department, a reliable source for meteorological information. By analyzing historical index and surface data, we evaluate and compare the performance of these models. Evaluation metrics such as Probability of Detection (POD), False Alarm Rate (FAR), Heidke Skill Score (HSS), and Critical Success Index (CSI) are employed to assess their accuracy and effectiveness. We also consider machine learning metrics such as accuracy, precision, recall, and F1 score.

Preliminary findings indicate that the Random Forest model outperforms XGBoost and SVM, exhibiting higher accuracy scores across all states and evaluation metrics. These results contribute to improving the forecasting capabilities of thunderstorms in the North East states of India. Enhanced forecasting accuracy supports decision-making processes in disaster management and facilitates effective mitigation efforts.

Thunderstorms bring both advantages and disadvantages. On one hand, they contribute to the water cycle by providing rainfall, which is vital for agriculture and replenishing water sources. Thunderstorms also help in removing air pollutants and refreshing the atmosphere. However, they can also pose risks such as lightning strikes, strong winds, heavy rainfall, and hailstorms, which can lead to property damage, power outages, and threats to human safety.

By employing machine learning techniques, we aim to enhance our understanding and prediction of thunderstorms in the North East states of India, ultimately assisting in better preparedness and minimizing the impacts of these weather events on the region.

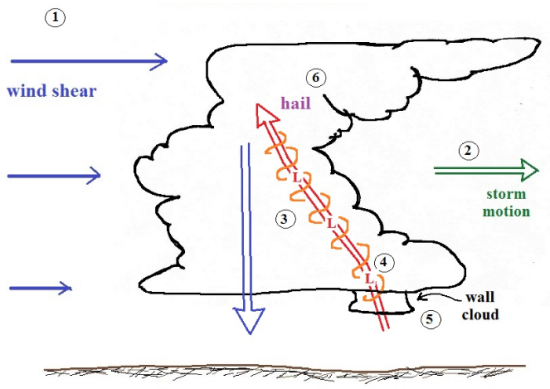

Figure 1: Thunderstrom Formation

2 METHODOLOGIES

2.1 Data

Data for the thunderstorm prediction model was collected daily from 1980 to 2020, utilizing two different types of data: index data and surface data.

2.1.1 Index Data

Two types of weather balloons were used to collect index data for thunderstorm prediction[1]. The first balloon, called the Radiosonde, was launched at 5:30 AM IST (0 UTC) and equipped with sensors to gather air pressure, temperature, and relative humidity data. It ascended at a rate of approximately 20 km/hr and carried around 375 grams of hydrogen gas. This balloon provided observations above ground level. The second balloon, known as the Pilot Weather Balloon, was launched at 4:30 PM IST (11 AM UTC) without sensors. Its main purpose was to detect wind speed and direction using a theodolite. The amount of hydrogen gas filled in the balloon varied based on cloud cover, with 125 grams used in the presence of more clouds and up to 325 grams in clear skies. The weight of the balloon varied accordingly, allowing for the determination of wind characteristics.

2.1.2 Surface Data

Surface data was recorded twice a day—at 8:30 AM IST (3 UTC) and 5:30 PM IST (12:30 PM UTC). Ground instruments were used to measure various factors, including wind speed, temperature (minimum and maximum), rainfall, cloud visibility, wind direction, evaporation, and dew point[1]. Surface data collection instruments included a Cup Anemometer for wind speed and direction, Rain Gauge for rainfall measurement, Single and Double Stevenson Screens for temperature and humidity recordings, Self-Recording Rain Gauge for automated rainfall measurement, AWS with Solar Panels for various weather parameters, Soil Thermometers for irrigation research, Open Pan Evaporation for daily evaporation, and Dew Gauge Stand for measuring dew points. The combination of index data collected from weather balloons and surface data provided a comprehensive dataset for training the neural network model and predicting thunderstorm occurrences.

2.1.3 Relevant Indexes

The dataset includes several relevant indexes that play a crucial role in assessing atmospheric instability and thunderstorm development. These indexes are:

- GMT (Greenwich Mean Time) - Time reference for the data collection.
- SWEAT (Severe Weather Threat) - Measures atmospheric instability and indicates the potential for severe weather. Higher values suggest a greater likelihood of thunderstorm development and severe conditions.

- Showalter Index - Represents the vertical temperature difference between an air parcel lifted dry adiabatically and the environment. Negative values indicate unstable conditions and the potential for thunderstorm development.
- LIFTED Index - Represents the temperature difference between the environmental air and an air parcel lifted adiabatically. Negative values indicate unstable conditions and the potential for thunderstorm development.
- K Index - Measures the potential for thunderstorm development based on the vertical temperature lapse rate and moisture content. Higher values indicate greater instability and the potential for thunderstorm development.
- Cross Totals Index - Combines temperature, moisture, and wind data to assess the potential for severe weather. It indicates the overall instability in the atmosphere and the likelihood of thunderstorm development.
- Vertical Totals Index - Represents the total vertical temperature difference in the atmosphere. It indicates the potential for thunderstorm development and severe weather based on the vertical temperature gradient.
- Totals Totals Index - Measures atmospheric instability and the potential for thunderstorm development. It combines temperature, moisture, and wind data to assess overall instability in the atmosphere.
- TLCL (Temperature at Lifted Condensation Level) - Represents the temperature at which an air parcel would reach its condensation level when lifted.
- PLCL (Pressure at Lifted Condensation Level) - Represents the pressure at which an air parcel would reach its condensation level when lifted.
- CINE (Convective Inhibition Energy) - Measures the energy required to initiate convection and the potential for thunderstorm development.
- CAPE (Convective Available Potential Energy) - Indicates the available energy for convection and the potential for thunderstorm development.
- PRECIPITABLE WATER - Represents the vertical depth of water vapor in the atmosphere and indicates the potential for precipitation.
- 1000-500 THICKNESS - Measures the thickness of the atmospheric layer between the 1000 hPa and 500 hPa pressure levels. It relates to temperature gradients and atmospheric stability.

These thresholding parameters are commonly accepted in meteorological studies and provide a general indication of the potential for thunderstorm development based on the respective indexes.

accurate predictions or decisions, and continuously improve their performance.

3.1.1 Data Collection

The data collected includes surface and index data, which are recorded at 0 and 12 GMT.

The data collection period spans from 1980 to 2020, providing a substantial historical dataset for analysis and model training. By gathering data from multiple observatories across different regions, a diverse and representative dataset is obtained, enabling the development of robust machine learning models.

It is important to note that the data collection process follows the standard protocols and quality control measures established . These measures ensure the reliability and accuracy of the collected data, which is essential for obtaining meaningful insights and building reliable machine learning models.

3.1.2 Data Preprocessing

Data preprocessing is a crucial step in the machine learning pipeline. In this phase, the collected data is transformed and cleaned to ensure its quality and suitability for analysis and model training.

Since the collected data is already accurate and does not contain redundant or irrelevant information, the focus of preprocessing was on preparing the data for further analysis. The following preprocessing steps were performed:

- **Data Filtering:** In the index table, data recorded at 0 GMT was selected for further analysis. This filtering was done to ensure consistency and relevance in the dataset.
- **Column Selection:** From the surface table, only the columns related to thunderstorm (TH) and hailstorm (HA) were considered. These two variables were combined into a single column using the logical OR operator, as both TH and HA are reported under the cumulonimbus cloud category. This consolidation of columns helps capture the occurrence of thunderstorms or hailstorms accurately.
- **Joining Tables:** The index and surface data tables were joined based on the date column using an inner join operation. This resulted in a merged dataset containing relevant information from both tables.
- **Feature Engineering:** Several new columns were created based on domain knowledge and the available variables in the merged dataset. The following feature engineering steps were performed:
  - The columns 'Showalter index' and 'LIFTED index' were combined into the 'Environmental_Stability' column.

  - The column 'PRECIPITABLE WATER' was retained as 'Moisture_Indices'.
  - The columns 'CAPE' and 'CINE' were combined into the 'Convective_Potential' column.
  - The column '1000-500 THICKNESS' was retained as 'Temperature_Pressure'.
  - The column 'PLCL' was retained as 'Moisture_Temperature_Profiles'.
- **Selected Indices:** The columns 'SWEAT index', 'K index', and 'Totals totals index' were retained as separate variables without any modifications.

After performing these preprocessing steps, the merged dataset was ready for further analysis and model training. The resulting dataset consisted of the following columns:

- Date
- SWEAT index
- K index
- Totals totals index
- Environmental_Stability
- Moisture_Indices
- Convective_Potential
- Temperature_Pressure
- Moisture_Temperature_Profiles
- TH

These columns capture various environmental factors and indices that are relevant for predicting thunderstorm occurrences accurately.

3.1.3 Data Analysis

In the data analysis phase, the preprocessed data is explored and analyzed to gain insights and understand its characteristics. This step involves descriptive statistics, data visualization, and exploratory data analysis techniques to identify patterns, correlations, and outliers in the data. The analysis results help in selecting appropriate features and understanding the relationships between variables.

Following tables show some basic statistical analysis of our data:

3.1.4 Model Training

Model training is an essential step in building a machine learning model for thunderstorm prediction. The goal is to develop a model that can accurately predict the occurrence of thunderstorms based on various independent variables (features) while using the TH variable as the dependent variable. In this section, we outline the process of model training and mention the list of features used in the training.

The thunderstorm prediction model was trained using a dataset that includes the following features:

- SWEAT index
- K index
- Totals totals index
- Environmental stability
- Moisture indices
- Convective potential
- Temperature pressure
- Moisture-temperature profiles

The TH(Thunderstrom Occurrence) variable is the target variable that we aim to predict using these features.

The model training process involves the following steps:

1. Data Preparation: The dataset is prepared by gathering historical data of the independent variables (features) and the corresponding TH values. The data is cleaned, preprocessed, and formatted appropriately for training the model.
2. Model Selection: An appropriate machine learning algorithm is selected for the thunderstorm prediction task. Commonly used algorithms for regression tasks include linear regression, decision trees, random forests, support vector machines (SVM), or neural networks.
3. Data Split: The dataset is divided into training and validation sets. The training set is used to train the model, while the validation set is used to evaluate its performance and make necessary adjustments.
4. Model Training: The selected algorithm is trained using the training set. The model learns the patterns and relationships between the independent variables and the TH variable during this process. The model's parameters are optimized to minimize the prediction errors.

The trained model can then be used to make predictions on new, unseen data to determine the likelihood of thunderstorms based on the given independent variables.

3.1.5 Model Testing

Once the model is trained, it is crucial to evaluate its performance to determine how well it can predict thunderstorms. The model is tested using a separate test dataset that was not used during the training process. This allows us to assess how well the model generalizes to unseen data and provides valuable insights into its accuracy and effectiveness.

During the model testing phase, various evaluation metrics are calculated to measure the model's performance. These metrics include accuracy, precision, recall, and others, depending on the specific requirements of the thunderstorm prediction task. These metrics help us understand the model's ability to correctly classify thunderstorms and non-thunderstorm events, as well as its overall predictive power.

In addition to the standard evaluation metrics, there are also specific scientific metrics that are essential for thunderstorm predictions. These metrics may include:

False Alarm Rate: This metric measures the percentage of false alarms generated by the model, indicating the instances when the model predicted a thunderstorm, but no thunderstorm occurred. Minimizing false alarms is crucial to avoid unnecessary warnings and reduce unnecessary disruptions.

Hit Rate: The hit rate measures the percentage of correctly predicted thunderstorm events. It indicates the model's ability to identify actual thunderstorms accurately.

Probability of Detection: This metric represents the percentage of observed thunderstorm events correctly predicted by the model. It measures the model's sensitivity in detecting thunderstorms.

Critical Success Index: The critical success index evaluates the model's overall skill in predicting both thunderstorm and non-thunderstorm events. It considers both hits and correct rejections, providing a comprehensive assessment of the model's performance.

These metrics, along with others specific to thunderstorm prediction, are typically included in the results section of the model evaluation. The results section presents a detailed analysis of the model's performance. It also serves as a basis for further improvements in the model, such as revisiting the data analysis stage and selecting appropriate features if the model performs poorly.

By continuously refining the model based on its performance evaluation, we can develop a robust and accurate thunderstorm prediction system.

3.1.6 Deployment

The deployment phase of the machine learning lifecycle involves making the trained model available in a production environment where it can be used to make real-time predictions or decisions. The goal is to integrate the model into a user-friendly system that allows users to input relevant indexes and obtain the predicted outcome of a thunderstorm occurrence.

To ensure a smooth user experience, the deployment system should be designed in a way that makes it simple for a normal human to interact with. Users should only need to input the relevant indexes, such as the SWEAT index, K index, Totals totals index, Environmental_Stability, Moisture_Indices, Convective_Potential, Temperature_Pressure, and Moisture_Temperature_Profiles. The deployed model will then process this input and provide the output in a user-friendly format.

The output of the deployed model can be in a boolean format, indicating whether a thunderstorm is predicted to occur or not. Additionally, the system can also provide the probability or confidence score associated with the prediction. This probability score represents the model's estimation of the likelihood of a thunderstorm occurrence based on the input indexes.

By providing the output in a boolean format along with the probability of a thunderstorm occurrence, users can make informed decisions based on the model's prediction. They can assess the level of confidence in the prediction and take appropriate actions or precautions accordingly.

It is crucial to ensure ongoing monitoring and maintenance of the deployed model. Regular monitoring helps to detect any performance degradation or drift that may occur over time. If necessary, the model can be retrained periodically using updated data to maintain its accuracy and reliability.

Overall, the deployment phase aims to make the trained model easily accessible and usable by normal users. By providing a straightforward interface for inputting relevant indexes and presenting the prediction output in a clear and understandable manner, the system can effectively support decision-making regarding thunderstorm occurrences.

3.2 Machine Learning in Thunderstorm Prediction

Machine Learning plays a crucial role in thunderstorm prediction due to its ability to handle complex patterns and relationships in large datasets. By training models on historical weather data and corresponding thunderstorm occurrences, Machine Learning algorithms can learn to recognize the patterns indicative of thunderstorm conditions and make accurate predictions.

3.3 Machine Learning Approaches for Binary Predictions

In binary prediction tasks, where the goal is to classify data into two categories (e.g., thunderstorm or non-thunderstorm), several Machine Learning approaches can be utilized. The following table provides an overview of different Machine Learning approaches and their suitability for binary predictions in a tabular format:

Table 10: Machine Learning Approaches for Binary Predictions

Approach	**Description**	**Relevant Information**
Logistic Regression	Linear model with a logistic function to model binary outcomes.	Requires feature scaling and assumes linear relationship between features and outcome.
Random Forest	Ensemble of decision trees that make predictions based on multiple tree outputs.	Handles non-linear relationships, feature importance analysis, and mitigates overfitting.
Support Vector Machines (SVM)	Separates data into different classes using hyperplanes in a high-dimensional space.	Effective in high-dimensional spaces, but may be sensitive to parameter tuning.
XGBoost	Gradient boosting framework that combines multiple weak models to create a strong predictor.	Handles complex relationships, handles missing values, and provides feature importance analysis.
Neural Networks	Deep learning models that consist of multiple interconnected layers for complex pattern recognition.	Capable of learning complex patterns, but requires large amounts of data and computational resources.

These approaches offer different strengths and can be applied based on the characteristics of the data and the desired performance. By leveraging these Machine Learning techniques, accurate binary predictions for thunderstorm occurrence can be achieved.

3.4 Machine Learning Techniques Involved

In this research, we employ three popular machine learning techniques for binary classification of thunderstorms: Random Forest, XGBoost, and Support Vector Machines (SVM).

3.4.1 Random Forest

Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. It creates a collection of decision trees, where each tree is trained on a random subset of the data and features. The final prediction is determined by aggregating the predictions of all individual trees.

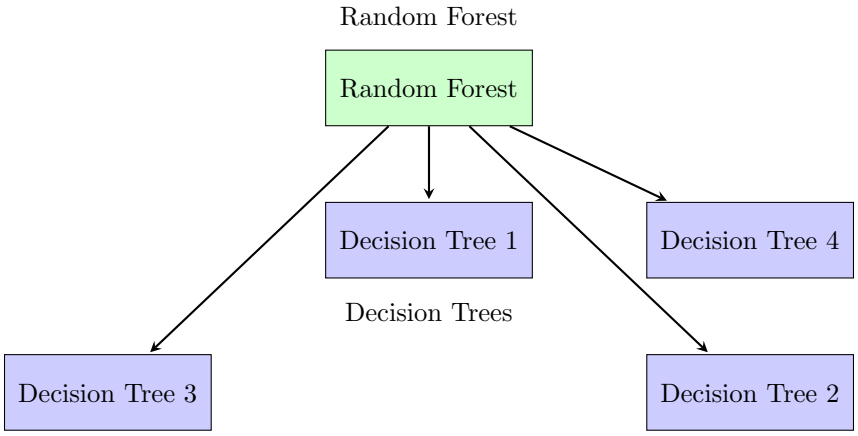

Figure 4: Random Forest Classifier for Binary Thunderstorm Prediction

Figure 4 illustrates the concept of a Random Forest with multiple decision trees. Each decision tree independently classifies an instance, and the final prediction is determined by majority voting or averaging the predictions.

3.4.2 XGBoost

XGBoost (eXtreme Gradient Boosting) is another ensemble learning method that combines multiple weak prediction models, typically decision trees, to create a strong predictive model. It trains the weak models sequentially, where each subsequent model corrects the errors made by the previous models.

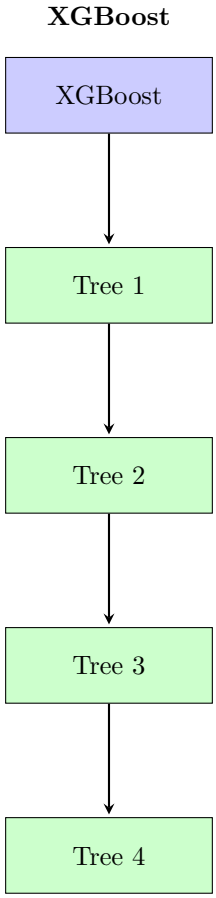

Figure 5: XGBoost Classifier for Binary Thunderstorm Prediction

Figure 5 demonstrates the process of XGBoost, where each weak learner (decision tree) focuses on the residual errors from the previous learner. The final prediction is the sum of predictions from all weak learners.

3.4.3 Support Vector Machines (SVM)

Support Vector Machines (SVM) is a powerful supervised learning algorithm used for both classification and regression tasks. In binary classification, SVM aims to find an optimal hyperplane that separates the data points of different classes while maximizing the margin between them.

Figure 6 showcases the concept of SVM, where the optimal hyperplane separates the two classes (thunderstorm and non-thunderstorm) with the maximum margin. The data points closest to the hyperplane are called support vectors.

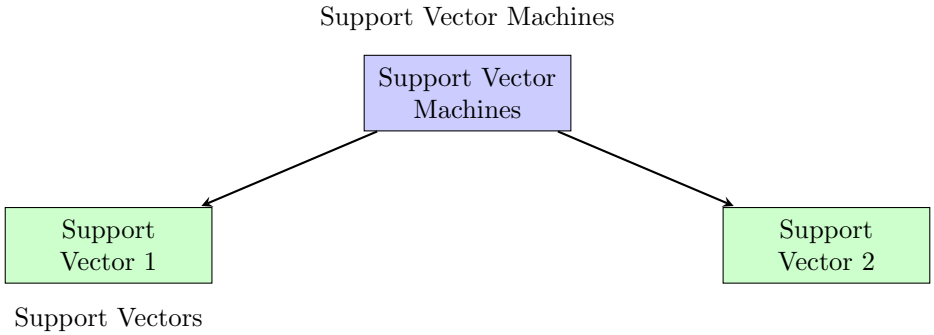

Figure 6: Support Vector Machines (SVM) Classifier for Binary Thunderstorm Prediction

These machine learning techniques are applied to the binary classification of thunderstorms by training them on the collected data, including the various indexes and surface data. The trained models can then predict whether a given set of meteorological features indicates the presence or absence of a thunderstorm.

4 Results

In this section, we present the performance metrics used to evaluate the thunderstorm forecasting models. These metrics provide different perspectives on the model's classification performance.

4.1 Binary Classification Metrics

For binary classification, we utilize the following metrics:

- **Accuracy**: Accuracy measures the overall correctness of the classification model.
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$
- **Precision**: Precision measures the proportion of correctly predicted positive instances out of the total instances predicted as positive.
$$Precision = \frac{TP}{TP + FP}$$
- **Recall (Sensitivity or True Positive Rate)**: Recall measures the proportion of correctly predicted positive instances out of the total actual positive instances.
$$Recall = \frac{TP}{TP + FN}$$

- **F1 Score**: The F1 score is the harmonic mean of precision and recall, providing a balanced measure between the two.

$$F1Score = \frac{2 \cdot (\text{Precision} \cdot \text{Recall})}{\text{Precision} + \text{Recall}}$$

4.2 Additional Classification Metrics

In addition to the binary classification metrics, we also consider the following metrics:

- **Probability of Detection (POD)**: POD, also known as Sensitivity or True Positive Rate, measures the proportion of actual positive instances correctly predicted as positive.

$$\text{POD} = \frac{TP}{TP + FN}$$

- **False Alarm Rate (FAR)**: FAR measures the proportion of actual negative instances incorrectly predicted as positive.

$$\text{FAR} = \frac{FP}{FP + TN}$$

- **Heidke Skill Score (HSS)**: HSS evaluates the skill of a classification model compared to random chance. It considers the improvement of the model over the random model.

$$\text{HSS} = \frac{2 \cdot (TP \cdot TN - FP \cdot FN)}{(TP + FN) \cdot (FN + TN) + (TP + FP) \cdot (FP + TN)}$$

- **Critical Success Index (CSI)**: CSI, also known as Threat Score or Gilbert Skill Score, measures the proportion of correctly predicted events (both positive and negative) out of the total events.

$$\text{CSI} = \frac{TP}{TP + FP + FN}$$

These metrics provide valuable insights into the performance of our thunderstorm forecasting models. We will now discuss and interpret the results in the following section.