International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 5, Sep-Oct-2026, ISSN (Online): 2395-566X

# INTELLIGENT WEATHER PREDICTION SYSTEM USING MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

**Indhuprakash M, Gokul R, Harishragavendra M, Parthasharathy R (Project Guide: M. Premkumar, Assistant Professor)**
Department of Artificial Intelligence and Data Science, Sengunthar Engineering College, Tiruchengode, Tamil Nadu, India

**Abstract Weather prediction supports agriculture, trans-portation, travel planning, and public safety. This paper presents an Intelligent Weather Prediction System that combines machine learning techniques with an accessible application interface. The proposed workflow uses weather observations such as tempera-ture, humidity, atmospheric pressure, wind speed, cloud cover, and rainfall, depending on the data source. It includes data collection, preprocessing, exploratory analysis, feature prepara-tion, model training, evaluation, and presentation of predictions. Candidate models include regression methods for continuous weather variables, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Long Short-Term Memory (LSTM) for suitable classification, regression, or sequential forecasting tasks. The application may also provide AI-assisted weather question answering and concise weather-news summaries when these components are implemented and connected to reliable sources. The system is intended to make weather information easier to understand and use. Actual performance must be measured on the selected dataset; no accuracy result is claimed in this draft. The final paper should report the implemented models, dataset, evaluation results, and system screenshots.**

**Index Terms—weather prediction, machine learning, artifi-cial intelligence, rainfall forecasting, SVM, KNN, LSTM, weather application.**

© 2026 IJSRET

## I. INTRODUCTION

Weather conditions influence daily routines, farming, transport, tourism, water management, and emergency preparedness. Temperature, humidity, atmospheric pressure, wind, cloud cover, and precipitation interact in complex ways. A change in one variable may be associated with changes in other variables, while local geography and seasonal patterns can further influence the observed conditions. Because of these interactions, weather prediction is a challenging problem.Traditional forecasting approaches use physical models and meteorological expertise. These methods are valuable, but processing large and diverse datasets can require substan-tial computing resources and specialist knowledge. Machine learning provides an additional approach: it learns patterns from historical observations and uses those patterns to estimate future values or classify expected conditions. Its usefulness depends on the quality, relevance, and coverage of the training data.

This project proposes an Intelligent Weather Prediction System using Machine Learning and Artificial Intelligence. The aim is to organize weather data, apply suitable predictive models, and present results through a user-friendly application. Depending on the final implementation, the system can accept a location or weather-related input, retrieve weather information, process the required features, and display a prediction. AI-assisted functions can make the output easier to explore through natural-language questions or short summaries.

The system is designed as a decision-support tool rather than a replacement for official meteorological services. Users should consider the forecast horizon, data freshness, and limi-tations of the model when interpreting a result. Official warn-ings and emergency instructions should always take priority.

## II. RELATED WORK

Research on rainfall prediction has explored statistical mod-els, decision trees, support vector machines, ensemble meth-ods, and neural networks. Hassan et al. presented a study ti-tled "Machine Learning-Based Rainfall Prediction: Unveiling Insights and Forecasting for Improved Preparedness" in IEEE Access in 2023 [1]. Their approach describes weather-station data, preprocessing, outlier analysis, feature selection, model comparison, and a web-based application built with Flask. The study also examines classification metrics and regional patterns.

The reference paper is useful for understanding how a rainfall-prediction study can be organized from data prepara-tion through evaluation and application development. It dis-cusses models including Naive Bayes, Decision Tree, Support Vector Machine, Random Forest, Logistic Regression, Artifi-cial Neural Network, and Long Short-Term Memory. Its re-ported experimental results apply to its own dataset, features, experimental design, and implementation. They must not be represented as results of the present project.

The proposed project differs according to its intended combination of weather prediction and AI-assisted user interaction. To make the comparison meaningful, the final version should identify the dataset actually used, the selected target variables, the models actually implemented, and the measured performance on a held-out test set. A fair comparison should use the same data split and compatible evaluation metrics for each candidate model.

## III. PROBLEM STATEMENT

Weather information may be spread across multiple sources and can be difficult for users to interpret. A prediction interface that does not clearly identify the location, time, source, or limitations of its output may cause confusion. A machine learning model can also produce unreliable results when trained with incomplete, biased, outdated, or poorly prepared data. Therefore, the project addresses the need for a simple system that organizes relevant weather features, applies an evaluated prediction method, and presents the result in a clear format.

The system must also distinguish between measured weather information, model-generated predictions, and AI-generated explanations. If a live weather API or an AI service is unavailable, the application should communicate that limitation rather than fabricate a result.

## IV. OBJECTIVES

The objectives of the proposed system are:

- To collect or load a suitable weather dataset from a docu-mented source.
- To inspect the dataset and handle missing values, dupli-cates, inconsistent formats, and unsuitable features.

© 2026 IJSRET

- To identify relevant meteorological features for the selected prediction target.
- To train and compare appropriate machine learning mod-els.
- To evaluate predictions with metrics that match the target variable.
- To present weather information through a clear applica-tion interface.
- To explore AI-assisted features such as a weather chatbot or weather-news summary when supported by the imple-mentation.
- To document limitations and avoid unsupported claims about model accuracy.

## V. PROPOSED SYSTEM

The proposed system follows a sequence of stages: data collection, preprocessing, feature preparation, model training, model evaluation, prediction, and result presentation. A user interacts with the application, and the application obtains the relevant input data from the configured dataset or weather service. The preprocessing component transforms the input into the format expected by the trained model. The model then produces the selected output, which is displayed together with relevant context.

The precise workflow depends on the deployed version. If the project predicts rainfall occurrence, it can be treated as a classification problem. If it predicts temperature or rainfall amount, it is a regression problem. If it uses ordered observations across time, a time-series method such as LSTM may be considered. These tasks should not be combined into one evaluation without defining their individual targets.

### A. Data Collection

Weather data may contain date and time, location, minimum and maximum temperature, humidity, pressure, wind speed, wind direction, cloud cover, sunshine, and rainfall. The selected data source must be identified in the final paper. If a live API is used, the paper should specify the provider, rele-vant endpoint or data fields, and how missing or unavailable responses are handled. API keys and other credentials must not be included in the published paper or source code.

### B. Data Preprocessing

Preprocessing is important because machine learning algorithms depend on the quality and representation of input features. The process may include removing duplicate records, checking data types, converting date fields, handling missing values, encoding categorical features, and scaling numerical values where required. Outliers should be examined rather than automatically removed, because extreme weather values can be meaningful observations.

The dataset should be divided into training and testing por-tions before fitting the model. For time-series prediction, a chronological split is generally more appropriate than randomly mixing earlier and later observations. This helps re-duce data leakage, where information from the future acci-dentally influences training. Any transformation learned from data, such as scaling or imputation, should be fitted on training data and then applied to the test data.

### C. Feature Analysis and Selection

Feature analysis helps identify which variables may con-tribute useful information to the target prediction. Descrip-tive statistics and visualizations can reveal distributions, missingness, outliers, and possible relationships between variables. Correlation analysis can identify associations, but correlation alone does not establish causation. Feature selection should be carried out using the training data and a method appropriate to the model and task.

For example, humidity, pressure, temperature, wind conditions, cloud cover, and recent rainfall may be relevant to rainfall prediction. The final list must reflect the fields available in the selected dataset and the actual experiment. Features that would not be available at prediction time should not be used to create an unrealistically strong evaluation.

## VI. MACHINE LEARNING MODELS

The project may evaluate the following candidate ap-proaches. The final paper must retain only the models that are actually implemented and tested.

1. **Regression Models**
   Regression methods estimate continuous target values, such as temperature or rainfall amount. A simple linear regression model estimates a relationship between input features and a numerical output. More complex regression approaches can

© 2026 IJSRET

represent nonlinear relationships. Regression models are eval-uated with error-based metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). They are not

directly interchangeable with classifiers that predict categories such as “rain” or “no rain.”

2. **Support Vector Machine**
   Support Vector Machine (SVM) is a supervised learning method that can be used for classification or regression. In classification, it attempts to identify a decision boundary that separates classes, with support vectors helping define that boundary. Kernel functions can allow nonlinear decision boundaries. SVM performance can depend on feature scal-ing and hyperparameters, so these choices should be selected using training data and validation rather than the test set.

3. **K-Nearest Neighbors**
   K-Nearest Neighbors (KNN) makes predictions by identi-fying training observations that are close to a new observation under a chosen distance measure. For classification, the neigh-boring labels can be combined by voting; for regression, their numerical target values can be averaged or weighted. KNN is straightforward to understand, but its performance can be af-fected by feature scaling, the choice of distance measure, the number of neighbors, and the size of the training dataset.

4. **Long Short-Term Memory**
   Long Short-Term Memory (LSTM) is a recurrent neural-network architecture designed to learn patterns in sequential data. Its memory cells and gates help retain or discard informa-tion across time steps. It may be appropriate when the project has enough chronologically ordered weather observations and aims to forecast future values from sequences of earlier obser-vations. The sequence length, input features, forecast horizon, and training setup should be documented. LSTM should not be described as superior unless an experiment demonstrates that it performs better on the selected task.

## VII. SYSTEM ARCHITECTURE AND WORKFLOW

The proposed high-level workflow is:

- The user enters a location or chooses a weather-prediction function.
- The application obtains the required observations from its configured data source.
- The data-preparation component validates fields and transforms inputs.
- The selected trained model receives the prepared features.
- The prediction component returns the output for the de-fined target and time horizon.
- The interface displays the result and, where available, rel-evant source or timestamp information.
- Optional AI functions respond to weather-related ques-tions or summarize available weather information.

The application should validate user input, handle network and service errors, and avoid showing stale information as if it were current. If different locations are supported, the se-lected location should be passed through the complete request pipeline rather than replaced with a fixed default. The system should also make clear whether a value comes from an external weather provider or from the machine learning model.

## VIII. AI-ASSISTED FEATURES

An AI-assisted chatbot can provide a natural-language in-terface for questions about the weather information available to the application. For example, it may explain the meaning of humidity or summarize the forecast for a selected location. The chatbot should use relevant data supplied by the applica-tion and should not invent current weather conditions when the source is unavailable

A weather-news summary feature may turn selected weather reports or official notices into concise text. The source and time of the information should remain identifiable. Generated summaries should not change the meaning of an official warn-ing or replace the warning itself. If an external language-model API is used, the final paper should state the service and de-scribe the general role it plays, without publishing secret keys.AI-generated explanations and machine learning forecasts serve different roles. A forecast model estimates a target from

© 2026 IJSRET

input features, while a language model generates text. The application should not treat fluent generated text as evidence that a forecast is accurate.

## IX. IMPLEMENTATION ENVIRONMENT

A Python-based implementation can support data process-ing, model training, and evaluation. Libraries should be listed only when they are actually used. For example, pandas and NumPy can support tabular data preparation, scikit-learn can support common machine learning models and evaluation, and a deep-learning framework may be used for LSTM.

Flask may be used to connect Python prediction logic to a web interface if that matches the actual application.The final implementation section should record the actual operating environment, programming language and version, libraries, dataset source, API provider, model parameters, and deployment method. This information makes the experiment easier to reproduce. The paper should not claim a tool or feature was used merely because it is a possible choice.

## X. EXPERIMENTAL EVALUATION

Model evaluation should be designed around the prediction target. For a binary target such as rainfall occurrence, accuracy, precision, recall, and F1-score can describe different aspects of performance. Accuracy alone may be misleading when one class is much more common than another. A confusion matrix can show true positives, true negatives, false positives, and false negatives.

For continuous targets such as temperature or rainfall amount, MAE and RMSE measure prediction error in the target's units, while the coefficient of determination may provide additional context when appropriate. Results should be calculated on held-out data that was not used to fit themodel or select its hyperparameters. If multiple models are compared, they should use consistent data partitions and evaluation procedures.
Experimental Details To Be Completed From Actual Tests

| Evaluation item | Information to report |
|---|---|
| Dataset | Source, location coverage, date range, and record count |
| Prediction target | Rain occurrence, temperature, rainfall amount, or another defined target |
| Data split | Training, validation, and test approach |
| Models | Only implemented and tested models |
| Metrics | Appropriate classification or regression metrics |
| Best model | Selected from measured results, not assumed in advance |

## XI. RESULTS AND DISCUSSION

Project-specific results are not supplied in the materials used to create this draft. Therefore, numerical accuracy, preci-sion, recall, F1-score, MAE, RMSE, or claims about the best-performing model are intentionally not invented. After testing, the results section should include a table of measured metrics, a comparison of the models, and a discussion of errors and limitations.
The discussion should examine whether performance changes across locations, seasons, or weather conditions when the dataset supports such analysis. Rainfall events may be less frequent than non-rainy periods, so the evaluation should consider whether the model misses important rainfall cases. For continuous predictions, the paper should inspect the magnitude and distribution of prediction errors. Any conclusions should be limited to the tested data and forecast horizon.

Screenshots of the implemented application can be included near the relevant discussion. Useful screenshots may show the input form, location selection, prediction output, chatbot interface, or weather-news summary. Each screenshot should be genuine evidence from the running project, with a numbered caption and readable labels. Diagrams and screenshots must be added as image files and referenced correctly in the LaTeX source.

## XII. ADVANTAGES

The proposed approach can bring several practical benefits when implemented and validated appropriately. First, it can provide a single interface for viewing weather-related predictions. Second, machine learning can identify patterns in historical data that may help estimate selected weather variables. Third, a clear interface can make predictions easier to interpret for users who do not have a technical background. Fourth, op-

© 2026 IJSRET

tional natural-language features may help users ask questions about the displayed information. Finally, a modular design allows the data source, predictive model, and user interface to be improved independently.

## XIII. LIMITATIONS AND RESPONSIBLE USE

Prediction quality depends on the dataset, geographical coverage, missing values, model assumptions, and forecast horizon. A model trained on one region may not generalize to another region without suitable data and testing. API outages, network delays, outdated observations, and changes in data formats can also affect the application.

Weather prediction is inherently uncertain, and a machine learning output should not be presented as a guarantee. The interface should identify the forecast time and source where possible. The system should avoid generating false emergency warnings and should direct users to official meteorological information for safety-critical decisions. The final paper should disclose limitations and should not claim real-time functionality unless the application actually obtains and displays current data.

## XIV. CONCLUSION AND FUTURE WORK

This paper describes a proposed Intelligent Weather Predic-tion System using machine learning and AI-assisted interac-tion. The workflow covers weather-data collection, prepro-cessing, feature analysis, model training, evaluation, and pre-sentation through an application. Regression models, SVM, KNN, and LSTM are candidate methods, but their inclu-sion in the final implementation and their comparative perfor-mance must be verified experimentally. Optional chatbot and weather-news features can improve accessibility when con-nected to reliable information sources.

Future work can include testing on a larger and more geographically diverse dataset, evaluating predictions across different forecast horizons, improving handling of missing or delayed observations, and presenting uncertainty more clearly. Further development should also compare candidate models using reproducible experiments and monitor whether performance remains reliable across locations and seasons. The fi-nal version of this paper should include the actual dataset description, implementation details, measured results, genuine screenshots, and any additional verified references.

## REFERENCES

1. M. M. Hassan et al., “Machine Learning-Based Rainfall Pre-diction: Unveiling Insights and Forecasting for Improved Pre-paredness,” IEEE Access, vol. 11, pp. 132196132222,2023,doi:10.1109/ACCESS.2023.3338876

© 2026 IJSRET