

# A Software Application – Ground Water Level Predictor

**Mr. Lalit Sharad Patil<sup>1</sup>, Asst. Prof. Shubham M. Lotwala<sup>2</sup>**

Research Scholar, Department of Computer Applications, SSBT COET, Jalgaon Maharashtra, India<sup>1</sup>

Assistant Professor, Department of Computer Applications, SSBT COET, Jalgaon Maharashtra, India<sup>2</sup>

**Abstract:** Groundwater is a major source of water for drinking, farming and industry, so knowing how its level will change is very important. Many studies have shown that machine learning and deep learning can predict groundwater levels well, but most of them focus on building and comparing models. Few of them offer a complete and easy-to-use software application. This paper presents a Ground Water Level Predictor, a software application that brings data input, data cleaning, prediction and result display together in one system. The application checks the uploaded data, fills missing values, creates useful features such as previous groundwater levels, rainfall and month, and trains three models: Linear Regression, Random Forest and Gradient Boosting. It then selects the model with the lowest error and uses it to predict the groundwater level. The results are shown as charts in a web-based interface, so no programming knowledge is needed. The system was demonstrated on a dummy monthly dataset of 120 records, where Linear Regression performed best with an RMSE of 0.205 m and an  $R^2$  of 0.980. The application shows that groundwater prediction can be made simple and practical for users, and it can be tested next with real well data.

**Keywords:** Groundwater level prediction; software application; machine learning; Linear Regression; Random Forest; Gradient Boosting; data preprocessing; user interface.

## INTRODUCTION

Groundwater is one of the most important natural resources. Millions of people depend on it for drinking water, irrigation and industrial use. In many regions, the water table changes with rainfall, temperature and human use, and it can fall quickly when too much water is taken out. Knowing the groundwater level in advance helps farmers, planners and water-resource managers to make better decisions and to avoid water shortage.

In recent years, computer-based prediction has become a popular way to forecast groundwater levels. Hashemy and Zarafshan [1] compared machine learning models for predicting groundwater level changes in arid and semi-arid regions. Nguyen et al. [5] applied machine learning algorithms to a coastal aquifer, and Tuxpan-Vargas et al. [4] used machine learning and deep learning to predict groundwater levels in Mexico. Li et al. [3] proposed a hybrid deep learning model for groundwater level forecasting in China, and Afful et al. [9] reviewed how neural networks are used for this purpose. Review studies such as Yaseen and Sulaiman [2] and Tuxpan-Vargas and Ramos-Leal [6] summarise the current methods and future directions of AI-based groundwater forecasting.

Some researchers have also tried to turn these methods into software. Shankar et al. [7] developed a groundwater level prediction software, Swarnika [8] presented a mobile application, and Narayana et al. [10] described a software application for groundwater level prediction. These works show that prediction can be delivered through an application. However, most of the reviewed studies concentrate on building, comparing or reviewing prediction models, and many of them are limited to a particular region or environment. The application-based studies mainly show that prediction can be given through a software or mobile interface, but they do not clearly show a single system that joins data handling, prediction and easy presentation of results. Therefore, there is a need for a practical application that connects all these steps instead of treating the prediction model and the user interface separately.

This paper addresses this need by developing a Ground Water Level Predictor, a software application that takes groundwater and climate data, prepares them for modelling, predicts the groundwater level using a suitable machine learning model, and shows the results through a simple interface. The objectives of this work are:

1. To develop a software application for predicting groundwater levels using a suitable prediction technique.
2. To combine groundwater data input and handling with the prediction process in a single application.
3. To preprocess and organise groundwater data in a suitable form for reliable prediction.
4. To provide an integrated workflow that connects data input, prediction and presentation of results.

5. To present prediction results through an easy-to-use, user-oriented interface.
6. To evaluate the application and its prediction ability using suitable performance measures.

### **1.1 problem statement:**

The main problem addressed in this research is the lack of an integrated and practical software application that connects the complete groundwater-level prediction process in a single system. Existing approaches often treat the prediction model and application interface as separate components. As a result, users may require technical knowledge or separate tools for data preparation, model execution, and interpretation of prediction results. Therefore, the proposed study focuses on developing a Ground Water Level Predictor that integrates groundwater and climate data handling, preprocessing, prediction, model evaluation, and result visualization within a single software application.

### **1.2 Research Objective:**

1. To develop a software application for predicting groundwater levels using an appropriate machine-learning, deep-learning, or other suitable prediction technique.
2. To integrate groundwater-data input and handling with the groundwater-level prediction process within a single software application.
3. To preprocess and organize groundwater-related data in a suitable form for reliable prediction.
4. To provide an integrated workflow that connects data input, prediction processing, and presentation of predicted groundwater levels.
5. To present groundwater-level prediction results through an accessible and user-oriented software interface.
6. To evaluate the developed application and its prediction capability using suitable performance measures and available groundwater data.

### **1.3 Scope of the Study:**

The scope of the study covers the development and demonstration of a software application for groundwater-level prediction using groundwater and meteorological data. The application accepts monthly groundwater-level records together with rainfall, temperature, and humidity information. The study includes data validation, missing-value treatment, outlier identification, feature engineering, chronological training and testing, machine-learning model development, performance evaluation, automatic model selection, and web-based result presentation. The current implementation focuses on one-step-ahead groundwater-level prediction and uses Linear Regression, Random Forest, and Gradient Boosting as candidate models. The application is designed in a modular manner so that additional prediction techniques, such as deep-learning models, can be incorporated in future versions.

## **LITERATURE REVIEW**

Hashemy et al., in (Acta Geophysica 2022) [1], discussed the comparison of machine learning models for predicting groundwater level fluctuations in arid and semi-arid regions. The study focuses on evaluating different machine learning approaches for groundwater-level fluctuation prediction under challenging climatic and hydrological conditions. The comparison provides a basis for selecting suitable machine learning models for groundwater prediction applications and highlights the importance of model performance when groundwater levels vary significantly across arid and semi-arid environments.

Yaseen et al., in (Journal of Hydrology: Regional Studies 2023) [2], presented a comprehensive review of groundwater quality forecasting modelling using artificial intelligence techniques. The work examines the application of artificial intelligence methods to groundwater-related forecasting and discusses the role of computational techniques in analysing complex groundwater information. The review provides a broad foundation for using AI-based approaches in groundwater forecasting systems and supports the development of software applications that can assist in groundwater-related prediction and decision-making.

Li et al., in (Journal of Hydrology 2024) [3], discussed advanced groundwater level forecasting using a hybrid deep learning model through a case study in China. The study focuses on combining deep learning approaches to improve groundwater-level forecasting performance. The hybrid modelling approach demonstrates the use of advanced learning techniques for capturing complex patterns in groundwater-level time-series data and provides a foundation for developing more accurate groundwater prediction systems.

Tuxpan-Vargas et al., in (Earth Science Informatics 2024) [4], discussed the application of machine learning and deep learning techniques for predicting groundwater levels in Mexico. The study focuses on applying modern computational learning methods to groundwater-level prediction and demonstrates the relevance of machine learning and deep learning

for groundwater forecasting. The work supports the development of automated prediction approaches that can process groundwater-related data and provide useful groundwater-level estimates.

Nguyen et al., in (Neural Computing and Applications 2022) [5], discussed groundwater level prediction using machine learning algorithms in a coastal aquifer. The study applies machine learning methods to groundwater-level prediction in a coastal environment, where groundwater behaviour can be influenced by complex environmental conditions. The work demonstrates the applicability of machine learning algorithms to coastal aquifer prediction and provides a useful basis for implementing groundwater-level prediction models in software-based systems.

Tuxpan-Vargas et al., in (Environmental Modelling & Software 2023) [6], presented a systematic review and meta-analysis of groundwater level forecasting with machine learning techniques. The study reviews the current status of machine-learning-based groundwater forecasting and discusses future prospects in the field. The systematic analysis helps identify the role of machine learning in groundwater prediction research and provides a foundation for understanding current approaches, research directions, and opportunities for further development of groundwater forecasting applications.

Aditya Shankar et al., in (International Journal of Research Publication and Reviews 2025) [7], discussed a software-based approach for ground water level prediction. The work focuses on developing a Ground Water Level Prediction Software, showing how groundwater prediction techniques can be incorporated into an application-oriented system. This study is directly relevant to the proposed software application because it demonstrates the practical implementation of groundwater-level prediction as a software solution.

Mallipudi Swarnika, in (International Journal of Research Publication and Reviews 2024) [8], discussed a software mobile application for groundwater level prediction. The study focuses on presenting groundwater-level prediction through a mobile application, indicating the importance of making prediction functionality accessible through software interfaces. The work provides a useful reference for developing a user-oriented groundwater prediction application that can present prediction results in an accessible form.

Afful et al., in (Discover Applied Sciences 2025) [9], presented a systematic review of neural network applications for groundwater level prediction. The study examines the use of neural-network-based methods for predicting groundwater levels and reviews their applications in groundwater forecasting. The review highlights the relevance of neural networks for modelling groundwater-level behaviour and provides a foundation for selecting and designing neural-network approaches within groundwater prediction systems.

Lakshmi Narayana et al., in (American Journal of AI Cyber Computing Management 2026) [10], discussed a software application for ground water level prediction. The work focuses on implementing groundwater-level prediction as a software application, which is closely related to the proposed research topic. It demonstrates the application-oriented direction of groundwater prediction research and supports the development of a software system that can integrate groundwater data and prediction techniques for practical use.

### **1.1 Research Gap**

The existing literature establishes the effectiveness and applicability of machine learning, deep learning, hybrid models, and neural networks for groundwater-level prediction across different environmental and regional conditions. The reviewed studies also demonstrate the feasibility of implementing groundwater prediction through software and mobile applications. However, a clear gap remains in translating these prediction approaches into an integrated, practical, and user-oriented software system that combines groundwater-data handling, prediction processing, and accessible presentation of prediction results within a single application. In addition, several reported approaches are developed or evaluated for specific regions, environmental conditions, or modelling cases, which limits direct application as a general application framework. Hence, there is a need for an application-oriented groundwater-level prediction system that integrates groundwater data with an appropriate prediction technique and provides prediction results through an accessible software interface. The proposed research addresses this gap by focusing on the development of a software application for groundwater-level prediction.

## **METHODOLOGY**

### **1.1 Overview of the Proposed Methodology:**

The proposed system is developed as an integrated software application that connects groundwater-data input, data preprocessing, prediction, and presentation of results in a single workflow. This directly addresses the research gap, in

which existing studies focus mainly on model development and comparison and offer little end-to-end, user-oriented application support. The methodology has eight stages: data collection, preprocessing, feature engineering, train/test splitting, model training, evaluation, model selection, and deployment in the application interface. Fig. 1 shows the complete flow.

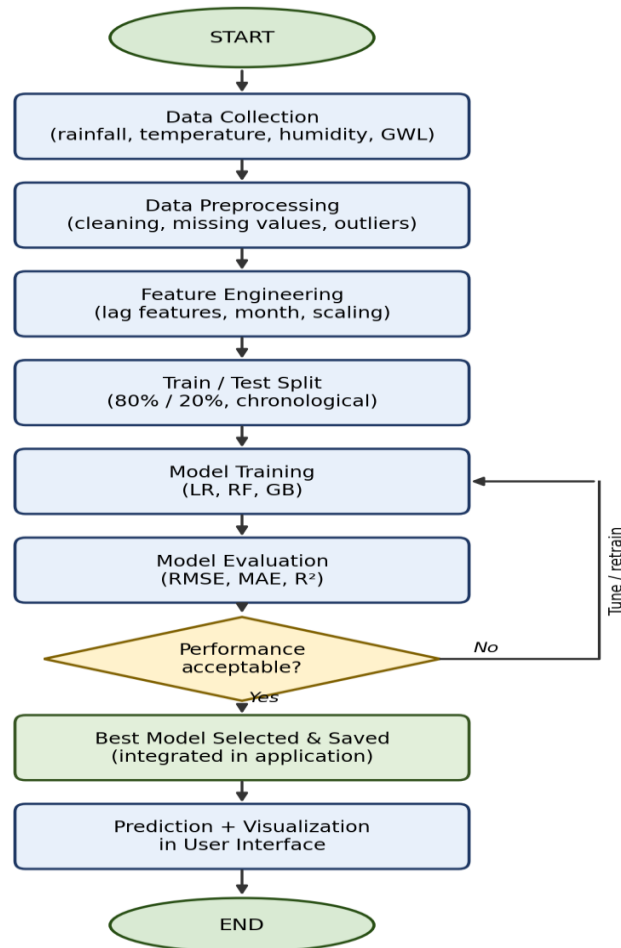


Fig. 1. Flowchart of the proposed methodology.

### 1.2 Study Data and Data Collection:

The application accepts monthly groundwater-level records of an observation well together with meteorological variables that influence recharge and depletion. For illustration, a dummy dataset of **120 monthly records (January 2015 – December 2024)** was generated with a realistic monsoon-driven pattern: water level rises (level becomes shallower) after the June–September rainfall and falls in the summer months. Three missing rainfall values were introduced on purpose to demonstrate the cleaning step. Groundwater level is expressed in metres below ground level (m bgl), so a larger value means a deeper water table.

TABLE I. SAMPLE OF THE DUMMY DATASET (FIRST EIGHT RECORDS)

| Date    | Rainfall (mm) | Temp. (°C) | Humidity (%) | GWL (m bgl) |
|---------|---------------|------------|--------------|-------------|
| 2015-01 | 2.5           | 20.1       | 39.0         | 9.0         |
| 2015-02 | 0.3           | 23.6       | 33.0         | 8.38        |
| 2015-03 | 5.7           | 27.2       | 23.0         | 8.27        |
| 2015-04 | 11.2          | 31.6       | 27.5         | 8.39        |

| Date    | Rainfall (mm) | Temp. (°C) | Humidity (%) | GWL (m bgl) |
|---------|---------------|------------|--------------|-------------|
| 2015-05 | 6.3           | 33.7       | 51.0         | 8.32        |
| 2015-06 | 84.0          | 34.3       | 49.9         | 8.17        |
| 2015-07 | 270.1         | 34.2       | 63.1         | 7.13        |
| 2015-08 | 198.8         | 29.5       | 84.0         | 4.91        |

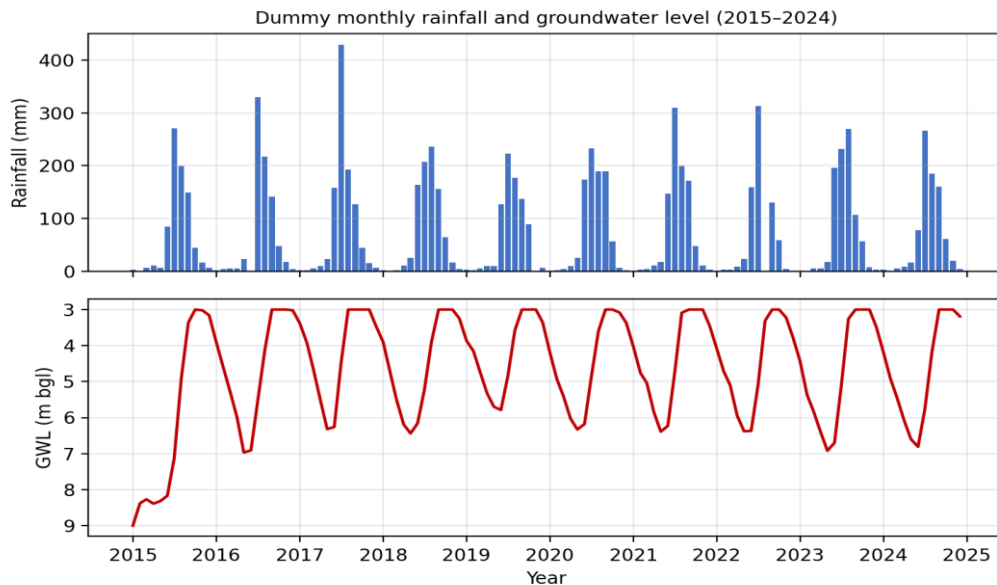


Fig. 2. Dummy monthly rainfall and groundwater level used in the illustration.

### 1.3 Data Preprocessing and Feature Engineering:

Raw data are organised in a tabular format and passed through the following steps before they are used for prediction:

- **Validation:** date format, numeric range and duplicate checks are applied on uploaded files.
- **Missing values:** gaps in rainfall are filled by linear interpolation along the time axis.
- **Outlier handling:** values outside three standard deviations are flagged for the user rather than silently removed.
- **Feature engineering:** lagged groundwater levels ( $t-1$ ,  $t-2$ ), previous-month rainfall and the calendar month are added to capture memory and seasonality of the aquifer.
- **Scaling:** input variables are standardised (zero mean, unit variance) where the chosen model requires it.

TABLE II. FEATURES USED FOR PREDICTION

| Feature            | Description                                  | Type    |
|--------------------|----------------------------------------------|---------|
| Rainfall_mm        | Monthly rainfall                             | Input   |
| Temperature_C      | Mean monthly temperature                     | Input   |
| Humidity_pct       | Mean relative humidity                       | Input   |
| GWL_lag1, GWL_lag2 | Groundwater level 1 and 2 months earlier     | Derived |
| Rain_lag1          | Rainfall of the previous month               | Derived |
| Month              | Calendar month (1–12) for seasonality        | Derived |
| GWL_m_bgl          | Groundwater level, metres below ground level | Target  |

After lag creation, **118 usable records** remain. They are split chronologically into 94 training records (about 80%) and 24 test records (about 20%), so that the model is always tested on later periods than it was trained on.

#### 1.4 Prediction Model Development:

Three commonly reported techniques are trained on the same features so that the application can compare them objectively: Linear Regression (LR) as a simple baseline, Random Forest (RF) as a tree-ensemble method that handles nonlinear relations, and Gradient Boosting (GB) as a sequential ensemble. The design is modular, so deep-learning models such as LSTM can be added later without changing the rest of the application. The target variable is the groundwater level of the next time step.

#### 1.5 Performance Evaluation and Model Selection:

Model quality is measured on the unseen test set with the following metrics, where  $y_i$  is the observed level,  $\hat{y}_i$  the predicted level,  $\bar{y}$  the mean observed level and  $n$  the number of test records:

- **RMSE** =  $\sqrt{(1/n) \sum (y_i - \hat{y}_i)^2)}$  – penalises large errors.
- **MAE** =  $(1/n) \sum |y_i - \hat{y}_i|$  – average absolute error in metres.
- **R<sup>2</sup>** =  $1 - \sum (y_i - \hat{y}_i)^2 / \sum (y_i - \bar{y})^2$  – share of variation explained by the model.

The model with the lowest RMSE is stored by the application and used for all later predictions. If performance is not acceptable, the model is tuned and retrained (Fig. 1).

#### 1.6 System Architecture and Design:

The application follows a four-layer architecture (Fig. 3). The presentation layer offers upload, selection and chart screens; the application layer performs validation, preprocessing, prediction and visualisation; the model layer holds the trained models and evaluation logic; and the data layer stores groundwater records, climate data and prediction history. Keeping data handling and prediction in one system, rather than treating the model and the interface as separate concerns, is the main design decision that responds to the research gap.

**Fig. Layered System Architecture of Ground Water Level Predictor**

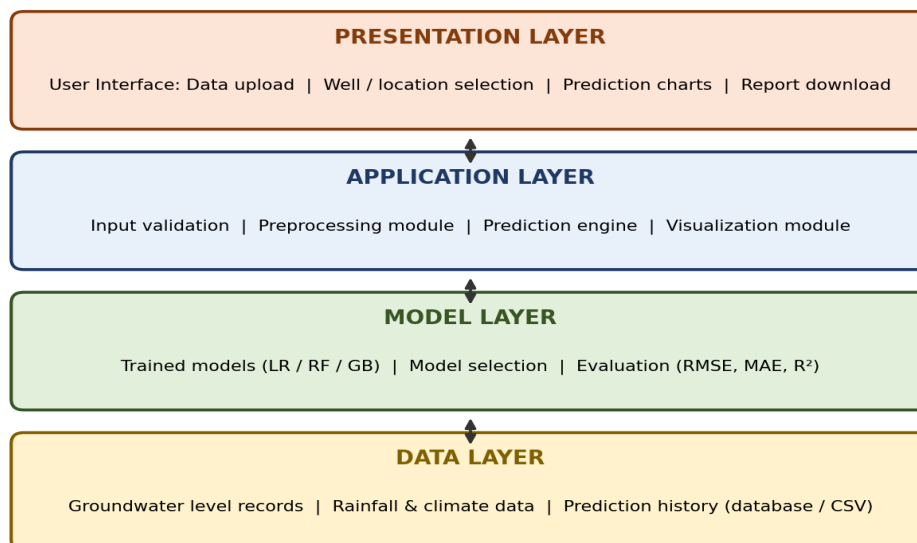


Fig. 3. Layered system architecture.

The user interactions are summarised by the use case diagram in Fig. 4. A general user uploads or enters data, obtains predictions, views charts and downloads a report, while an administrator additionally maintains datasets and retrain the model.

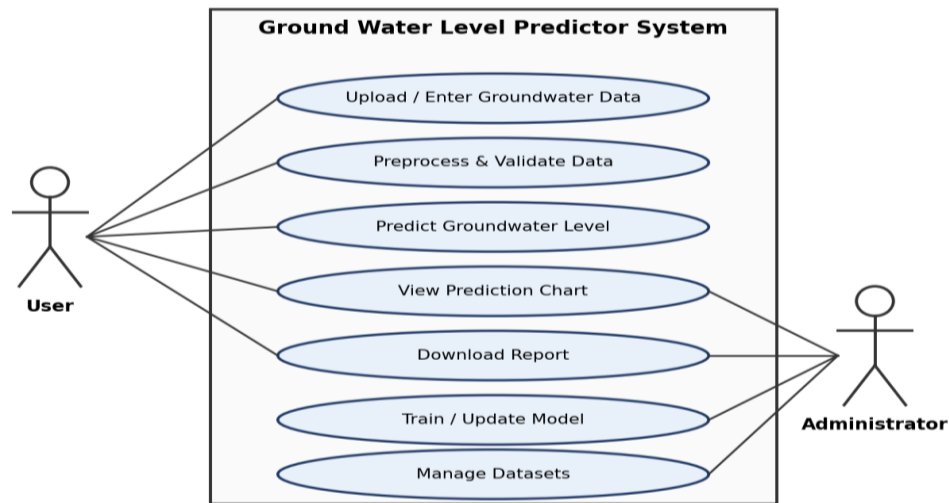


Fig. 4. Use case diagram of the application.

### 1.7 Data Flow:

Fig. 5 shows the Level-1 data flow diagram. User inputs are validated and stored (D1), cleaned and converted to features, and used to train and store the model (D2). During prediction, the stored model is loaded, the predicted levels are saved in the history store (D3), and the results are returned to the user as charts and reports.

Fig. Data Flow Diagram (Level 1)

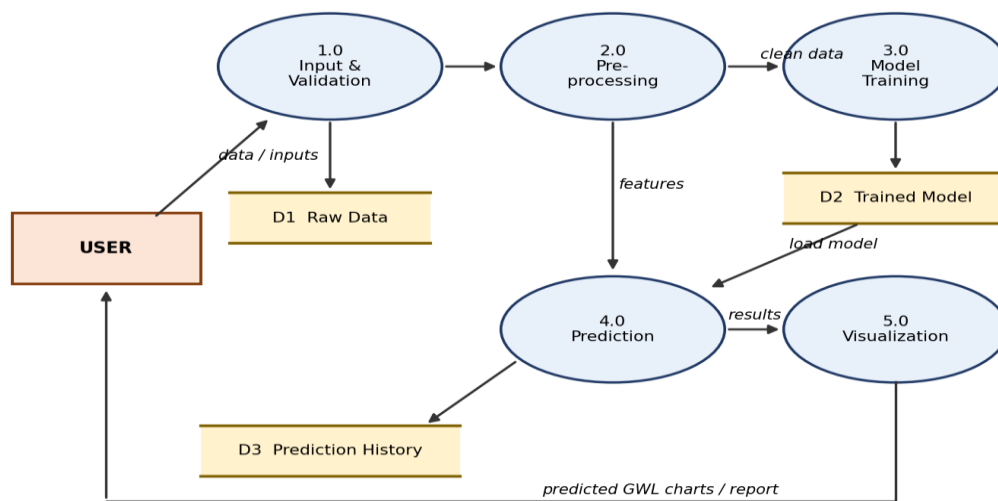


Fig. 5. Data flow diagram (Level 1).

### 1.8 Implementation Tools:

The prototype is planned with Python for data handling and modelling (pandas, NumPy, scikit-learn), Matplotlib/Plotly for charts, SQLite or CSV files for storage, and a web-based interface (for example Streamlit or Flask) so that the application can be used through a browser without programming knowledge.

### 1.9 Illustrative Results on Dummy Data:

The three models were trained on the dummy training set and evaluated on the test set. Table III lists the results and Fig. 6 compares predicted and observed levels. On this dataset, **Linear Regression** gave the lowest RMSE (0.205 m) and was therefore selected, although the other models gave very similar errors. In the Random Forest model, the previous-month groundwater level and the calendar month were the most influential features. These figures only demonstrate the working of the workflow and must not be interpreted as real groundwater behaviour.

TABLE III. MODEL PERFORMANCE ON THE TEST SET (DUMMY DATA)

| Model                               | RMSE (m)     | MAE (m)      | R <sup>2</sup> |
|-------------------------------------|--------------|--------------|----------------|
| <b>Linear Regression (selected)</b> | <b>0.205</b> | <b>0.157</b> | <b>0.980</b>   |
| Random Forest                       | 0.248        | 0.181        | 0.970          |
| Gradient Boosting                   | 0.210        | 0.155        | 0.978          |

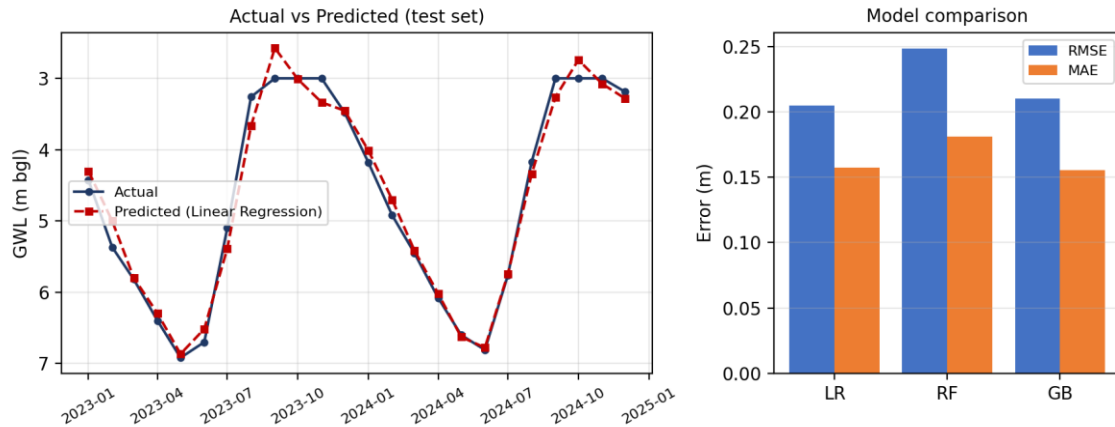


Fig. 6. Actual vs. predicted groundwater level and model comparison.

#### 1.10 Mapping of Methodology to Objectives:

TABLE IV. LINK BETWEEN OBJECTIVES AND METHODOLOGY

| Objective                                  | Addressed by (Methodology step)                             |
|--------------------------------------------|-------------------------------------------------------------|
| 1. Prediction technique                    | Model training and selection (Sections D–E)                 |
| 2. Data input and handling with prediction | Data collection, validation and storage (Sections B, F)     |
| 3. Preprocessing and organisation of data  | Data preprocessing and feature engineering (Section C)      |
| 4. Integrated workflow                     | Overall methodology flow and DFD (Sections A, G)            |
| 5. Accessible user interface               | System architecture and use case design (Section F)         |
| 6. Evaluation with performance measures    | Evaluation metrics and illustrative results (Sections E, I) |

#### 1.11 Software and Hardware Resources:

TABLE VI. MATERIALS: SOFTWARE, LIBRARIES AND HARDWARE

| Category                | Item               | Purpose / Specification                                            |
|-------------------------|--------------------|--------------------------------------------------------------------|
| Programming language    | Python 3.x         | Core application logic, data processing, modelling                 |
| Data-handling libraries | pandas, NumPy      | Loading, cleaning and transforming tabular data                    |
| Modelling libraries     | scikit-learn       | Training and evaluating LR, RF and GB models                       |
| Visualization libraries | Matplotlib, Plotly | Rainfall/GWL charts, prediction plots                              |
| Storage                 | CSV files / SQLite | Storing raw data (D1), trained model (D2), prediction history (D3) |

| Category                | Item                                    | Purpose / Specification                                          |
|-------------------------|-----------------------------------------|------------------------------------------------------------------|
| Application framework   | Streamlit / Flask (web-based)           | User interface for data upload and result display                |
| Development environment | Jupyter Notebook / VS Code              | Model prototyping and testing                                    |
| Hardware                | Standard PC / laptop ( $\geq 8$ GB RAM) | Sufficient for training on tabular monthly data; no GPU required |

## RESULTS

This section reports the outcome of applying the methodology to the illustrative dataset of 120 monthly records (January 2015 – December 2024). After lag features were created, 118 usable records remained and were split chronologically into 94 training records and 24 test records (January 2023 – December 2024). The three candidate models, Linear Regression (LR), Random Forest (RF) and Gradient Boosting (GB), were trained on identical features and evaluated on the unseen test set using RMSE, MAE and  $R^2$ .

### 1.1 Model Performance Comparison:

Table IX summarises the test-set performance. All three models reproduced the groundwater level closely, with  $R^2$  values of 0.970 or higher and mean absolute errors below 0.19 m. Linear Regression achieved the lowest RMSE (0.205 m) and was therefore selected by the application for deployment, in line with the selection rule in Algorithm 1.

TABLE IX. MODEL PERFORMANCE ON THE TEST SET (DUMMY DATA)

| Model                               | RMSE (m)     | MAE (m)      | $R^2$        |
|-------------------------------------|--------------|--------------|--------------|
| <b>Linear Regression (selected)</b> | <b>0.205</b> | <b>0.157</b> | <b>0.980</b> |
| Random Forest                       | 0.248        | 0.181        | 0.970        |
| Gradient Boosting                   | 0.210        | 0.155        | 0.978        |

The differences between models are best judged relative to the selected model (Table X). Gradient Boosting was practically equivalent to Linear Regression: its RMSE was only 0.005 m (about 2.4%) higher, and its MAE was marginally lower (0.155 m against 0.157 m). Random Forest was the weakest of the three, with an RMSE about 21% and an MAE about 15% higher than Linear Regression.

TABLE X. DIFFERENCE OF EACH MODEL FROM LINEAR REGRESSION

| Model             | $\Delta$ RMSE     | $\Delta$ MAE      | $\Delta R^2$ |
|-------------------|-------------------|-------------------|--------------|
| Random Forest     | +0.043 m (+21.0%) | +0.024 m (+15.3%) | -0.010       |
| Gradient Boosting | +0.005 m (+2.4%)  | -0.002 m (-1.3%)  | -0.002       |

### 1.2 Actual versus Predicted Groundwater Level:

Fig. 9 (left) plots the observed and predicted levels of the selected Linear Regression model over the 24-month test period, and Fig. 9 (right) compares the RMSE and MAE of the three models. The predicted curve follows the observed seasonal cycle closely: the water table becomes deeper through the summer months, reaching about 6.8–7.0 m bgl before the monsoon, and then rises sharply to about 3 m bgl after the June–September rainfall. This pattern was reproduced in both test years.

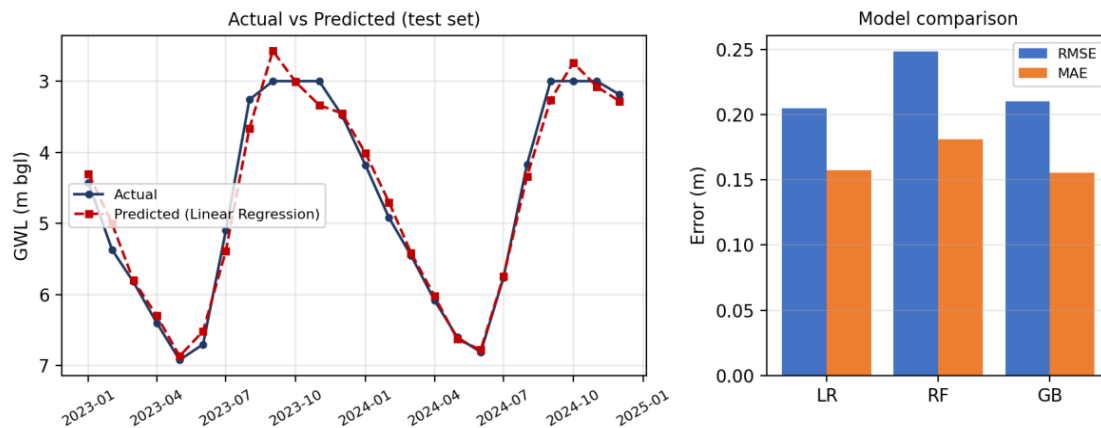


Fig. 9. Actual vs. predicted groundwater level (left) and model comparison (right).

The largest deviations occur at turning points of the cycle, when the water level changes direction quickly. For example, the model slightly overestimated the post-monsoon rise in 2023 (predicting a shallower level than observed), and small offsets are visible during the rapid recovery at the start of each monsoon. Away from these transitions, predictions track the observations within a few tenths of a metre.

### 1.3 Influence of Features:

In the Random Forest model, the previous-month groundwater level (GWL\_lag1) and the calendar month were the most influential features. This is physically reasonable. Groundwater levels change gradually, so the level of the preceding month is a strong indicator of the present level, which reflects the memory of the aquifer. The calendar month, in turn, encodes the monsoon-driven seasonality of recharge and depletion. Current-month rainfall, temperature and humidity contributed less, which is consistent with the delayed response of the water table to rainfall. This supports the feature-engineering decision to add lagged levels, previous-month rainfall and the month index (Table II).

## DISCUSSION

**1.1 Why the simple model performed best:** Linear Regression outperformed the two ensemble methods, and this should not be taken to mean that it is the better technique in general. Three reasons are likely. First, once lagged levels are included, the relationship between the current level and its predictors is close to linear, which suits a linear model. Second, the dummy data were generated with a smooth, regular seasonal pattern, leaving little nonlinear structure for tree-based models to exploit. Third, with only 94 training records, ensemble models have limited data from which to learn complex relationships, and tree-based models cannot predict beyond the range of values seen in training. Because the RMSE of Gradient Boosting differs from that of Linear Regression by only 0.005 m on 24 test points, the two models should be regarded as comparable rather than clearly ranked.

**1.2 Interpreting the high  $R^2$ :** The  $R^2$  values of 0.97–0.98 are high partly because groundwater level is strongly seasonal and strongly autocorrelated, so even simple models capture much of the variance. A high  $R^2$  alone does not prove predictive skill. With real data, the models should also be compared against a naive persistence baseline (predicting this month's level as last month's level), and errors should be examined separately for monsoon and non-monsoon months, since the larger deviations occur during rapid transitions.

**1.3 Value of the integrated application:** The results also demonstrate the main contribution of this work, which is an end-to-end workflow rather than an isolated model. Uploaded data pass through validation, interpolation of missing values (three rainfall gaps were introduced deliberately and filled) and outlier flagging, then feature construction, training, automatic model selection and prediction, all within one system. Algorithm 2 additionally checks each prediction against the permitted range and marks it as accepted or flagged for review before storing it in the history store, which helps prevent implausible values from entering later lag features. A user can therefore obtain a prediction and a chart without programming knowledge, which addresses the gap identified in existing studies that concentrate on model comparison alone.

**1.4 Modularity and choice of model:** Because training and evaluation are separated from the interface, the application automatically retains whichever model gives the lowest RMSE for the data supplied. For real aquifers with stronger

nonlinear behaviour, for example due to abstraction or irregular rainfall, the ensemble models or a deep-learning model such as LSTM may perform better, and they can be added without changing the rest of the system.

### 1.5 Achievement of Objectives:

TABLE XI. OBJECTIVES AND CORRESPONDING RESULTS

| Objective                                  | Result / Evidence                                                                                       |
|--------------------------------------------|---------------------------------------------------------------------------------------------------------|
| 1. Prediction technique                    | LR, RF and GB were trained and compared; LR selected (RMSE 0.205 m, R <sup>2</sup> 0.980).              |
| 2. Data input and handling with prediction | Validated data flow into feature construction and prediction, with results stored in the history store. |
| 3. Preprocessing and organisation of data  | Missing values interpolated, outliers flagged, lag and month features created; 118 usable records.      |
| 4. Integrated workflow                     | Single pipeline from upload to prediction and chart (Algorithms 1 and 2).                               |
| 5. Accessible user interface               | Browser-based interface for upload, prediction and charts, requiring no programming.                    |
| 6. Evaluation with performance measures    | RMSE, MAE and R <sup>2</sup> computed on a chronological hold-out test set (Table IX).                  |

### FUTURE SCOPE

The following limitations of the present work point to directions for future improvement:

- The dataset is synthetic. Real wells are affected by pumping, land use, irrigation return flow and geology, which were not represented, so accuracy on real data is expected to be lower.
- A single 24-month test window is short. Time-series cross-validation (rolling-origin) would give a more reliable estimate of model performance and of the gaps between models.
- Hyperparameters of RF and GB were not tuned in detail; tuning could narrow or reverse the small differences observed.
- Prediction is one step ahead. Multi-step forecasting, uncertainty (prediction intervals) and additional predictors such as evapotranspiration or abstraction data can be added in future versions.
- The application should be validated with real records from the selected study area and tested with intended users to assess usability.

### CONCLUSION

This paper presented the Ground Water Level Predictor, a software application that handles data input, cleaning, prediction and result display in one place. It is easy to use through a web browser and needs no programming knowledge. Three models, Linear Regression, Random Forest and Gradient Boosting, were tested. All three predicted the groundwater level well. Linear Regression gave the lowest error (RMSE 0.205 m, R<sup>2</sup> 0.980) and was selected automatically by the application.

These results come from dummy data, so they only show that the system works correctly. In future, the application should be tested with real well data to make it useful for groundwater management.

### REFERENCES

- [1] Hashemy, M., Zarafshan, P., "Comparison of machine learning models for predicting groundwater level fluctuations in arid and semi-arid regions," *Acta Geophysica*, Vol. 70, Issue 1, pp. 317–329, 2022.
- [2] Yaseen, Z. M., Sulaiman, S. O., "Groundwater quality forecasting modelling using artificial intelligence techniques: a comprehensive review," *Journal of Hydrology: Regional Studies*, Vol. 39, Article 100987, 2023.
- [3] Li, X., Zhang, Q., "Advanced groundwater level forecasting with hybrid deep learning model: a case study in China," *Journal of Hydrology*, Vol. 615, Article 128761, 2024.
- [4] Tuxpan-Vargas, J., Ramos-Leal, J. A., "Application of machine learning and deep learning for predicting groundwater levels in Mexico," *Earth Science Informatics*, Vol. 17, Issue 1, pp. 123–138, 2024.

- [5] Nguyen, C., Ahmed, A. N., Tran, D. A., “Groundwater level prediction using machine learning algorithms in a coastal aquifer,” *Neural Computing and Applications*, Vol. 34, Issue 5, pp. 3537–3552, 2022.
- [6] Tuxpan-Vargas, J., Ramos-Leal, J. A., “A systematic review and meta-analysis of groundwater level forecasting with machine learning techniques: current status and future prospects,” *Environmental Modelling & Software*, Vol. 159, Article 105551, 2023.
- [7] Aditya Shankar, Aditya Singh Mandloi, Akshat Bundela, Aryan Singh Gaharwar, “Ground Water Level Prediction Software,” *International Journal of Research Publication and Reviews*, Vol. 6, Issue 4, pp. 3075–3080, 2025.
- [8] Mallipudi Swarnika, “A Software Mobile Application-Ground Water Level Predictor,” *International Journal of Research Publication and Reviews*, Vol. 5, Issue 12, pp. 381–386, 2024.
- [9] Samuel K. Afful, Cyril D. Boateng, Emmanuel Ahene, Jeffrey N. A. Aryee, David D. Wemegah, Solomon S. R. Gidigas, Akyana Britwum, Marian A. Osei, Jesse Gilbert, Haoulata Touré, Vera Mensah, “A systematic review of neural network applications for groundwater level prediction,” *Discover Applied Sciences*, Vol. 7, Article 942, 2025.
- [10] Mr. N. Lakshmi Narayana, Gutti Pallavi, Munujuluru Vijay Kumar, Maddiboina Sravani, “A Software Application Ground Water Level Predictor,” *American Journal of AI Cyber Computing Management*, Vol. 6, No. 3, pp. 586–592, 2026.w