Vietnam National University - Ho Chi Minh City
University of Information Technology
Faculty of Computer Science

FINAL PROJECT REPORT

Data Mining and Application - CS313

Project:

Drought Forecasting based on Soil & Weather Data

Instructor:

PhD. Vo Nguyen Le Duy

Group Members:

Le Van Hoang	22520465
Ha Huy Hoang	22520460
Nguyen Duy Hoang	22520467
Dang Vinh Hoi	22520490
Pham Duc Huy Hoang	22520474
Thach Minh Hoang	22520477

Ho Chi Minh City, May 2025

Contents

1. Introduction

1.1. Definition and Impact of Drought

Drought is a prolonged period of below-average precipitation that leads to reduced soil moisture, lower groundwater levels, and diminished surface water, negatively affecting agriculture, ecosystems, and human livelihoods. Unlike sudden disasters, drought develops slowly, with impacts emerging weeks or months later and possibly lasting years [1].

Key consequences include:

- **Agriculture:** Crop failures due to irrigation shortages.
- **Water supply:** Limited fresh water for daily and industrial use.
- **Environment:** Land degradation, forest loss, biodiversity decline.
- **Socio-economic:** Income loss, unemployment, and economic damage.

1.2. The Importance of Drought Forecasting

Reliable drought prediction is essential for early warning, agricultural planning, and water management. Medium-term forecasting (e.g., six weeks ahead) is particularly valuable, though challenging due to trade-offs between short-term practicality and long-term accuracy.

1.3. Problem Statement and Research Objective

This study focuses on predicting drought severity using meteorological time-series and static soil/topographic data across U.S. regions. The objective is to forecast drought levels (scale 0–5) for the next six weeks. The model is trained and validated on U.S. data, with potential for real-time and global deployment in the future.

2. Dataset

2.1. Feature Description

The dataset includes both static and time-series features that describe the geographical, environmental, and climatic characteristics of each region.

2.1.1. Static Features

Topographic features include average elevation (`Elevation`) and slope classification (`Slope1` to `Slope8`), representing terrain steepness from flat to mountainous areas.

Land use features describe the proportion of land occupied by different types, including water bodies (`WAT_LAND`), barren or sparsely vegetated land (`NVG_LAND`), urban areas (`URB_LAND`), grasslands or shrublands (`GRS_LAND`), forests (`FOR_LAND`), total cultivated land (`CULT_LAND`), and irrigated cropland using surface or groundwater (`CULTRF_LAND`, `CULTIR_LAND`).

Soil quality is represented by seven indicators (`SQ1` to `SQ7`) that reflect properties such as nutrient retention, aeration, salinity, toxicity, and overall suitability for agriculture.

2.1.2. Time-Series Features

Each daily record includes:

- **Location data:** `fips` (region ID), `lat`, `lon`, `date`, and `flip` (technical flag for sequence tracking).
- **Meteorological variables:**
  - `PRECTOT` – Daily precipitation (mm/day)
  - `PS` – Surface pressure (kPa)
  - `QV2M` – Specific humidity at 2 meters (g/kg)

  - `T2M`, `T2MDEW`, `T2MWET` – Air temperature, dew/frost point, and wet bulb temperature at 2 meters (°C)
  - `T2M_MAX`, `T2M_MIN`, `T2M_RANGE` – Daily max, min, and range of temperature (°C)
  - `TS` – Earth skin temperature (°C)
- **Wind speed:**
  - At 10 meters: `WS10M`, `WS10M_MAX`, `WS10M_MIN`, `WS10M_RANGE`
  - At 50 meters: `WS50M`, `WS50M_MAX`, `WS50M_MIN`, `WS50M_RANGE`
- **Drought severity label:** `score` indicates drought level on a scale from 0 (no drought) to 5 (exceptional drought).

2.2. Training and Evaluation Dataset

The training and evaluation data are based on a publicly available Kaggle dataset that contains over 23 million daily meteorological records across 3,108 regions in the U.S. from 2000 to 2020 [2, 3]. Additionally, the dataset includes soil features such as slope, aspect... The time series dataset has been split as follows:

- **Training set:** 2000–2017 (after merging with validation set)
- **Testing set:** 2018–2020 (held out for final model evaluation)

This dataset includes the weekly drought severity labels (`score`) as ground truth for supervised learning.

2.3. Demonstration Dataset

For real-world deployment and demonstration (e.g., on recent data from the past 175 days), a separate dataset is constructed using the latest available data from external sources:

- **Static data source:** Harmonized World Soil Database and Global Terrain Data [4]
- **Time-series data source:** NASA POWER Project API [5]
- **Drought severity labels:** U.S. Drought Monitor archive [6]

Note that the demonstration dataset may not include drought severity labels in real time and is primarily used for model inference.

3. Exploratory Data Analysis and Preprocessing

3.1. Time-Series Data

The time-series data, consisting of daily meteorological variables, presents several notable characteristics:

- **Missing values:** The `score` column, which represents drought severity, contains over 85% missing values. This is due to the index being reported only weekly, specifically every Tuesday. To fill in the missing values, we used linear interpolation via `scipy.interpolate.interp1d`:

```
from scipy.interpolate import interp1d

f = interp1d(known_timestamps, known_scores,
    kind='linear', fill_value='extrapolate', bounds_error=False)
```

- **Outliers:** Natural weather data is inherently noisy and contains outliers in most features. To mitigate their impact without distorting the distribution, we applied the RobustScaler, which scales data using the median and interquartile range (IQR), making it robust to extreme values.
- **Temporal encoding:** Since weather patterns exhibit strong yearly seasonality, we encoded the day-of-year cyclically to help the model learn periodic trends:

$$\sin\left(\frac{2\pi \cdot \text{day}}{365}\right), \quad \cos\left(\frac{2\pi \cdot \text{day}}{365}\right)$$

3.2. Static Soil and Terrain Data

The static dataset includes soil properties, elevation, and land cover characteristics. These features do not vary over time but also contain outliers due to natural heterogeneity across different regions. We applied RobustScaler to normalize the data while preserving the effect of informative extreme values.

4. Model Development

4.1. Architecture Overview

Each training sample consists of two components:

- **Time series data** $\mathbf{X}_{\text{time}} \in \mathbb{R}^{175\times 21}$: represents 175 consecutive days, each with 21 meteorological features.
- **Static features** $\mathbf{X}_{\text{static}} \in \mathbb{R}^{29}$: includes time-invariant properties such as soil characteristics.

The static vector is passed through a small two-layer MLP with ReLU activations, producing a 16-dimensional hidden representation. Meanwhile, the time series is fed into one of four models: GRU, LSTM, LSTM with Attention, or Transformer. The sequence representation is then concatenated with the static embedding and passed through a fully connected head to produce the output $\hat{\mathbf{y}} \in \mathbb{R}^6$, which predicts drought levels for the next 6 weeks.

4.2. GRU Architecture

Gated Recurrent Unit (GRU) [7] is a simplified recurrent neural network architecture that retains temporal information via gating mechanisms. Compared to LSTM, GRU has a simpler structure as it removes the separate memory cell and instead merges it with the hidden state. GRU uses two gates:

- **Update gate** z_t: controls how much of the past information to retain.
- **Reset gate** r_t: controls how much of the past information to forget.

This simplified structure leads to faster training while still maintaining comparable performance to LSTM in many tasks. The GRU hidden state update equations are:

$$
\begin{aligned}
z_t &= \sigma(W_z x_t + U_z h_{t-1}) \\
r_t &= \sigma(W_r x_t + U_r h_{t-1}) \\
\tilde{h}_t &= \tanh(W_h x_t + U_h(r_t \odot h_{t-1})) \\
h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t
\end{aligned}
$$

A multi-layer GRU is used, and the final hidden state h_{175} is extracted as the sequence representation.

4.3. LSTM Architecture

Long Short-Term Memory (LSTM) [8] is a more expressive recurrent network that incorporates a separate memory cell c_t in addition to the hidden state h_t. It uses three gates:

- **Forget gate** f_t: decides what information to discard from the memory.
- **Input gate** i_t: determines which new information to store.
- **Output gate** o_t: determines what part of the memory to output as hidden state.

In contrast to GRU, LSTM separates memory and output mechanisms, which allows it to model longer dependencies more robustly, at the cost of increased complexity and more parameters.

The LSTM equations are:

$$
\begin{aligned}
f_t &= \sigma(W_f x_t + U_f h_{t-1}) \\
i_t &= \sigma(W_i x_t + U_i h_{t-1}) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1}) \\
\tilde{c}_t &= \tanh(W_c x_t + U_c h_{t-1}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

We use a two-layer LSTM, and the final hidden state h_{175} is taken as the sequence representation.

4.4. LSTM with Attention

Instead of using only the final hidden state, this model applies attention over all LSTM outputs $(h_1, h_2, \ldots, h_{175})$ to form a context vector [9]. Attention weights α_t indicate the relevance of each time step:

$$\alpha_t = \frac{\exp(w^\top h_t)}{\sum_{k=1}^{T} \exp(w^\top h_k)}$$

The context vector is computed as:

$$\text{context} = \sum_{t=1}^{175} \alpha_t h_t$$

4.5. Transformer Architecture

The Transformer architecture uses a self-attention mechanism to model dependencies across the entire sequence, allowing the model to capture long-term relationships without relying on sequential processing. The main steps in the Transformer model are as follows:

1. Each input $x_t \in \mathbb{R}^{21}$ is projected into a higher-dimensional space $\mathbb{R}^{128}$ via a linear embedding layer.
2. Learnable positional encodings are added to the embedded sequence to preserve the temporal order of the data.
3. The embedded sequence with positional encodings is passed through multiple Transformer encoder layers, which capture dependencies across time steps in the sequence.
4. The output sequence $(z_1, \ldots, z_{175})$ from the Transformer encoder is averaged over time (along the seq_len dimension) to create a global context vector:

$$z = \frac{1}{175} \sum_{t=1}^{175} z_t$$

This final context vector is then combined with static features and passed through fully connected layers to produce the model's final output.

4.6. Final Prediction

The final sequence representation (from GRU, LSTM, attention, or Transformer) is concatenated with the static embedding. This combined vector is passed through a three-layer fully connected network to predict the output $\hat{\mathbf{y}} \in \mathbb{R}^6$. This design enables the model to jointly learn from both temporal and static features.

5. Results and Evaluation

The experimental setup includes the following configurations:

- **Evaluation Dataset:** Data from 2018 to the end of 2020 [3].
- **Fixed Jump Step Parameter:** A step size of 7 for sampling the evaluation set.
- **Loss Function:** Mean Squared Error (MSELoss), optimized using AdamW with a OneCycleLR schedule over 15 epochs.

The drought score index used for training was obtained from the U.S. Drought Monitor, which is available only for the United States. This score is not readily applicable to other geographic regions. Therefore, we also explored models that exclude the drought score from input features, focusing solely on globally available weather and soil data. These inputs were sourced from NASA POWER [5] and the Harmonized World Soil Database [4].

5.1. Forecast Performance Using Drought Score

We first trained models using the drought score as an input feature. Table 1 reports the Mean Absolute Error (MAE) for weekly drought level predictions across various neural network architectures.

Model	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Average
GRU	**0.0918**	0.1456	0.1906	0.2357	0.2730	0.3102	0.2078
LSTM	0.0922	**0.1432**	**0.1881**	0.2338	0.2686	0.3075	0.2056
LSTM + Attention	0.1016	0.1495	0.1935	0.2343	0.2697	0.3003	0.2082
Transformer	0.1045	0.1484	0.1902	**0.2324**	**0.2603**	**0.2910**	**0.2045**

Table 1: MAE for Weekly Drought Forecasts (Using Drought Score)

5.2. Forecast Performance Without Using Drought Score

To enable global application of the model, we trained the same architectures without including the drought score. Table 2 shows the MAE performance in this setting.

Model	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Average
GRU	0.3707	0.3765	0.3843	0.3979	0.4108	0.4307	0.3951
LSTM	0.3536	**0.3584**	**0.3685**	**0.3857**	**0.4004**	**0.4234**	**0.3817**
LSTM + Attention	**0.3510**	0.3655	0.3789	0.3996	0.4139	0.4330	0.3903
Transformer	0.3921	0.4021	0.4183	0.4362	0.4541	0.4764	0.4298

Table 2: MAE for Weekly Drought Forecasts (Without Using Drought Score)

5.3. Discussion

Figure 1 illustrates a clear performance gap between models that incorporate the historical drought score (solid lines) and those that do not (dashed lines). Models using the drought score consistently achieve lower MAE across all forecast weeks, especially in the early stages, highlighting the drought index as a strong predictive feature. Furthermore, MAE steadily increases with the forecast horizon for all models, which aligns with the expected challenge of long-range prediction.

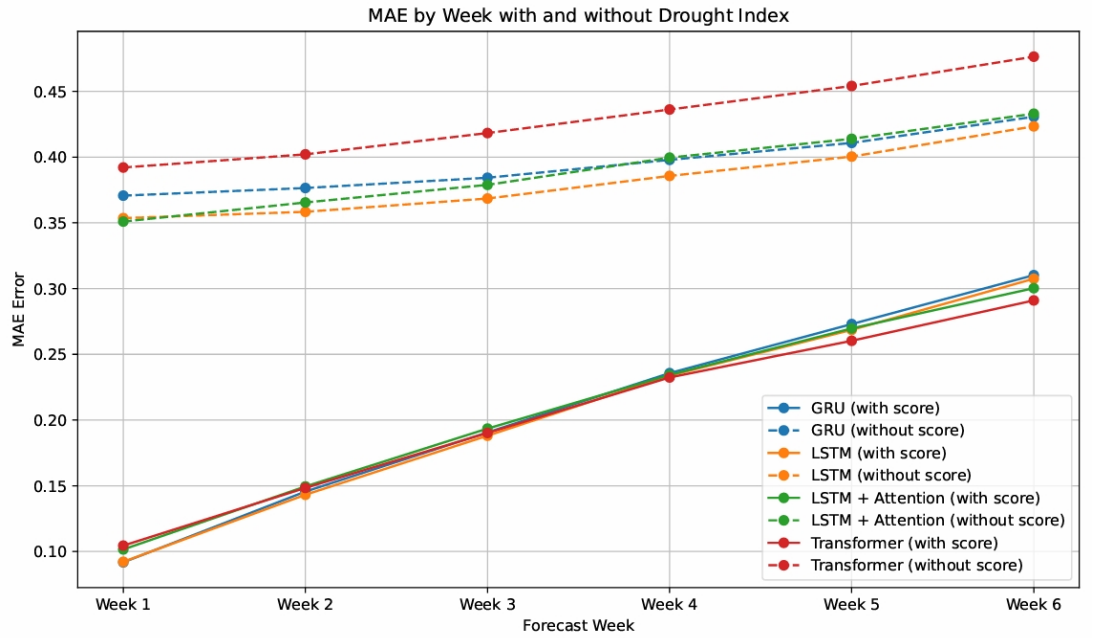
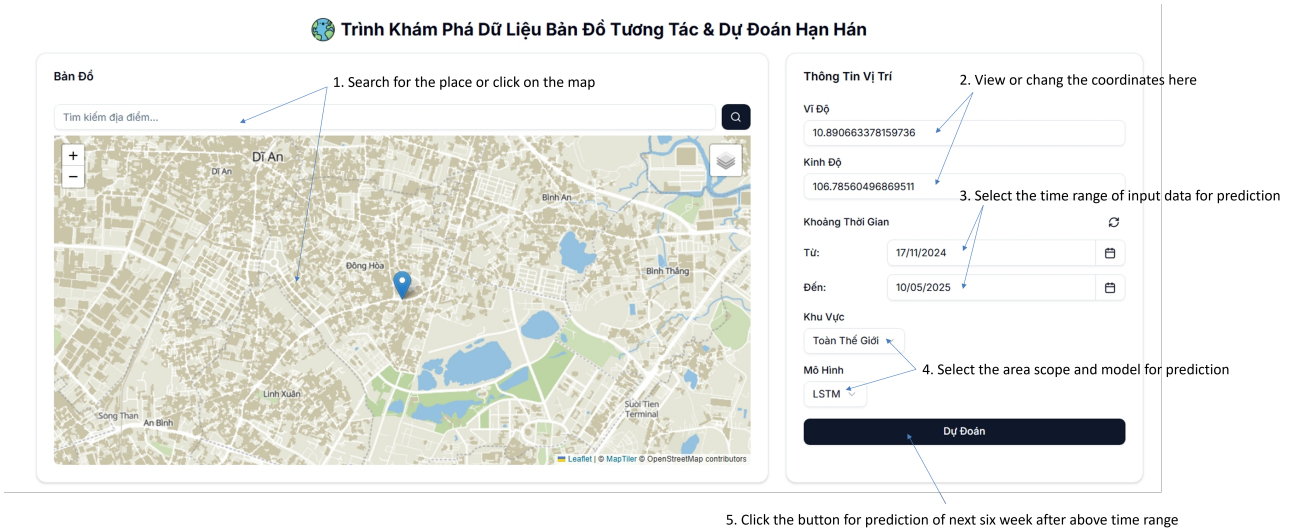


Figure 1: MAE by week with and without drought score. Solid lines denote models using the score; dashed lines indicate models without it.

6. Demo

We developed an interactive web-based system for drought forecasting, allowing users to explore spatiotemporal data and make location-specific predictions [10].



To run the demo locally, clone the repository and follow the `README.md`:

Clone the project repository

```
git clone https://github.com/l1aF-2027/Drought-Prediction
```

References

[1] National Geographic Society. *Drought*. 2023. URL: https://education.nationalgeographic.org/resource/drought/.

[2] cdminix. *US Drought Meteorological Data*. URL: https://www.kaggle.com/datasets/cdminix/us-drought-meteorological-data.

[3] lvanhoang and group 5. *US Drought Meteorological Data*. URL: https://www.kaggle.com/datasets/lvanhoang/drought-dataset-bychristoph.

[4] IIASA and FAO. *Harmonized World Soil Database and Global Terrain Data*. URL: https://webarchive.iiasa.ac.at/Research/LUC/External-World-soil-database/HTML/global-terrain-doc.html.

[5] NASA Langley Research Center. *NASA POWER Project API – Daily Single Point Data*. URL: https://power.larc.nasa.gov/api/pages/.

[6] National Drought Mitigation Center. *U.S. Drought Monitor – Map Archive*. URL: https://droughtmonitor.unl.edu/Maps/MapArchive.aspx.

[7] Dive into Deep Learning. *Chapter: Gated Recurrent Unit (GRU)*. https://d2l.ai/chapter_recurrent-modern/gru.html. 2020.

[8] Dive into Deep Learning. *Chapter: Long Short-Term Memory (LSTM)*. https://d2l.ai/chapter_recurrent-modern/lstm.html. 2020.

[9] GeeksforGeeks. *Machine Learning - Attention Mechanism*. https://www.geeksforgeeks.org/ml-attention-mechanism/. 2021.

[10] huyhoang and group 5. *Drought Prediction*. URL: https://github.com/l1aF-2027/Drought-Prediction.