

AURANET: Recurrent Alternate Update Training for Spatiotemporal Prediction with Incomplete Observations

Farshina Nazrul Shimim; Shilan Felegari; Brett Griesbaum; Bradley M. Whitaker; Paul W. Nugent

Accessibility Disclaimer:

For a more accessible version of this document, please submit an accessibility request form through the Montana State University Library website.

AURANET: Recurrent Alternate Update Training for Spatiotemporal Prediction with Incomplete Observations

Farshina Nazrul Shimim
Electrical and Computer Engineering
Montana State University
Bozeman, Montana, USA

farshin.nazrulshimim@student.montana.edu

Shilan Felegari
College of Agriculture
Montana State University
Bozeman, Montana, USA

shilan.felegari@montana.edu

Brett Griesbaum
College of Agriculture
Montana State University
Bozeman, Montana, USA

brett.griesbaum@student.montana.edu

Bradley M. Whitaker
Electrical and Computer Engineering
Montana State University
Bozeman, Montana USA

bradley.whitaker1@montana.edu

Paul W. Nugent
College of Agriculture
Montana State University
Bozeman, Montana, USA

paul.nugent@montana.edu

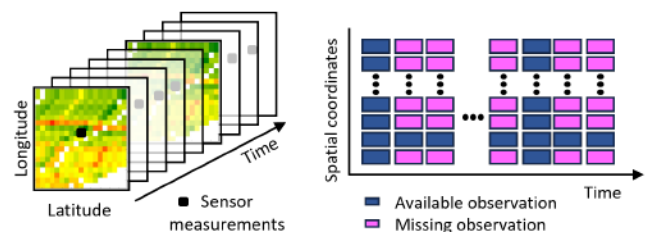
Abstract—Spatiotemporal prediction using data fusion from multiple sources presents significant challenges, especially without proper equipment or resources causing incomplete observations or missing data in the prediction hyperspace. Statistical interpolation methods and conventional machine learning models struggle to capture the true data distribution in highly sparse spatiotemporal settings. To address these limitations, we propose *AURANET* (Alternate Update with Recurrent Approach for Neural nETworks)—a novel training framework that alternately updates the model trainable parameters and the corresponding predictions iteratively until a convergence criterion is met for the available observations. *AURANET* operates in two major steps—(1) with sparse observations as input, it generates a recurrent chain-of-prediction to gap-fill a target spatiotemporal data cube with fixed network parameters (i.e., weights and biases), and (2) at the end of the prediction chain, it updates these network parameters only for the available ground truth observations, i.e., use a subset of the predictions to train the neural network at each training step. In this paper, *AURANET* is implemented to gap-fill and predict daily spatial plant evapotranspiration (ET) data in Montana potato fields, where complete spatial observations are available only on certain days of the growing season, and complete temporal observations are available only for one specific location in the field. The corresponding predictions show a validation MSE of $1.36 \text{ mm}^2 \text{ d}^{-2}$ on the 2023 growing season and a test error of -1.47 mm d^{-1} on the 2024 growing season. The *AURANET* framework extends beyond ET prediction and can be applied to various domains requiring recurrent learning structure with incomplete data.

Index Terms—Spatiotemporal Prediction, Recurrent Neural Networks, Delayed Update Learning, Sparse Training, Sensor-fusion Technologies

I. INTRODUCTION

Spatiotemporal prediction is essential for many real-world applications, such as weather forecasting [1], [2], disease

This material is based upon work supported by the Montana Department of Agriculture under Specialty Crop Block Grant AM22SCBPMT1127.



(a) Aerial imagery (available every 6–10 days) and daily in-field sensor data. (b) Visualization of missing and available observations in the spatiotemporal domain.

Fig. 1: Illustration of sparse spatiotemporal data availability in precision agriculture.

prevention strategies [3], and human behavior prediction [4]–[6]. However, it becomes particularly challenging when observations are sparse due to physical or system limitations—for example, when spatial data are available only at certain time steps, or temporal measurements are recorded at only a few specific geolocations. Fig. 1a illustrates a real-world example in precision agriculture, where aerial imagery of an agricultural field is captured every 6–10 days using satellites [7]. Meanwhile, one or more infield sensors continuously measure weather parameters, capturing daily variability. As a result, many days lack spatial data entirely, and all pixels except those at sensor locations lack temporal data, as shown in Fig. 1b.

Traditional statistical approaches, such as data interpolation [8] and matrix factorization [9], have been used to fill missing values in such scenarios [10]–[13]. However, they rely on strong assumptions about spatial and temporal continuity, making them less effective at capturing complex variability.

Day	d1	d2	d3	d4	d5	d6	d7
Sensor data	T_{d1}	T_{d2}	T_{d3}	T_{d4}	T_{d5}	T_{d6}	T_{d7}
Spatial data (Predicted/Actual)	S_{d1}	S_{d2}	S_{d3}	S_{d4}	S_{d5}	S_{d6}	S_{d7}

Fig. 2: Illustration of the input structure for the multi-input Feedforward Neural Network (FNN) at each time step. The model receives (1) meteorological sensor data from the past w days (in this example, $w = 4$) and (2) spatial predictions (or actual values, if available) for the past w days to predict the next day's spatial data for a given pixel. While training the network, only the spatial predictions for the days with aerial imagery (d1 and d6) is used to update the model parameters.

More importantly, they do not explicitly learn the underlying data distribution from the available observations.

Although machine learning (ML) models such as neural networks (NN) are able to learn the underlying distribution from data, a key challenge in applying conventional NN training methods in sparse settings is the assumption that ground truth labels are available at all time steps [14]. To address this, recurrent neural networks (RNNs), including gated recurrent units (GRUs) and long short-term memory (LSTM) networks, have been explored to model time-series data with missing values [14]–[17]. Some of these approaches introduce imputation strategies [15], probabilistic modeling [16], or multi-directional learning [17]. However, these methods often struggle under extreme sparsity and fail to generalize to highly variable datasets like daily meteorological parameters.

To overcome these limitations, we present **AURANET** (Alternate Update with Recurrent Approach for Neural Networks), a novel training framework that iteratively refines both predictions and network parameters under extremely sparse observations. Unlike conventional methods that require dense training data, AURANET alternates between optimizing model parameters and estimating missing temporal information at each spatial coordinate (as shown in Fig. 1), allowing it to model spatiotemporal variations despite having limited ground truth availability. Inspired by alternating optimization strategies [18], [19], AURANET does not rely on full input sequences but instead trains on sparse observations through an iterative alternate update mechanism, enabling effective learning in sparse spatiotemporal prediction settings.

II. METHODOLOGY

AURANET's training mechanism consists of two key components: (1) recurrent chain prediction—the model iteratively generates daily spatial data, and (2) delayed update mechanism—model parameters are updated only on days with available ground truth. This approach enables it to handle highly sparse spatiotemporal observations by propagating past information while maintaining stability during training.

AURANET utilizes a multi-input feedforward neural network (FNN) that integrates two distinct data streams (Fig. 2): (1) a field-specific input comprising a time series of daily

sensor-based meteorological measurements, and (2) a pixel-specific input consisting of time series predictions for each pixel. These inputs at the pixel level are generated primarily through a recurrent chain prediction mechanism. However, for the days with aerial imagery, the actual data replaces the predicted data in the pixel-specific input to enhance accuracy and reliability. Thus at each time step, the model predicts the next day's spatial data for a given pixel.

The FNN structure, including the different types of layers, follows the design in [7]. The overall training pipeline is illustrated in Fig. 3, with step-by-step details provided below.

- **Step 1 - Model Initialization:** At the beginning of training, the multi-input FNN is initialized with random weights, denoted as $weights_1$, and biases (not shown in Fig 3), denoted as $bias_1$.
- **Step 2 - Recurrent Chain Prediction:** To prepare a valid input window of w consecutive days, the missing days between first two aerial imagery days are interpolated using proportional-offset interpolation or POI, as described in [7]. This is the only instance where interpolation is used and is treated as an edge case. Then, for each pixel, the model performs a day-ahead multivariate time-series prediction spanning the entire season. As shown in Fig. 2, each day's prediction depends on the previous w days' predictions or actual values (if available), creating a recurrent structure. This approach allows the model to estimate missing spatial data using past trends and meteorological inputs.
- **Step 3 - Delayed and Sparse Update:** Once predictions for the entire season are generated, model parameters are updated only on days when ground truth (aerial imagery) is available. No updates occur on intermediate days without ground truth, ensuring that learning happens only at meaningful time steps. After the update, the new model parameters are denoted as $weights_2$ and $bias_2$. Even though the subset of meaningful predictions are days apart, the model still needs to predict for each day until the update day. This is necessary because, due to the recurrent structure, each day's prediction relies on the previous w days' outputs.
- **Step 4 - Evaluation of stopping criteria:** At the end of each model update, the stopping condition is checked. Training stops if the maximum number of iterations, N_{max} , is reached. If not, the process returns to Step 2 for another iteration. Although other criteria, such as convergence thresholds or loss improvements, can also be used in general, at present, AURANET relies solely on the maximum iteration limit for simplicity and stability.
- **Final step - Training Completion:** The model continues iterating through Steps 2–4 until the stopping condition is met. Like most ML algorithms, the time complexity of this process is intractable. However, N_{max} provides an upper bound based on the dataset size and computational constraints. Once training is complete, the model is ready to run real-time.

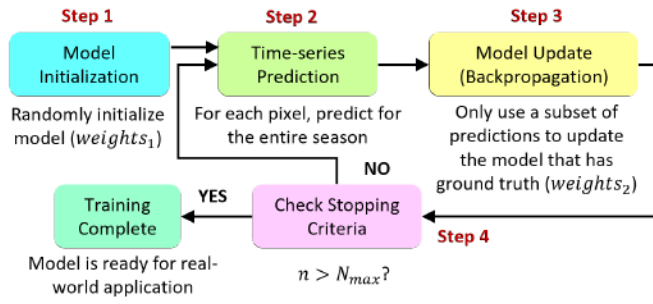


Fig. 3: AURANET's iterative training process—recurrently predicting daily spatial data and updating model parameters only on a subset of days (with ground truth).

The core mathematical framework of AURANET can be formulated as follows.

A. Recurrent Chain Prediction (Forward Pass)

To initialize the pixel-specific network, spatial data between the first two days of aerial imagery (D_s) are interpolated using POI [7] to create a consecutive daily sequence. Only the first w days (in this study, $w = 7$) are fed into the model as input. The network then predicts the spatial value for day $w + 1$ (day eight). This prediction is subsequently fed into the input for the next time step, replacing the earliest day to maintain a fixed window size.

Thus, for any day, $d_k \in D$ within the season, at iteration n , the corresponding prediction function is

$$\hat{S}_{d_k} = f_{\theta_n}(\hat{X}_{d_{k-1}}, \hat{X}_{d_{k-2}}, \dots, \hat{X}_{d_{k-w}}), \quad (1)$$

where, $\hat{S}_{d_k}$ represents the predicted spatial data on day d_k , and $\hat{X}_{d_{k-l}} \mid l = 1, 2, \dots, w$ denotes the set of available (predicted and/or measured) sensor and spatial data for the previous w days. The function f_{θ_n} is the FNN at iteration n with the learnable parameters $\theta_n = \{weights_n, bias_n\}$. Note that $\hat{X}_{d_{k-l}}$ can be represented as

$$\hat{X}_{d_{k-l}} = \{T_{d_{k-l}}, \hat{S}_{d_{k-l}}\}, \quad (2)$$

where $\hat{S}_{d_{k-l}}$ is predicted spatial data (obtained from eqn. 1 for a previous day) and $T_{d_{k-l}}$ is actual temporal sensor data. For days with actual spatial data, $\hat{S}_{d_{k-l}}$ is replaced by $S_{d_{k-l}}$.

To model temporal dependency, each prediction is iteratively fed back into the input, incorporating corresponding daily sensor measurements, to forecast subsequent days. This process continues until the season's end (defined by the user), while intermediate activations for each layer of the FNN are stored for the custom backpropagation step (described in the 'Sparse and Delayed Update' section).

B. Sparse and Delayed Update (Backpropagation)

Once all predictions for the season are generated (delayed update), parameter updates occur selectively based on available reference data from aerial imagery days, D_s (sparse update). Using the mean squared error (MSE) [20], the loss

function for these reference days at iteration n is calculated as follows.

$$\mathcal{L}_n = \frac{1}{|D_s|} \sum_{d_k \in D_s} \frac{1}{M} \sum_{m=1}^M (\hat{s}_{d_k}^m - s_{d_k}^m)^2 \quad (3)$$

where, each $m \in M$ represents a pixel in the spatial data, S , $\hat{s}_{d_k}^m \in \hat{S}_{d_k}$ is the corresponding prediction of the m^{th} pixel, and $s_{d_k}^m \in S_{d_k}$ is the actual (ground truth) value. The loss gradients are then propagated backward across layers, with updates applied using the *Adam* optimizer [21].

This two-step process consisting of a forward pass for the entire season followed by a sparse and delayed update, is iterated using the following equation until convergence:

$$\theta^* = \arg \min_{\theta} \sum_{n=1}^{N_{max}} \mathcal{L}_n. \quad (4)$$

Here, $\theta^* = \{weights^*, bias^*\}$ represents the optimal set of FNN learnable parameters (weights and biases) found within the user-defined hyperparameter search space.

III. EXPERIMENTAL SETUP

A. Dataset Description

The dataset for this experiment includes four potato fields in Montana, USA. Three of these fields (named 2023A, 2023B and 2023C, respectively) were cultivated during the 2023 growing season (from July 9, 2023, to August 28, 2023) and were used for training and validation of AURANET. The fourth field (named 2024A), cultivated during the 2024 growing season (from July 5, 2024, to August 27, 2024), was used for testing. Each site contains one infield sensor (Arable Mark 3 [22]) that measures the daily total of 24 numerical field- and crop-specific parameters, including temperature, solar radiation, relative humidity and Canopy ET (ETc).

In addition, spatial data for the 2023 fields were collected from Sentinel-2 satellite multispectral imagery at a 20 m resolution once every 10 days. Aerial images for the 2024 field were collected on eight different days within the growing period using a DJI Matrice 300 RTK drone equipped with a MicaSense Altum-PT sensor. The multispectral aerial images from the 2023 growing season were converted to spatial ET using an Earth Observation (EO) direct approach [23]. The multispectral drone images from the 2024 growing season were processed using *Pix4Dmapper* software and then converted to spatial ET using the Surface Energy Balance Algorithm for Land (SEBAL) method [24] at 30 m resolution.

B. Model Training, Validation and Testing

The first half of the 2023 growing season (July 9, 2023–August 7, 2023) was used to train the multi-input FNN and the rest of that season (August 8, 2023–August 28, 2023) was used for validation. In this regard, an ensemble feature selection method containing 10 statistical algorithms—spanning filter, wrapper and embedded approaches—was used to identify the most relevant sensor parameters [25]. These include deterministic and stochastic techniques, model-based and model-free

methods, such as Ftest [26], minimal-redundancy-maximal-relevance [27], neighborhood component analysis [28], relief [29], Gaussian process regression [30], linear regression [20], lasso [26], support vector machine [31], generalized additive model [32], and random forest [33], respectively.

The ensemble method ranked all 24 numerical sensor parameters from most to least important for predicting spatial ET. The final set of selected features, along with the FNN architecture and training hyperparameters, was optimized using Bayesian Optimization [34], as described in the following section. For performance evaluation, the model was retrained using the entire 2023 data (both training and validation sets) and the data from 2024 was used for testing using an ‘Out-Of-Sample’ approach [35], where the model was evaluated on unseen data without retraining to assess its generalization performance. The results were compared against a baseline method, Last Observation Carried Forward (LOCF) [36], which propagates the most recent available observation forward in time, assuming temporal persistence.

AURANET was implemented using customized functions in *MATLAB* [37], with the help of various toolboxes, including *Deep Learning Toolbox* [38], *Statistics and Machine Learning Toolbox* [39], and *Global Optimization Toolbox* [40]. The source code and relevant data files are available on the GitHub repository cited as [41].

C. Hyperparameter Optimization

Hyperparameter optimization, the process of selecting the best model parameters, is a crucial step in maximizing network performance. In this study, six hyperparameters were tuned: three defining the FNN architecture, two adjusting training settings, and one determining the number of input features (ranked based on the ensemble feature selection algorithm), respectively. Table I presents the predefined high-dimensional grid search space for these hyperparameters.

TABLE I: Predefined high-dimensional grid (H) for the hyperparameter search space for AURANET

Hyperparameter name	Search space
Hidden Layer nodes (x3)	10, 15, 20, ..., 50, 100, 150, ..., 500
Initial learning rate	0.0001, 0.0005, ..., 0.01
Max epochs	10, 20, 30, 40, 50
Number of features	1, 2, ..., 24

These six parameters were optimized using the Bayesian optimization function, *bayesopt* [34]. A ‘black box’ objective function was defined as the average validation MSE across five pseudo-randomly initialized [42] FNNs, ensuring reproducibility. Validation was performed for all sites over D_s days between August 8, 2023–August 28, 2023.

IV. RESULTS AND DISCUSSION

The optimal result of *bayesopt*, found after 100 evaluations using parameter combinations from Table I, was $h^* = \{25, 200, 150, 0.0005, 20, 8\}$ with a mean validation MSE of $1.36 \text{ mm}^2 \text{ d}^{-2}$ across five pseudo-randomly initialized FNN. These correspond to: a field-specific hidden layer with 25

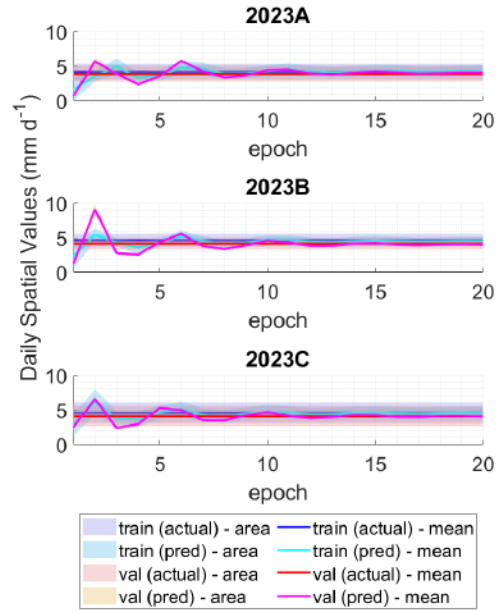


Fig. 4: Progression of AURANET predictions over 20 epochs for training and validation data, showing the mean and standard deviation of actual and predicted values at each epoch.

nodes, a pixel-specific layer with 200 nodes—which concatenates and passes through a shared layer with 150 nodes before predicting the next day’s pixel-specific output of the day $w + 1$ at each instance (here, $w = 7$). The FNN’s initial learning rate is 0.0005, which is adjusted at each step using the *Adam* optimization algorithm. The optimal number of training epochs is 20 and the first eight features (including reference ET, ETc, sunshine duration, and average temperature) were selected.

Fig. 4 shows the progress of the AURANET predictions for both the training and the validation data at each epoch. Early epochs (1–7) exhibited high prediction variability, particularly at site 2023B, but after epoch 8, the predictions stabilized, with the mean validation MSE converging at $1.36 \text{ mm}^2 \text{ d}^{-2}$ by epoch 20.

To evaluate AURANET’s generalization ability, predictions were compared to the baseline LOCF method for the 2023 and 2024 growing seasons. Fig. 5 presents the evaluation for 2023 training data with (a) a scatterplot of actual vs. predicted values for AURANET (blue) and LOCF (magenta) and (b) a boxplot of error distribution across all aerial imagery days. The median values for 2023 data are 4.14 mm d^{-1} (actual), 4.32 mm d^{-1} (LOCF), and 4.15 mm d^{-1} (AURANET), respectively. While AURANET predicted the median ET more accurately than LOCF, it struggled to capture variability, leading to predictions clustered around the median. In contrast, LOCF followed the best-fit line more closely.

For 2024 predictions, the 2023-trained model was tested on five test dates (July 23, 26, and 31, and August 5 and 27) at site 2024A, with the first three dates initializing the prediction chain via aerial imagery interpolation. Fig. 6 presents the

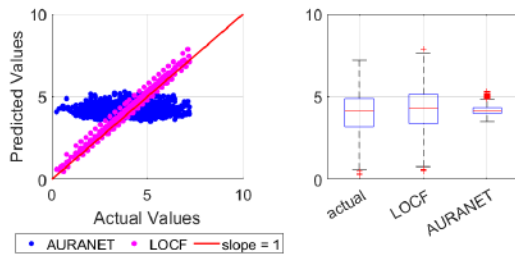


Fig. 5: Comparison of actual ET with AURANET predictions and LOCF baseline for the 2023 sites: (a) Scatterplot of actual vs. predicted ET, (b) Boxplot of prediction errors.

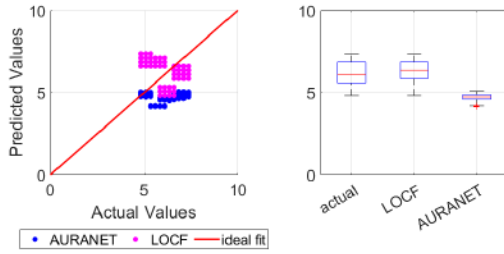


Fig. 6: Comparison of actual ET with AURANET predictions and LOCF baseline for the 2024 test site: (a) Scatterplot of actual vs. predicted ET, (b) Boxplot of prediction errors.

performance evaluation for 2024 using the same scatterplot and boxplot subfigures similar to Fig. 5. Table II provides AURANET's prediction error statistics for each test date in 2024.

AURANET mostly underpredicted ET in 2024 compared to the actual values, except on July 26, 2024 (Table II). The median ET prediction from AURANET (4.8 mm d^{-1}) is lower than both the actual median (6.1 mm d^{-1}) and LOCF (6.35 mm d^{-1}), suggesting a systematic bias in the model's predictions.

Notably, 2023 and 2024 had significantly different data distributions, with 2024 showing a higher range of values compared to the data AURANET was trained on. Even for 2024 predictions, AURANET's outputs tended toward the median values that it was trained on, reducing its ability to capture higher ET values in 2024.

Three possible reasons for this underprediction are (1) the significant differences in weather conditions between 2023 and 2024, (2) the distinct soil type at Site 2024A, and (3) the data source for each year. The 2024 season exhibited greater day-to-day variation compared to 2023, and much

higher spatial values compared to 2023. Furthermore, all 2023 training sites were on Brocko silt loam soil, while site 2024A consisted of a mixture of Glendive sandy loam and Bobkitty clay loam [43], which were not encountered during training. Differences in soil properties can influence plant water availability and consequently ETc, which may contribute to the observed underprediction. Finally, the 2023 data was collected via satellite imaging while the 2024 data was collected via drone flights. These mechanisms have very different resolutions and sensor capabilities, which could contribute to the less prediction accuracy in 2024. In general, AURANET's systematic underprediction for 2024 data indicates a need for more robust training with multi-year, multi-site data.

A major limitation was the Bayesian optimization process, which was restricted to 100 evaluations. This may have led to a suboptimal hyperparameter set, contributing to AURANET's difficulty in capturing data variability. Moreover, while MSE reflects mean accuracy, it does not assess variability. Future evaluations will include residual spread, coefficient of variation, and variance ratio to quantify dispersion, along with R^2 to measure explained variability, ensuring a more comprehensive evaluation beyond mean accuracy.

Although AURANET struggled to capture data variability, its structured learning approach allows it to incorporate daily weather variations, unlike LOCF, which assumes persistence from the last observation. With additional training data spanning multiple years and diverse site conditions, AURANET's predictive performance is expected to improve.

V. CONCLUSION AND FUTURE WORK

In this study, we introduced AURANET, a novel training framework designed for spatiotemporal prediction in highly sparse datasets. Unlike conventional neural network approaches that require dense training data, AURANET iteratively refines both network parameters and missing temporal predictions through a recurrent prediction approach and delayed and sparse update mechanism. This framework enables learning from extremely limited observations, making it particularly suitable for applications where ground truth labels are available only at sparse time steps or locations.

Applied to daily spatial ET prediction in Montana potato fields, AURANET struggled to capture the full variability of ET. Instead, it primarily learned the mean of the training data, resulting in systematic underprediction for the 2024 test set, which had a higher median than the 2023 training data. The discrepancy likely stems from differences in weather patterns, soil types, and sensor sources between years, as well as limitations in the model's training process. The restricted Bayesian optimization search space may have further contributed to AURANET's inability to adapt to the higher ET values observed in 2024.

Future work will focus on expanding the dataset across multiple years, crops, and environmental conditions to improve generalization and minimize biases introduced by site-specific characteristics. Additionally, exploring more advanced optimization strategies, such as Bayesian optimization with a

TABLE II: Mean and standard deviation of prediction errors (predicted - actual) for 2024A site across different test dates

Date	Mean Error (mm d ⁻¹)	Std. Dev. Error (mm d ⁻¹)
23-Jul-2024	-2.1	0.25
26-Jul-2024	0.14	0.19
31-Jul-2024	-1.5	0.25
05-Aug-2024	-2.1	0.19
27-Aug-2024	-1.79	0.24

larger search space or meta-learning approaches, could help refine the model's performance.

Beyond ET prediction, AURANET's framework can be extended to other spatiotemporal prediction tasks, including applications across various datasets and domains. By enabling robust learning in sparse conditions, AURANET has the potential to bridge critical gaps in data-driven decision-making across multiple domains.

REFERENCES

- [1] S. Rasp, P. D. Dueben, S. Scher, J. A. Weyn, S. Mouatadid, and N. Thuerey, "WeatherBench: A Benchmark Data Set for Data-Driven Weather Forecasting," *Journal of Advances in Modeling Earth Systems*, vol. 12, no. 11, p. e2020MS002203, 2020.
- [2] Y. Wang, H. Wu, J. Zhang, Z. Gao, J. Wang, P. S. Yu, and M. Long, "PredRNN: A Recurrent Neural Network for Spatiotemporal Predictive Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 2208–2225, 2023.
- [3] W. Xu, Q. Wang, and R. Chen, "Spatio-temporal prediction of crop disease severity for agricultural emergency management based on recurrent neural networks," *Geoinformatica*, vol. 22, p. 363–381, Apr. 2018.
- [4] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, vol. 3, pp. 32–36, 2004.
- [5] J. Zhang, Y. Zheng, and D. Qi, "Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, Feb. 2017.
- [6] X. Zhao, Y. Sun, T. Zhang, and B. Xu, "Enhanced Spatiotemporal Prediction Using Physical-guided And Frequency-enhanced Recurrent Neural Networks," *arXiv preprint*, vol. arXiv:2405.14504, 2024.
- [7] F. N. Shimim, E. M. Glenn, S. Felegari, B. Griesbaum, J. Fike, B. M. Whitaker, and P. W. Nugent, "Integrating Satellite Imagery and Infield Sensors for Daily Spatial Plant Evapotranspiration Prediction: A Machine Learning-Driven Approach," *2024 Intermountain Engineering, Technology and Computing, IETC 2024*, pp. 162–167, 2024.
- [8] E. Meijering, "A chronology of interpolation: From ancient astronomy to modern signal and image processing," *Proceedings of the IEEE*, vol. 90, pp. 319–342, 2002.
- [9] C. Eckart and G. M. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, pp. 211–218, 1936.
- [10] C. Shi, N. Wang, Q. Zhang, Z. Liu, and X. Zhu, "A Comprehensive Flexible Spatiotemporal Data Fusion Method (CFSDAF) for Generating High Spatiotemporal Resolution Land Surface Temperature in Urban Area," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 9885–9899, 2022.
- [11] W. Jing, T. Lou, Z. Wang, W. Zou, Z. Xu, L. Mohaisen, C. Li, and J. Wang, "A Rigorously-Incremental Spatiotemporal Data Fusion Method for Fusing Remote Sensing Images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 6723–6738, 2023.
- [12] A. Tait and R. Woods, "Spatial interpolation of daily potential evapotranspiration for New Zealand using a spline model," *Journal of Hydrometeorology*, vol. 8, pp. 430–438, June 2007.
- [13] M. C. Anderson, W. P. Kustas, J. G. Alfieri, F. Gao, C. Hain, J. H. Prueger, S. Evett, P. Colaizzi, T. Howell, and J. L. Chávez, "Mapping daily evapotranspiration at Landsat spatial scales during the BEAREX'08 field campaign," *Advances in Water Resources*, vol. 50, pp. 162–177, Dec. 2012.
- [14] Q. Wang, A. Farahat, C. Gupta, and S. Zheng, "Deep time series models for scarce data," *Neurocomputing*, vol. 456, pp. 504–518, 2021.
- [15] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent Neural Networks for Multivariate Time Series with Missing Values," *Scientific Reports*, vol. 8, no. 1, p. 6085, 2018.
- [16] K. Greff, S. van Steenkiste, and J. Schmidhuber, "Neural expectation maximization," *Proceedings of the 31st International Conference on Neural Information Processing Systems*, p. 6694–6704, 2017.
- [17] J. Yoon, W. R. Zame, and M. van der Schaar, "Estimating Missing Data in Temporal Data Streams Using Multi-Directional Recurrent Neural Networks," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 5, pp. 1477–1490, 2019.
- [18] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data Via the EM Algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, pp. 1–22, Dec. 2018.
- [19] H. O. Hartley, "Maximum Likelihood Estimation from Incomplete Data," *Biometrics*, vol. 14, no. 2, pp. 174–194, 1958.
- [20] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, vol. 103. Springer New York, 2013.
- [21] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint*, vol. arXiv:1412.6980, 2014.
- [22] Arable, "Arable Mark 3 Specifications & Measurements," 2023.
- [23] S. Vanino *et al.*, "Capability of Sentinel-2 data for estimating maximum evapotranspiration and irrigation requirements for tomato crop in Central Italy," *Remote Sensing of Environment*, vol. 215, pp. 452–470, Sept. 2018.
- [24] W. Bastiaanssen, M. Menenti, R. Feddes, and A. Holtslag, "A remote sensing surface energy balance algorithm for land (SEBAL). 1. Formulation," *Journal of Hydrology*, vol. 212–213, pp. 198–212, 1998.
- [25] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, Mar. 2003.
- [26] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, 2011.
- [27] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1226–1238, Aug. 2005.
- [28] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Proceedings of the 17th International Conference on Neural Information Processing Systems*, pp. 513–520, MIT Press, 2004.
- [29] M. Robnik-Sikonja and I. Kononenko, "Theoretical and Empirical Analysis of ReliefF and RReliefF," *Machine Learning*, vol. 53, pp. 23–69, 2003.
- [30] D. J. C. MacKay, *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 2002.
- [31] C. Cortes, V. Vapnik, and L. Saitta, "Support-vector networks," *Machine Learning 1995 20:3*, vol. 20, pp. 273–297, Sept. 1995.
- [32] T. Hastie and R. Tibshirani, "Generalized Additive Models," *Wiley StatsRef: Statistics Reference Online*, Nov. 2014.
- [33] T. K. Ho, "Random decision forests," *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, vol. 1, pp. 278–282, 1995.
- [34] C. G. E. Boender and J. Mockus, "Bayesian Approach to Global Optimization: Theory and Applications," *Mathematics of Computation*, vol. 56, p. 878, Apr. 1989.
- [35] C. Bergmeir, R. J. Hyndman, and B. Koo, "A note on the validity of cross-validation for evaluating autoregressive time series prediction," *Computational Statistics & Data Analysis*, vol. 120, pp. 70–83, Apr. 2018.
- [36] X. Liu, "Methods and applications of longitudinal data analysis," *Methods and Applications of Longitudinal Data Analysis*, pp. 1–511, Jan. 2015.
- [37] The MathWorks Inc., "MATLAB Version: 9.14.0.2254940 (R2023a)," 2023.
- [38] The MathWorks Inc., "Deep Learning Toolbox Version 14.6 (R2023a)," 2023.
- [39] The MathWorks Inc., "Statistics and machine learning toolbox," 2023.
- [40] The MathWorks Inc., "Global Optimization Toolbox Version 4.8.1 (R2023a)," 2023.
- [41] F. N. Shimim and B. Whitaker, "BMW-lab-MSU/AURANET_alternate_update: v1.0." <https://doi.org/10.5281/zenodo.15300154>, Apr. 2025. Zenodo.
- [42] H. G. Katzgraber, "Random Numbers in Scientific Computing: An Introduction," *arXiv preprint*, vol. arXiv:1005.4117, May 2010.
- [43] S. S. Staff, "Web Soil Survey, Natural Resources Conservation Service, United States Department of Agriculture," 2024.