Hyperparameter optimization of machine learning models for predicting actual evapotranspiration

Machine Learning with Applications 20 (2025) 100661

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Hyperparameter optimization of machine learning models for predicting actual evapotranspiration

Chalachew Muluken Liyew [a,d,*], Elvira Di Nardo [b], Stefano Ferraris [c], Rosa Meo [a]

[a] Department of Computer Science, University of Turin, Turin, Italy
[b] Department of Mathematics "G. Peano", University of Turin, Turin, Italy
[c] Interuniversity Department of Regional and Urban Studies and Planning, Politecnico of Turin and University of Turin, Turin, Italy
[d] Faculty of Computing, Institute of Technology, Bahir Dar University, Bahir Dar, Ethiopia

ARTICLE INFO

Keywords:
Machine learning
Actual evapotranspiration
Bayesian Optimization
Hyperparameter Optimization

ABSTRACT

Direct measurement of actual evapotranspiration (AET) using eddy covariance and lysimeters is challenging, particularly in large areas, due to high cost, technical complexity, and the need for specialized instrumentation. Consequently, AET data is limited, prompting the use of meteorological and soil features for prediction. This study develops and evaluates machine learning models for AET prediction based on two input combinations. The first group, selected through Pearson correlation, tolerance, and VIF scores to address multicollinearity, includes net CO_2, sensible heat flux, air temperature, relative humidity, and wind speed. The second group, chosen for practical applicability and more accessible, consists of soil surface temperature, air temperature, relative humidity, and wind speed.

Two predictive approaches are proposed: (i) deep learning models (LSTM, GRU, CNN) and (ii) classical machine learning models (SVR, RF). Hyperparameters were optimized using Bayesian optimization and compared with grid search. Bayesian optimization demonstrated higher performance and reduced computation time. Model performance was evaluated using statistical indicators (RMSE, MSE, MAE, R^2). Deep learning methods outperformed classical methods, with LSTM achieving the best results (Bayesian optimization: RMSE=0.0230, MSE=0.0005, MAE=0.0139, R^2=0.8861).

Performance decreased with fewer predictors. LSTM maintained superiority, achieving R^2=0.8861 with five predictors and R^2=0.8467 with four. LSTM also slightly outperformed SVR (R^2 = 0.8456) with fewer predictors. Overall, deep learning methods, especially with Bayesian optimization, have been shown to be more effective than classical machine learning methods for AET prediction. This findings encourage future research using varied input combinations and advanced modeling approaches for AET accurate prediction.

1. Introduction

Actual evapotranspiration (*AET*) refers to the water loss from the surface of the soil and water bodies through evaporation as well as from plants through transpiration (Beven, 1979; Granata & Di Nunno, 2021). It is a key variable that plays a crucial role in regulating the water and energy balance of the soil as well as the atmosphere and vegetation system (Wang et al., 2019; Zhang et al., 2021). Soil drying can limit evapotranspiration which is often approximated by its potential evapotranspiration, which instead is driven only by meteorological forcing. This limitation by soil drying makes the modeling of this variable very challenging because of the nonlinearities involved. It is often the case in relevant applications, e.g. in the case of agricultural systems in low rainfall regions, or in natural ecosystems where it has important feedback on the meteorological variables. In practice, *AET* is a crucial variable to plan and manage the water resources and schedule irrigation for sustainable agricultural development. Hence, developing a prediction model is essential for irrigation management, water resource planning, and environmental monitoring (Beven, 1979; Granata & Di Nunno, 2021; Jiang et al., 2009; Mastrorilli et al., 1998). Granata and Di Nunno (2021) and Jiang et al. (2009) stated that accurate and temporally continuous study of the *AET* is required to optimize irrigation scheduling, secure agricultural and forest habitats, and efficient water management. However, directly measuring *AET* using eddy covariance or lysimeters is challenging, labor intensive, and costly, especially over large areas, which limits the availability of *AET*

* Corresponding author at: Department of Computer Science, University of Turin, Turin, Italy.
E-mail addresses: chalachewmuluken.liyew@unito.it (C.M. Liyew), elvira.dinardo@unito.it (E. Di Nardo), stefano.ferraris@unito.it (S. Ferraris), rosa.meo@unito.it (R. Meo).

https://doi.org/10.1016/j.mlwa.2025.100661
Received 4 January 2025; Received in revised form 21 April 2025; Accepted 23 April 2025
Available online 6 May 2025

data (Granata & Di Nunno, 2021; Talib et al., 2021). Therefore, estimating *AET* using readily available meteorological data is a practical alternative to direct measurements.

AET has to be estimated by statistical or deterministic methods using water and energy balance calculation (Ferreira et al., 2021). For example, to estimate the *AET* rate, the Penman–Monteith equation was often used, beside for computing potential evapotranspiration, e.g., in Beven (1979) and Shekar and Nandagiri (2016). The equation estimates *AET* based on the energy balance between the available energy at the surface and the energy used for *AET* and other processes. The relevant meteorological variables include air temperature (*Ta*), wind speed (*u*), relative humidity (*RH*), and solar radiation (*Rn*), as well as soil surface variables. According to Beven (1979) and Shekar and Nandagiri (2016), estimating the *AET* using the Penman–Monteith equation considers both the aerodynamic resistance and the surface resistance, which represent the resistance to water vapor transfer caused by the atmosphere and the surface, respectively. The daily *AET* is estimated in Mastrorilli et al. (1998) using the Time Domain Reflectometry techniques over a wide range of soil water content in a semi-arid region. Linear regression is used in Gisolo, Bevilacqua et al. (2022) to predict *AET* from meteorological variables such as net radiation, vapor pressure deficit, ground heat flux at the surface, wind speed, and air temperature, grouped into years. Among these variables, the most important driving meteorological variables for *AET* are net radiation and vapor pressure deficit, followed by wind speed. Due to the difficult task of measuring all the required variables needed to estimate *AET*, machine learning models are advocated to predict *AET* from the most available meteorological variables.

Machine learning methods simplified the prediction of *AET* choosing among the most available variables those carefully selected as potential predictors. Zhang et al. (2021) employed the Random Forest (*RF*) technique to predict the *AET* using meteorological variables (temperature, precipitation, and radiation), vegetation, and soil (soil water content and soil temperature). The results indicated that seasonal *AET* performed better than daily and monthly. In Granata (2019) four machine learning techniques were tested (M5P Regression Tree, Bagging, *RF*, Support Vector Regression (*SVR*)) using input variables such as net solar radiation, sensible-heat flux, moisture content, wind speed, relative humidity, and air temperature. Three models were evaluated: the first model utilized all input variables, the second model excluded sensible heat flux and moisture content, and the third model utilized only net solar radiation, relative humidity, and air temperature. The result showed that the first model outperformed the others. Analysis was carried out in Ye et al. (2022) with two machine learning algorithms, namely Dynamic Evolving Neural-Fuzzy Inference System and Multivariate Adaptive Regression Spline, with the optimization algorithms for the estimation of daily *ET* from daily maximum and minimum temperatures in Bangladesh. The result shows that the Dynamic Evolving Neural-Fuzzy Inference System model with the Whale Optimization Algorithm had a good performance. Two models, namely Long Short-Term Memory (LSTM) and nonlinear autoregressive network with exogenous inputs, were tested in *AET* prediction in Granata and Di Nunno (2021), and the LSTM model resulted to be better. Daily measurements of climatic variables are used in Granata et al. (2020) to estimate *AET*. They used random forest (RF), additive regression of decision stump, multilayer perceptron, and K-nearest neighbors models. The result showed that RF and K-nearest neighbors perform better when used as input variables for the measurements of net solar radiation, mean temperature, mean relative humidity, or wind speed. According to Izadifar and Elshorbagy (2010), Genetic Programming, ANNs, and statistical regression models need to be tested using meteorological variables. The authors compared these models for estimating the hourly *AET* and reported that Genetic programming and regression models performed better. The three regression models (RF, cubist regression, and gradient boosting machine) studied by Filgueiras et al. (2020) resulted in the cubist slightly outperforming. LSTM and RF models were tested to estimate and forecast daily ET (Talib et al., 2021), and the authors measured the performance with different input variables context.

In recent years, data-centric deep learning methods have been increasingly employed for forecasting hydrological variables like *AET*. Feng et al. (2024) explored and analyzed the four deep learning methods, including Long Short-Term Memory (LSTM), bi-directional LSTM, Deep Neural Network, and the Deep Belief Network on the estimation of the actual daily ET using the different combination of input variables of *Rn*, *RH*, *Ta*, and *u*, as well as soil water content and Babaeian et al. (2022) also examined the LSTM model using the combination of different input variables such as downward short-wave radiation, upward short-wave radiation, downward long-wave radiation, upward long-wave radiation, relative humidity, wind speed, wind direction, air temperature, and soil water content and showed better performance when all input variables feed into the model.

Generally, Machine learning (ML) techniques have shown promising results in the prediction of *AET* from remote sensing and meteorological data (Mahmoud & Gan, 2019). However, the performance of ML models heavily depends on the selection of appropriate hyperparameters that drive the methods (Wu et al., 2019; Ye et al., 2022), which determine the structure and complexity of the model. Traditional methods for hyperparameter tuning, such as grid search or random search, can be computationally expensive and inefficient, especially for complex models. Grid search was a simple and widely used method for hyperparameter selection; however, it is computationally expensive (Wu et al., 2019). It involves creating a grid of possible hyperparameter values and evaluating the model's performance for each combination of hyperparameters. Liyew et al. (2023) employed a grid search to select the hyperparameters of the ML model to estimate the *AET*. In this study, the Bayesian hyperparameter optimization tuning method is considered to optimize the performance of ML models to estimate the *AET*.

Machine learning algorithms rely on hyperparameters as they play a crucial role in determining how the training algorithms behave, which ultimately impacts the performance of the resulting models (Wu et al., 2019). Different techniques have been created and effectively utilized in specific application areas to optimize these hyperparameters. The authors also explained that Bayesian Optimization (BO) is a promising approach for hyperparameter tuning. It uses a probabilistic model to optimize the objective function efficiently. It iteratively selects the next set of hyperparameter values based on the previous evaluations. Thus, BO can quickly converges to the optimal hyperparameter values. BO has been successfully applied in various fields, including machine learning, robotics, and engineering design. The BO is integrated with LSTM (Habtemariam et al., 2023; Munem et al., 2020) for predicting the wind power and electric power load, respectively. According to Habtemariam et al. (2023), as the number of hyperparameters increases, the effectiveness of grid search strategies diminishes, and the computational complexity of the process becomes a concern. The study in the BO-assisted machine learning models for the prediction of *AET* using meteorological variables is limited. In this investigation, Bayesian optimization (BO) techniques are employed, and the performance of machine learning (ML) is evaluated in comparison to ML models optimized using grid search.

This paper assesses the performance of deep learning and classical machine learning integrating the hyperparameter optimization methods in the prediction of *AET* with the following summarized contributions: (1) Comparing the performance of the Deep Learning and classical ML methods for *AET* predictions. (2) Comparing the performance of machine learning models with a hyperparameter optimization of Grid search and BO. (3) Employing BO on the ML models to select the better hyperparameters of the model to increase its performance. (4) Testing the models with various combinations of inputs (readily measured meteorological variables). (5) Finally, detailed analyses are presented based on the experimental results obtained.

Table 1
Description of all observations and the missing entries of a dataset.

Variables	# Observations	# Missed observations	Missed observations (%)
Evapotranspiration (*AET*)	23 424	0	0
Sensible heat flux	23 424	0	0
Net CO2 flux	23 424	0	0
Air temperature	23 424	194	0.828
Air pressure	23 424	2374	10.135
Wind speed	23 424	2374	10.135
Wind direction	23 424	2374	10.135
Soil surface temperature	23 424	1070	4.568
Net solar radiation	23 424	440	1.878
Relative humidity	23 424	198	0.845
Soil water content	23 424	170	0.726

This paper is organized as follows. Section 2 will describe the study area, the dataset, and the used ML models. This section clearly shows how to select the relevant meteorological variables. In the same section, we briefly review the five models employed in the data analysis. In Section 3 we will present the experimental results and the performance evaluation of the proposed BO-assisted ML techniques. Finally, in Section 4 we will conclude the paper and discuss future research directions.

2. Material and methods

2.1. Study area and dataset

The dataset used in this study was collected from the Cogne meteorological station in Italy (1730 m above sea level, 45.615N, 7.3585E). It consists of readings taken every 30 min for 10 independent variables and one dependent variable *AET*, as summarized in Table 1. The data spans a period of four years, covering the growing seasons (June, July, August, and September) from 2014 to 2017, resulting in a total of 23,424 data.

The site is an abandoned pasture with a 26° slope and a 169° aspect. Between the years 1995 and 2019, the area received an average annual precipitation of 672 mm, with an average temperature of 5.3 °C, and vegetation primarily consisting of grass and shrubs (Gisolo et al., 2024). The dataset spans four growing seasons, including two wet and two dry years, with inter-annual *AET* variation exceeding 100 mm. Significant differences were observed in the mean and cumulative *AET* values between wet and dry seasons over the four years (Gisolo, Bevilacqua et al., 2022). Alpine ecosystems, according to Gisolo, Previati et al. (2022), are hotspots for climate and land use change, featuring complex terrains that complicate long-term measurements of water, energy, and matter fluxes. Consequently, data and modeling tools for accurately assessing current ecosystem conditions and predicting future scenarios are limited.

2.2. Applied machine learning models

In this paper, five ML models are reviewed and developed for the prediction of *AET* using the meteorological multivariate time series dataset. Direct measurement of certain variables, such as AET, sensible heat flux, $net\mathrm{CO}_2$, and solar radiation, can be difficult, necessitating alternative methods for their estimation and prediction. This study reviews and develops five machine learning models applied to meteorological multivariate time series data for predicting future values of Actual Evapotranspiration (AET). Moreover, ML models offer a promising approach for quantifying these variables, particularly *AET*. These models have demonstrated strong potential in predicting *AET* using readily available meteorological data. Both deep learning and classical machine learning techniques are employed in this study to predict AET. Specifically, three deep learning models – Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Convolutional Neural Networks (CNN) along with two classical machine learning models – Support Vector Regression (SVR) and Random Forest (RF) are selected and discussed in detail.

Long short-term memory Neural Network (LSTM) has gained significant popularity in recent years for forecasting time series data. The LSTM network is a type of recurrent neural network designed to effectively model temporal dependencies in sequential data, such as time series. The core of the LSTM architecture lies in its ability to manage and retain information over long periods through a specialized memory cell. This is achieved through three key gates that control the flow of information into and out of the memory cell, allowing the model to learn and adapt to the dynamics of the time series data. These gates are the forget gate (f_t), the input gate (i_t), and the output gate (o_t), each of which plays a distinct role in determining how the cell state is updated (Yadav & Thakkar, 2024; Yao et al., 2023).

The forget gate (f_t) controls which information from the previous time step is discarded from the memory cell by applying a Sigmoid function to the weighted sum of the previous output and current input, producing a value between 0 (completely forget) and 1 (completely retain). The input gate (i_t) determines what new information should be added to the cell state, using a Sigmoid function and multiplying it by a candidate value derived from the current input and previous output. The output gate (o_t) decides which information from the memory cell is passed to the next time step, applying a Sigmoid function to the weighted sum of the current input and previous output. Each gate has adjustable parameters—weight matrices (W_i), (W_f), (W_o) and biases (b_i), (b_f), (b_o) which are learned during training to optimize memory management and capture long-term dependencies in sequential data. Formally, the LSTM network is described as follows (Petneházi, 2019):

$$\mathbf{i}_t = sigmoid(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \tag{1}$$

$$\mathbf{f}_t = sigmoid(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \tag{2}$$

$$\mathbf{o}_t = sigmoid(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \tag{3}$$

$$\hat{\mathbf{c}}_t = tanh(\mathbf{W}_c[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_c) \tag{4}$$

$$\mathbf{c}_t = sigmoid(\mathbf{f}_t \times \mathbf{c}_{t-1} + \mathbf{i}_t \times \hat{\mathbf{c}}_t) \tag{5}$$

$$\mathbf{h}_t = \mathbf{o}_t \times tanh(\mathbf{c}_t) \tag{6}$$

Depending on the input feature $\mathbf{x}_t$ of the time series at time t, the input gate $\mathbf{i}_t$ in Eq. (1) and the forget gate $\mathbf{f}_t$ in Eq. (2) decide what information can be added and what information can be removed from the cell state $\mathbf{c}_t$ in Eq. (5) taking into account the previously hidden value $\mathbf{h}_{t-1}$ of the cell through the cell input activation $\hat{\mathbf{c}}_t$ in Eq. (4). These gates allow it to update the cell state at time t, which represents the long-term memory of the cell. The output gate $\mathbf{o}_t$ at time t in Eq. (3) produces the output vector. The hidden state $\mathbf{h}_t$ is then updated in Eq. (6) for the next time stamp. This latter one is the memory focused for its future use. This memory system enables the network to remember for a long time, provided the forget gate does not intervene.

The LSTM model is developed and applied to two different sets of input variable combinations. In the first case, the model is trained and evaluated using the five selected features as described in Section 3.1. In the second case, a different set of four readily available input variables is used to assess the performance of the model. For each group of input variables, the features are organized into time series, which are then divided into windows of size 48, representing one day of observations (one observation every half hour). These windows are provided as input to the first hidden layer of the LSTM network. At each time step, the window advances to the next, ensuring that successive windows overlap by 47 observations. The target variable in the output layer of the LSTM network is the dependent variable *AET*.

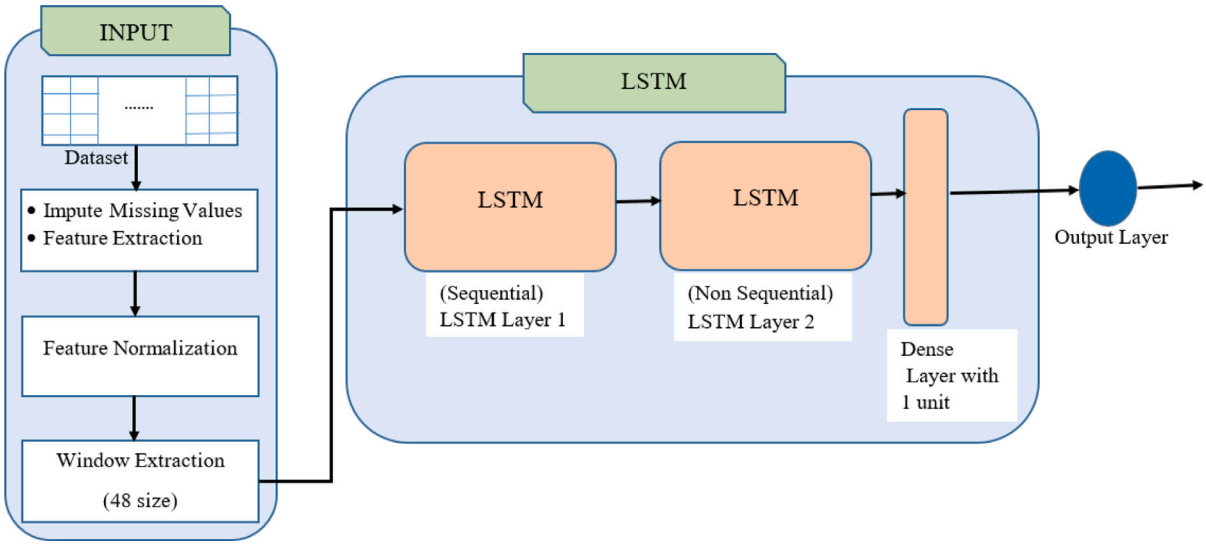

Fig. 1. The architecture of LSTM model.

The LSTM-based model architecture proposed for this study is shown in Fig. 1. The model is trained using the multivariate training set to predict the target variable *AET* with the Adam optimizer (Kingma & Ba, 2015). The hyperparameters of the model, including the number of training epochs, the number of LSTM units, batch size, and the learning rate, are selected using Bayesian optimization. The performance of the model will then be evaluated on the basis of the test set. According to Liu et al. (2021), the network is repeatedly trained by varying the number of hidden layers from 1 to 10. For each layer, the number of neurons is increased from 1 to 128 in increments of 8. The optimal LSTM architecture is then selected based on the Akaike Information Criterion. In contrast to the automatic definition of the hyperparameters, Wang et al. (2022) manually calibrated the hyperparameters of the LSTM model with multiple trials, which can lead to various problems such as time and computational cost, overfitting with the validation set, human error and bias.

Gated Recurrent Unit (GRU) is a type of optimized LSTM-based recurrent neural network (Mateus et al., 2021). Unlike the LSTM, which has separate gates for controlling input and forgetting information, the GRU combines these two gates into a single update gate. This simplification allows the GRU to have fewer parameters than the LSTM, making it more computationally efficient. The mathematical functions used in the GRU network are explained in detail by (Petneházi, 2019).

$$\mathbf{z}_t = sigmoid(\mathbf{W}_z[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_z) \tag{7}$$

$$\mathbf{r}_t = sigmoid(\mathbf{W}_r[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_r) \tag{8}$$

$$\hat{\mathbf{h}}_t = tanh(\mathbf{r}_t \times [\mathbf{h}_{t-1}, \mathbf{x}_t] \times \mathbf{W} + \mathbf{b}) \tag{9}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \times \hat{\mathbf{h}}_t + \mathbf{z}_t \times \mathbf{h}_{t-1}. \tag{10}$$

In comparison to the previous set of Eqs. (1)–(6), the GRU still involves weight matrices $\mathbf{W}_z$, $\mathbf{W}_r$, $\mathbf{W}$ and bias terms $\mathbf{b}_z$, $\mathbf{b}_r$, and $\mathbf{b}$. The update gate $\mathbf{z}_t$ in Eq. (7) and the reset gate $\mathbf{r}_t$ in Eq. (8) function similarly to the forget and input gates in the LSTM unit. Meanwhile, ($\hat{\mathbf{h}}_t$) in Eq. (9) represents the candidate hidden layer.

For this study, a recurrent neural network is designed consisting of a one-layer GRU, followed by two dense layers with ReLU activation function and single units with linear activation function. The selected five features are assigned as an input combination to the model with an arrangement of 48 window sizes corresponding to one day of observations. Each window is given as input to the GRU units. As with the LSTM model, the window is advanced to the next one at each time stamp, so that the windows given as input at two consecutive times overlap for 47 observations of each time series. The dependent variable to be predicted here is the *AET*. The hyperparameters of the GRU model in this particular case are the number of training epochs, the batch size, the learning rate, the units of the GRU, and the units of the dense layer. These parameters are selected using grid search and the mean squared error loss function has been minimized using the Adam optimizer. The performance of the model is evaluated and reported using the test set of the dataset. Bayesian optimization is the other algorithm used to select the hyperparameters suitable for optimizing the model designed for *AET* prediction. The performance of the model with BO and grid search is compared to identify the best model. The better model is retested using the most readily available input combinations of variables, and the performance is evaluated to compare it with the previous model's performance.

In this study, a recurrent neural network (RNN) is designed, consisting of a single-layer gated recurrent unit (GRU), followed by two dense layers with ReLU activation functions and a final layer with a single unit using a linear activation function. Two sets of combinations of groups of input variables are used in model training and evaluation. The group of four readily available input combinations and the group of five selected features are used as input variables to the model, structured into windows of size 48 to represent a full day of observations. Each window is provided as input to the GRU units. Similar to the LSTM model, the window is shifted forward by a one-time step to ensure that successive windows overlap by 47 observations for each time series. The target variable predicted by the GRU model is *AET*. The mean squared error (MSE) is minimized using the Adam optimizer. The hyperparameters of the GRU model such as the number of training epochs, batch size, learning rate, number of GRU units, and number of units in the dense layers are selected by Bayesian optimization. This is used to fine-tune the hyperparameters for better optimization of the model designed for *AET* prediction. The performance of the model is evaluated using BO to determine the most effective model. The selected optimal model is then re-evaluated using the most readily available combinations of input variables, and its performance is compared with the previous model to assess improvements.

Convolutional Neural Networks (1D-CNNs) are deep learning models that have been used effectively for time series prediction, including studies focused on *AET* prediction. CNNs are particularly useful in extracting key features from input data through their convolutional layers, which automate the feature extraction process by applying

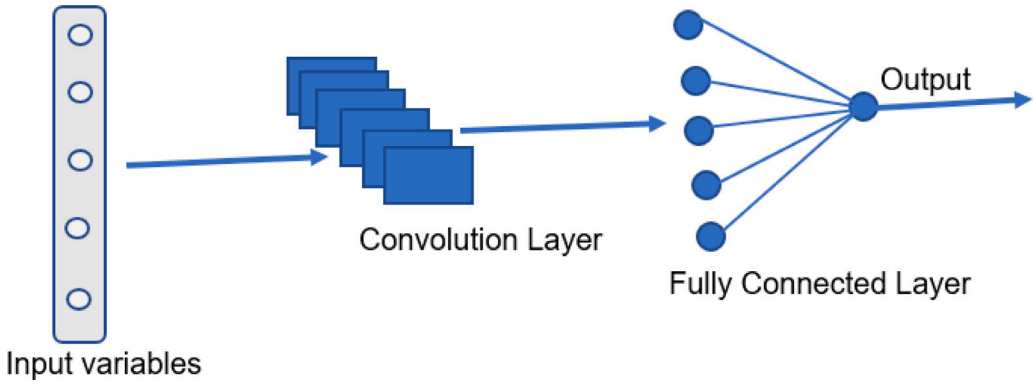

Fig. 2. The architecture of CNN model.

filters to the input, thereby capturing patterns and dependencies in the data (E Lucas et al., 2020). For the analysis performed in this study, the CNN architecture includes a convolutional layer for feature extraction followed by two fully connected layers, as shown in Fig. 2. The performance of the CNN model is determined by several key parameters: the number of filters in the convolutional layer (representing the number of parallel feature detectors), the kernel size (defining the receptive field of the filters), the learning rate, the number of training epochs, and the batch size. Adjusting these parameters allows the CNN to optimize its ability to learn and generalize from the *AET* time series data, leading to accurate predictions. The padding is set to "same".[1] The number of neurons in the first fully connected layer and the activation function are determined by the algorithms. In the second fully connected layer, however, it is the output layer that uses a single neuron and a linear activation function. These hyperparameters for window size, batch size, and learning rate were defined using the same approach used to train the GRU.

Support Vector Machines (SVM) are a classical supervised machine learning approach used for both classification and regression tasks. When applied to regression, SVM, specifically known as Support Vector Regression (SVR), is effective for predicting continuous, ordered variables through the use of a kernel function (Barzegar et al., 2021). The kernel function transforms low-dimensional, nonlinear data points into a higher-dimensional space to make patterns more linearly separable (Barzegar et al., 2021). In Dou and Yang (2018b), SVM was tested with three different kernel types – Radial Basis Function (RBF), Polynomial (Poly), and Sigmoid – for *AET* prediction, where the RBF kernel demonstrated superior performance over the Sigmoid and Polynomial kernels. Faramiñán et al. (2021) and Liu et al. (2021) applied support vector regression to predict the AET with a fivefold cross-validation and linear regression by formulating a linear decision function in a high-dimensional space after dimensional transformation, respectively. According to Dou and Yang (2018a), the radial basis function was used based on the trial and error procedure, and the values of the regularization factor and kernel width were determined by the grid search approach. The value of the insensitive error band width was set to 0.01 by default. In this paper, we configure the SVM model with a set of key parameters: the regularization parameter (C), the margin of tolerance (ϵ), the kernel coefficient (γ), and the specific kernel function. These parameters are optimized using BO to enhance the SVM model's predictive capability. The epsilon parameter[2] with the value determined by the algorithms. This model is tested with the same test data assigned to the other models.

Random Forest (RF) is a machine learning algorithm that constructs multiple decision trees, each trained on different subsets of the dataset, to form an ensemble model (Ferreira et al., 2021). For this study, since *AET* values are continuous, we implement RF for regression tasks to predict *AET*. To improve the accuracy and performance of the model, several hyperparameters are optimized using BO. These hyperparameters are: n_estimators, corresponding to the number of individual decision trees in the forest, max_depth giving the maximum depth of each tree, which directly affects how well the model captures complex patterns in the data, min_samples_split specifying the minimum number of samples required in a node to proceed with splitting, min_samples_leaf specifying the minimum number of samples required in each leaf node, which controls the size of the final branches, and max_features controlling the number of features considered at each split, which adds randomness by limiting each tree's knowledge of the feature set. Liu et al. (2021) set the maximum depth hyperparameter of the RF model from 1 to 30 to repeat the training of the RF model in order to reduce the overfitting situations, and its optimal max_depth was selected by comparing the result of the training and validation dataset models. In this study, these hyperparameters are optimized using BO methods to select the best configuration for accurate *AET* prediction. This fine-tuning helps to achieve a balance between model complexity and predictive performance, ensuring that the RF model is both effective and computationally efficient for time series *AET* data.

Therefore, designing the architecture of machine learning models for optimal performance is a significant challenge. Indeed the success of machine learning models depends heavily on the careful selection of hyperparameters and input combinations of variables as they directly influence the behavior of training processes and have a significant impact on the performance of machine learning models. In this study, Bayesian optimization techniques are applied to improve the performance of the selected machine learning models, reviewed and designed above in predicting *AET*. The performance of the models is then compared with their performance in the grid search studied by Liyew et al. (2023) using the same models with the same combinations of input variables.

2.3. Hyperparameter optimization in machine learning methods

The previously reviewed methods learn models from the observed data: the model learning algorithms are guided by some input parameters provided by the analyst, called hyperparameters to distinguish from the model parameters. In this work we employ an optimization method based on Bayesian theorem, called Bayesian optimization and introduced in the subsequent section, to find the optimal combination of the hyperparameter values. For simplicity, in this context, we call these hyperparameters simply parameters. The following specifies what they are for each method.

[1] Using the same padding ensures that the output dimensions remain consistent with the input dimensions throughout the network layers, eliminating the need for an aggregation function that would otherwise reduce the output size at each layer.

[2] Epsilon defines the tolerance margin within which prediction errors incur no penalty.

LSTM learning algorithm needs to know the number of neurons in each network layer, the batch size, the number of iterations (called epochs), the learning rate (to control the speed of the gradient descent), the dropout rate (this defines the probability of setting any given connection (weight) to zero during training).

The GRU learning algorithm needs to know the number of units in the first layers and the ones in the dense layer, the number of epochs, the learning rate, and the batch size.

CNN learning algorithm needs the number of filter units, the kernel size, the learning rate, the number of epochs, and the batch size.

SVM predicting regression, often called SVR, needs the regularization parameter C, ϵ, γ, and the type of function kernel; for instance, it could be Radial Basis function, polynomial function, linear or sigmoid.

RF algorithm needs the number of estimators (trees), the maximum depth of the trees, the minimum number of samples in a node that controls the tree node split, the minimum number of samples in leaf nodes, and a maximum number of features used by each estimator.

2.3.1. Bayesian Optimization (BO)

As said, the optimization problem discussed here aims to detect the optimal parameter combinations of the algorithms that train the prediction models. As we will see below, the BO method works iteratively by analyzing the candidate parameter values starting from previously tested ones. Yang and Shami (2020) ensured that the grid search and the random search took considerable time to assess the best-performing regions within the search space of the parameters. The authors examine the efficiency of BO, which outperforms Grid Search and Random Search.

BO builds a probability model to find the optimal parameters in a principled and efficient manner. According to Injadat et al. (2018), BO is an algorithm that minimizes a scalar objective function $f(x)$ with x a vector representing a combination of parameter values. It is an iterative algorithm widely used for solving parameter optimization problems based on the Bayesian theorem.

The parameter optimization is represented in Eq. (11):

$$x^{+} = \arg\min_{x \in X} f(x) \tag{11}$$

where X is the parameter space and x^{+} represents the parameter configuration used in the subsequent iteration. Thus the process of selecting x^{+} consists in minimizing the objective function $f(x)$. According to Injadat et al. (2018) the minimization process has three main components. The first component is a Gaussian process model used for the objective function $f(x)$. The second component is the Bayesian update process that modifies the Gaussian model after each new evaluation of the objective function. The last one is an acquisition function which is maximized to identify the next evaluation point. The expected improvement (EI) in the objective function is computed by:

$$EI(x) = \mathbb{E}[max(f(x) - f_{min}, 0)] \tag{12}$$

where f_{min} is the minimum value of the objective function $f(x)$. EI discards values of x that would increase Eq. (12). Moreover the overall iterative procedure updates f_{min} when the candidate configuration x improves the objective.

In general, to perform BO, the parameters should be selected and their respective ranges should be defined. According to s Yang and Shami (2020), the probabilistic surrogate model is designed to approximate the objective function, based on the performance of the prediction model (in our case, LSTM, GRU, CNN, SVR, and RF). For example, the Gaussian process is a surrogate model that approximates the objective function $f(x)$ allowing a more efficient exploration of the parameter space X. At each iteration, the BO algorithm uses an acquisition function to determine which parameter values to be evaluated next. To perform the iterative optimization, BO initializes the surrogate model with a single parameter configuration and sets the corresponding objective function value. The optimization process is iterative and performs the following steps:

1, **Surrogate Model Fitting:** fit the surrogate model to approximate the objective function.

2. **Acquisition Function:** propose the next parameter configuration according to Eq. (12).

3. **Optimization:** solve Eq. (11) to obtain the optimized parameter

4. **Surrogate Model update:** update the surrogate model with the new information obtained by querying the objective function $f(x^{+})$ at the selected x^{+}.

5. **Repeat previous steps 1–4** until a maximum number of iterations is reached.

6. **Selection of the optimal configuration:** select the parameters corresponding to the best-performing configuration observed during the previous steps.

3. Experimental results and discussion

3.1. Data preprocessing

In order to prepare the data for downstream analysis, various stages of filtering and dataset preparation were performed to make the dataset ready for the machine learning model. The missing values in the dataset were random. Therefore, a linear regression algorithm was used to impute or predict the missing values (e.g. air temperature and air pressure as shown in Table 1). We implemented multiple iterative regression imputation (Raghunathan et al., 2001), beginning with two features that have no missing values (sensible heat flux and ($Net\mathrm{CO}_2$) but show high correlation (verified through Pearson correlation) with features needing imputation. For each feature with missing values, a separate regression model was created, using as inputs the features already imputed up to that point. Following each imputation step, an additional feature was imputed and added to the set, which then served as input for the next step. Throughout this process, instances with missing values were held out from each regression model's training set, ensuring they were used only in the test set for each step.

Three variables such as air pressure, wind speed and wind direction show higher missing values of 10.137% each. On the other hand, air temperature, which is correlated with *AET*, shows the lowest missing values of 0.828% as shown in Table 1. According to the result of the correlation analysis in Table 2, the four variables air pressure, water content, wind direction, and soil surface temperature are less correlated with the target variable than the threshold value, so they are excluded from further analysis of the input variables. Further analysis is carried out on the remaining variables that are correlated with the target variables to impute the missing entries. Missing entries for air temperature are imputed first, as this variable has the lowest missing value and is highly correlated with sensible heat flux and $Net\mathrm{CO}_2$. Another regression model is then trained to impute the next variable with fewer missing entries, relative humidity, using sensible heat flux, $Net\mathrm{CO}_2$, and air temperature as regressors. The correlation between all features considered for further analysis is shown in Table 3. Based on how the regression models for imputation are applied, as shown in Table 1, and also tacking into account the correlations of the features with the target feature, as shown in Table 2, all variables are imputed and ready for model input combinations.

The dataset should first be normalized, and then divided into training and testing groups for the model. The machine learning models, especially the deep learning groups such as CNN, LSTM, and GRU, use a sigmoid function whose value varies in the range of 0 and 1. Therefore, the dataset is normalized to the interval $[0, 1]$ using the min–max normalization technique. The other advantage of using the normalization technique is to minimize the impact of outliers on the model training and to avoid scale differences between the features. The

Table 2
Correlation between input features and the target variable, *AET*.

Variables	Correlation with *AET*
Evapotranspiration (*AET*)	1.00
Sensible heat flux	0.82
$NetCO_2$	0.84
Air temperature	0.64
Air pressure	0.01
Wind speed	0.51
Wind direction	0.38
Soil surface temperature	0.41
Net solar radiation	0.89
Relative humidity	0.63
Water content	0.03

min–max normalization technique performs a linear transformation on the original data, preserving their relationships. In fact, the i-th observation of a variable X (denoted by x_i in (13)) is shifted back to the minimum observed value x_{min} of X and then normalized to the observed range x_{min}-x_{max}, i.e.

$$x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}}. \tag{13}$$

In order to identify a meaningful subset of variables for training and prediction, feature selection was performed. This process aimed to retain only the most relevant variables to improve model performance. The Pearson correlation method (Ramirez et al., 2020) was used to assess the strength and direction of the relationship between each predictor and the target variable, *AET*. By calculating correlation coefficients, we can prioritize the features most strongly associated with *AET*, ensuring that only influential predictors were included in the analysis (see Table 2).

The feature selection is performed taking into account not only those features that have a higher correlation with the target, but also those that have low redundancy within the selected variables. To determine the strength of the correlation, a threshold value of 0.5 is fixed. Therefore, all input predictors with a correlation value greater or equal to the threshold value are highly correlated and are selected as relevant candidate features for model development (Senawi et al., 2017). Table 2 shows that the input variables most strongly correlated with *AET* are net solar radiation (0.89) and $net\text{CO}_2$ (0.84). In contrast, the input variables with the weakest correlations with *AET* are water content (0.03) and air pressure (0.01). The presence of irrelevant and redundant features can degrade the performance of the derived model. Therefore, in addition to the Pearson correlation, another technique should be used to identify and exclude multicollinear features. The aim is to increase the overall performance and accountability of the model.

Table 3 shows the Pearson correlation coefficients between the selected features, highlighting the presence of multicollinearity among variables such as net solar radiation, $net\text{CO}_2$, and sensible heat flux. To quantify and address this multicollinearity, we calculated the (unadjusted) coefficient of determination R^2 by regressing each independent variable on the remaining variables. This regression approach helps measure how much the variance of a variable is explained by the other predictors. Using the R^2 values, we then applied the tolerance score and variance inflation factor (VIF) (Cristiano et al., 2016) to assess feature redundancy and identify the most relevant predictors. VIF quantifies how much the variance of a regression coefficient is inflated due to collinearity between variables. A VIF greater than 10, or a tolerance score lower than 0.1, indicates significant multicollinearity, which can distort model estimates and should be corrected (Senawi et al., 2017). By addressing multicollinearity, we ensure that the selected features contribute independently and meaningfully to the model's predictions.

In Table 4, the tolerance and VIF values for net solar radiation are 0.091 (below the threshold of 0.1) and 10.959 (greater or equal to 10), indicating that net solar radiation is a redundant variable. Consequently, this feature was excluded from the analysis, and the tolerances and VIF values were recalculated for the remaining variables. The five remaining features are then selected as input for predicting the next time step in the time series *AET*. This problem is thus reformulated as a supervised learning task with a time window of 48-time steps.

The dataset is divided into training, validation, and test sets for the development of the machine learning models. Initially, the models are trained on the training set, then refined on the validation set by adjusting model parameters, and finally evaluated on the test set to assess their performance. Model parameters are optimized using Bayesian optimization techniques, and their predictive performance is compared. For the aim of this study, the training set consists of 14,054 observations, the validation set contains 3514 observations, and the test set includes 5856 observations, totaling 23,424 observations corresponding to the five meteorological variables at each time step of the time series.

In fact, $NetCO_2$ and sensible heat flux are among the selected meteorological input variables that are complex and costly to measure, and the data are not easily accessible, especially over a large area. These two variables are potential candidates for the prediction model due to their high correlation with the target variable, *AET*. On the other hand, soil surface temperature is significantly correlated with *AET* at 0.41. Therefore, the soil surface temperature is included in place of these two expensive variables and form another combination group of input variables, that is soil surface temperature, air temperature, relative humidity, and wind speed. Then the ML models with the same configuration and the five input variable combinations are analyzed, and the performances are measured and compared using these four readily available meteorological variables.

3.2. Model performance measure statistical metrics

For the aim of this paper, the forecasting performance of each model was measured and compared to identify the best-performing model. The statistical metrics used to measure the model performance are:

The Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2} \tag{14}$$

The mean absolute error (MAE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - \hat{x}_i| \tag{15}$$

The mean squared error (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 \tag{16}$$

And the coefficient of determination R^2

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x}_i)^2} \tag{17}$$

where n is the total number of observed samples, x_i is the measured value for the i-th sample, $\hat{x}_i$ is the predicted value for the i-th sample, $\bar{x}_i$ is the mean, TSS is the total sum of squares and RSS is the residual sum of squares. The higher the value of R^2 is, the better the model, while the other measures $RMSE$, MAE, and MSE are interpreted as prediction error measures. In these cases, the lower the better.

The predictive performance of the five models is measured using the statistical analysis described above. However, it does not prove whether the observed differences are statistically significant in predictive performance. Flach (2012) describes the Friedman test as a non-parametric test to detect statistical significance between observed differences of multiple groups applied to multiple blocks (or observations). In this study, the role of the groups is played by the models (or algorithms). We examine the difference in performance by looking at the prediction error $(x_i - \hat{x}_i)$ over the thousand error test instances ($n = 1000$) of the five models ($k = 5$). A non-parametric test is more powerful because

Table 3
Pearson correlations between the input candidate features.

	Net solar radiation	$NetCO_2$	Sensible heat flux	Air temp	Relative humidity	Wind speed
Net solar radiation	1.00					
$NetCO_2$	−0.83	1.00				
Sensible heat flux	0.92	−0.80	1.00			
Air temp	0.61	−0.49	0.48	1.00		
Relative humidity	−0.60	0.51	−0.52	−0.67	1.00	
Wind speed	0.64	−0.51	0.62	0.28	−0.48	1.00

Table 4
Tolerance, and VIF scores of predictors of variables.

Variables	VIF	Tolerance	Re-VIF	Re-Tolerance
Net solar radiation	10.959	0.091	–	–
$NetCO_2$	3.384	0.296	2.911	0.344
Sensible heat flux	7.150	0.140	3.414	0.293
Air temperature	2.366	0.423	1.956	0.511
Relative humidity	2.167	0.461	2.167	0.461
Wind speed	1.927	0.519	1.751	0.571

with multiple models, some of the observations may not be distributed according to a parametric distribution.

The Friedman test applied to the case study works by ranking the models on each test according to their predictive error. The model with the lowest error gets the rank 1, the next gets 2, and so on. If there are ties, their average rank is used (Pereira et al., 2015). Then the Friedman test evaluates the average of ranks on all the test instances for each model. The Friedman test statistic Q is computed using the following formula:

$$Q = \frac{12}{nk(k+1)} \sum_{j=1}^{k} R_{.j}^2 - 3n(k+1) \tag{18}$$

where $R_{.j} = \sum_{i=1}^{n} R_{ij}$ is the sum of the ranks for model j over the n samples and k is the total number of models.

The following hypotheses are then tested:

- Null Hypothesis (H_0): There are no differences between the models
- Alternative Hypothesis (H_1): At least one model differs from the others.

If Q is higher than the critical value obtained from the chi-squared distribution, the test rejects the Null Hypothesis, indicating that significant differences were observed among the groups.

The Friedman test does not allow us to identify which specific group is superior. Hence, a pairwise post-hoc test should be considered. Flach (2012) and Pereira et al. (2015) mentioned the Nemenyi test, a post-hoc test applied to the observed differences between model pairs. this test compares the difference in average ranks between the two models with the Critical Difference (CD), obtained by the following formula:

$$CD = q_{\alpha,k} \sqrt{\frac{k(k+1)}{6n}} \tag{19}$$

where $q_{\alpha,k}$ depends on the fixed significance level α as well as the number k of models, and the number n of samples.

The analysis proceeded in two stages with distinct sets of input variables. Initially, utilizing all features present in the dataset, we employed the Pearson correlation, tolerance, and VIF methods (as detailed in Section 2.1) to identify significant candidate features for *AET* prediction. We discuss the obtained results in Section 3.3. Subsequently, we focused on a subset including easily obtainable weather variables, namely *soil surface temperature, air temperature, relative humidity*, and *wind speed*. Notably, the features *NetCO2* and *Sensible heat flux* were excluded due to their inherent complexity and the logistical challenges associated with widespread measurement. Among the variables considered, *soil surface temperature* demonstrated a Pearson correlation coefficient of 0.41 with *AET*, ensuring its relevance in predicting *AET*.

3.3. First experiments with a selected number of input variables

In this first case, we conducted experiments considering the selected variables that resulted most correlated with the target variable using the Pearson correlation, tolerance, and VIF methods. These variables are: *NetCO2, Sensible heat flux, air temperature, relative humidity, and wind speed*.

The randomly observed missing values of the dataset are imputed using linear regression and then five relevant features are selected using the Pearson correlation method as described in Section 3.1. The three deep learning neural network models (LSTM, GRU, and CNN) and the classical Machine Learning methods (SVR and RF) for time series are trained and evaluated on the same training, validating, and testing datasets with 48 window size. To optimize the performance of the machine learning models, we employed and compared two methods for finding the optimal parameter settings: the BO tuning method, already described in Section 2.3, and the *Grid search*. BO is more efficient while Grid search extensively performs a brute force search on all the parameter value combinations. The obtained machine learning models are compared by their performance using the mentioned statistical metrics (see the above Eqs. (14)–(17)). The results are given in Table 5. With the bold font, we underline the best results.

The analysis of the results indicates that all the tested models exhibit good predictive accuracy and are suitable for forecasting *AET*. From the results presented in Table 5, it is evident that the deep learning models outperform the other models when the Grid search is employed. Specifically, among the deep learning models, the LSTM demonstrates the highest performance (R^2 = 0.8747), surpassing both the CNN (R^2 = 0.8376) and GRU (R^2 = 0.8512) models. The GRU exhibits slightly better performance compared to the CNN. On the other hand, SVR and RF display relatively lower performance in *AET* prediction. Among the five tested models, SVR demonstrates the weakest accuracy (with R^2 = 0.8144) compared to the remaining four models. Considering the root mean square error (RMSE), the deep learning models exhibit the range [0.0242−0.0275], while SVR and RF yield RMSE values 0.0289 and 0.0281, respectively. In the case of BO parameter tuning, the LSTM again outperforms the other deep learning models, achieving a R^2 value of 0.8861, while the GRU and CNN models achieve R^2 values of 0.8750 and 0.8452, respectively. The BO-assisted SVR and RF models demonstrate similar performance measures R^2 of 0.8394 and 0.8542, respectively. Overall, the results of the experiment indicate that the models exhibit relatively high performance when employed using the BO parameter tuning method. The Grid Search method exhibited a decreased performance and a longer computational time. This observation aligns with the findings of Wu et al. (2019), who highlighted the computational expense of Grid Search and the lowered performance compared to BO for hyperparameter tuning. Fig. 3 illustrates a comparative analysis between the actual and predicted values of AET using five different machine learning models. The actual values are displayed alongside the model predictions, allowing visual evaluation of how well each model captures the underlying patterns and trends. The closer the predicted values align with the actual values, the better the model performance.

By calculating the difference between the actual values and the corresponding predicted ones, the errors for each model were determined. The model errors in the test data are shown in Fig. 4. RF and SVR errors

Table 5
Model performance results with parameters tuned by Grid search and BO on the five input variables.

Model	Grid search				Bayesian optimization			
	RMSE	MSE	MAE	R^2	RMSE	MSE	MAE	R^2
LSTM	**0.0242**	**0.0006**	**0.0155**	**0.8747**	**0.0230**	**0.0005**	**0.0139**	**0.8861**
GRU	0.0264	0.0007	0.0161	0.8512	0.0242	0.0006	0.0152	0.8750
CNN	0.0275	0.0008	0.0169	0.8376	0.0268	0.0007	0.0162	0.8452
SVR	0.0289	0.0008	0.0221	0.8144	0.0272	0.0008	0.0171	0.8394
RF	0.0281	0.0008	0.0167	0.8250	0.0259	0.0007	0.0158	0.8542

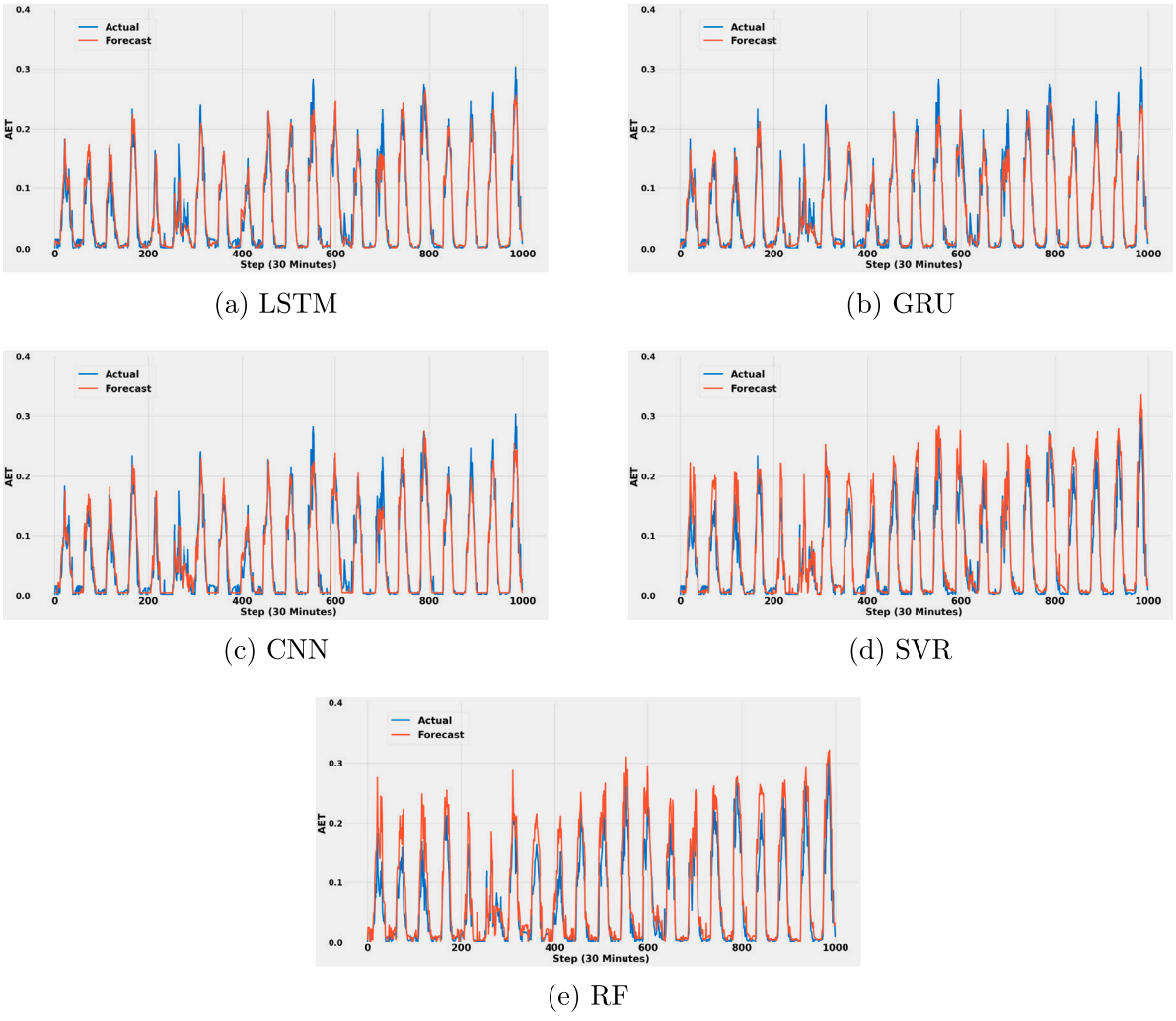

Fig. 3. The time series of the actual values (in blue) Vs. predicted ones (in red) of the different models using the five selected variables. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

tend to be negative and show some outliers that have an impact on the global average result. Deep learning models (LSTM, GRU, and CNN) make positive and negative errors as well and have fewer outliers.

The results of BO-assisted machine learning models show close performance values. It is necessary to apply the Friedman test to the observed differences in prediction errors. The sum of ranks for the models are: LSTM = 3368, GRU = 3303, CNN = 3755, SVR = 2594, and RF = 1980. The Friedman statistic is $Q = 800.998$ and the p-value is approximately zero, so the null hypothesis is rejected: there are significant differences between some of the models. Hence, to analyze the results pairwise, the Nemenyi test is applied. The heatmap in Fig. 5 displays the p-values associated with the results of the test. Each cell represents the statistical significance of the performance difference between the two models. Cells with values below the significance threshold (0.05) indicate a significant difference between the corresponding models. Indeed, there is not a significant difference between LSTM and GRU models, whose p-value is 0.9. Instead, all the other models show a statistically significant difference.

The study (Faramiñán et al., 2021) applied support vector regression to estimate AET using NASA POWER data. The model performed well overall and was more accurate in humid regions than in semi-arid regions. The approach provides a useful alternative for estimating AET in regions with limited ground-based observations, helping to fill data gaps in Argentina. According to the study (Wang et al., 2022), deep learning, specifically the LSTM model, is effective for estimating evapotranspiration in data-poor regions such as the Qinghai-Tibetan Plateau. Key findings include: (1) LSTM performs better with rich input data, but can still works well with only key inputs, unlike process-based models that require more data types; (2) using data from multiple stations improves model performance and demonstrates LSTM's ability

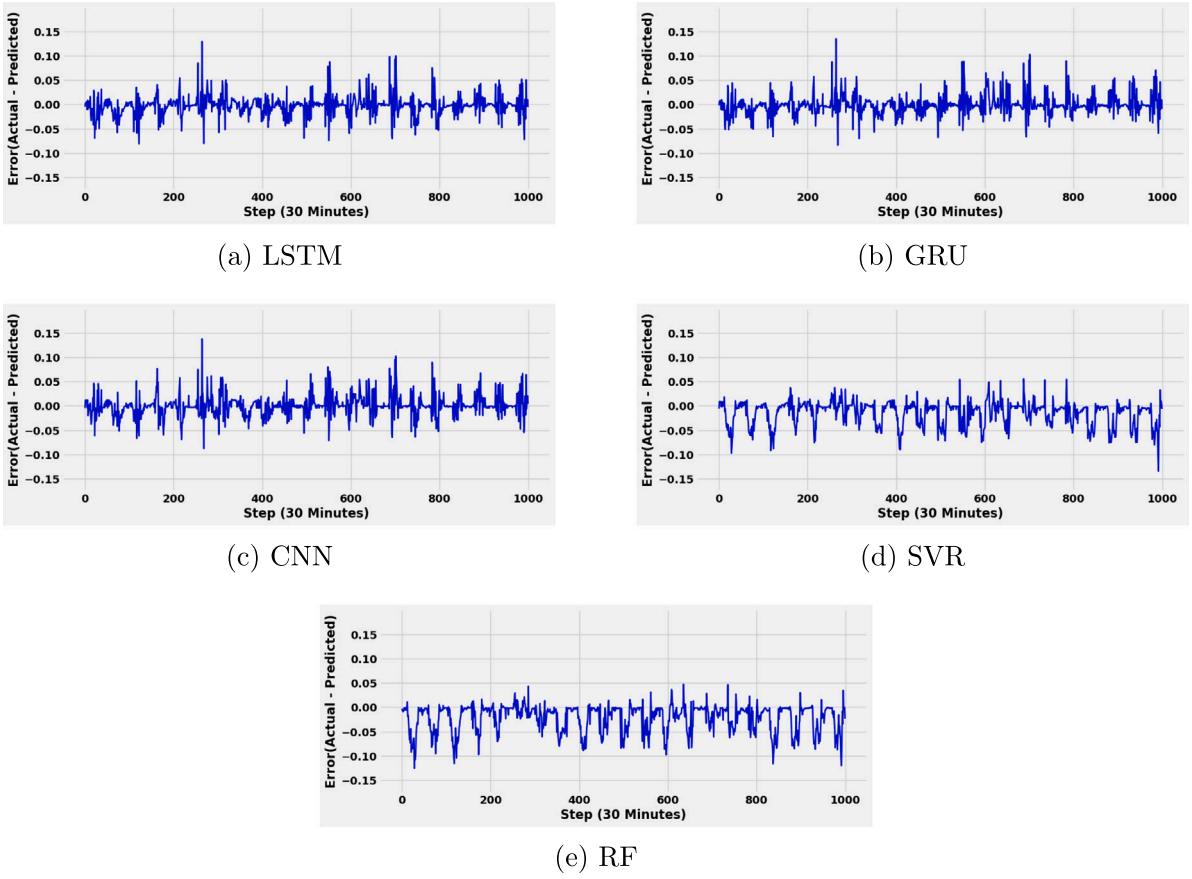

Fig. 4. The error measured between the time series of actual and predicted values of the different models using the five selected variables.

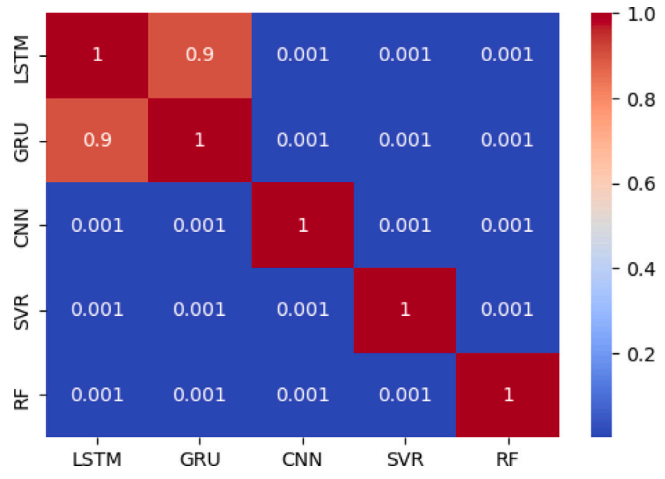

Fig. 5. *p*-values for the Nemenyi post-hoc test results for pairwise comparisons of model performance.

to generalize across regions; and (3) considering the input time series over the length of the sequence improves accuracy, providing a way to enhance training in data-poor areas. The results were presented with specific input combinations, so the analysis suggests repeating the study with more input data. Talib et al. (2021) employed and evaluated the potential of RF and LSTM models to estimate and forecast a daily AET for corn, soybeans, and potatoes in diverse agricultural farms during 2003–2019 in the Midwest USA for the growing season (April–October). The models are evaluated with three set of predictors showing different performances with each predictors. However, the paper stated that the LSTM is more sensitive to uncertainty in ensemble forecast meteorology than RF.

3.4. Second experiment with a smaller number of variables

Once again, we conducted an experiment using the same model architecture, this time employing readily available weather variables.: *soil surface temperature, air temperature, relative humidity*, and *wind speed*. With respect to the first experiment, we eliminated $NetCO_2$ and *Sensible heat flux* because they are difficult and expensive to collect and measure. However, we wanted to perform experiments with a limited set of input variables. In fact, deep neural networks need a sufficient number of observations in order to perform satisfactorily. Consequently, we added to the set of input variables also *soil surface temperature* being considerably correlated with *AET* (Pearson correlation value is 0.41). In this second round of experiments, the machine learning models are employed only with the BO hyperparameter selection method. This decision was motivated by the fact that this variable exhibited significantly notable results for the prediction of *AET* in the previous experiment. In this second round of experiments, the LSTM model results confirm the observations from the first set of experiments. The outcomes of this experiment are detailed in the accompanying Table 6.

The machine learning model aided by BO and utilizing four readily obtainable meteorological input variables exhibited notable performance in *AET* prediction. Specifically, the LSTM model demonstrated the highest performance among the five machine learning models assessed, achieving an R^2 value of 0.8467, closely followed by SVR with an R^2 value of 0.8456. Fig. 6 illustrates a comparative analysis between the actual and predicted values of AET using five different machine learning models with the four easily measurable meteorological variables. The differences between the actual and predicted values highlight each model's strengths and limitations in forecasting AET.

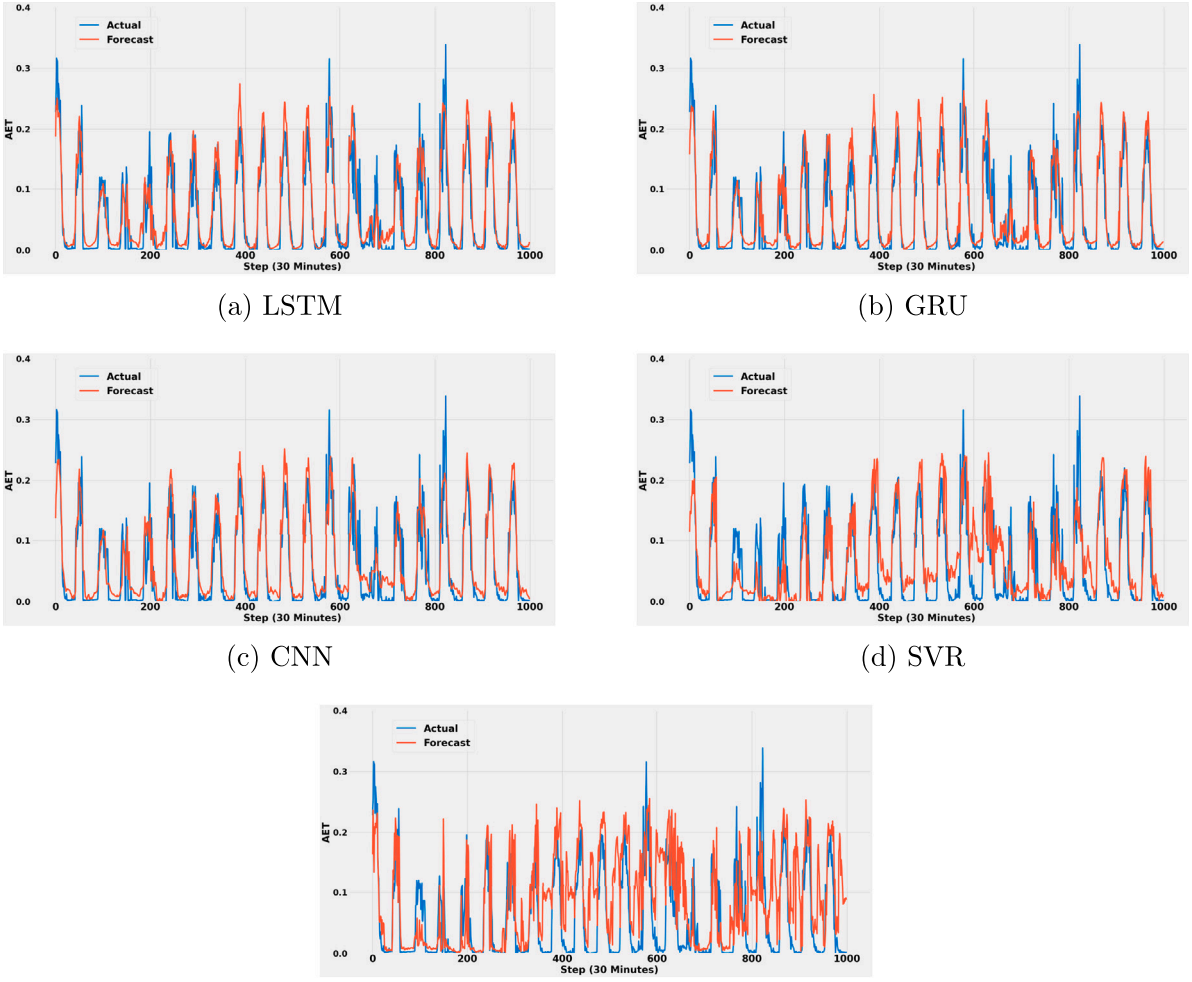

Fig. 6. The time series of the actual values (in blue) Vs. predicted ones (in red) of the different models using four input variables. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 6
Performance results of BO assisted model applied on four input variables.

Model	Statistical measures			
	RMSE	MSE	MAE	R^2
LSTM	**0.0261**	**0.0007**	**0.0164**	**0.8467**
GRU	0.0299	0.0009	0.0189	0.8008
CNN	0.0320	0.0010	0.0215	0.7827
SVR	0.0264	0.0007	0.0171	0.8456
RF	0.0277	0.0008	0.0178	0.8304

In Fig. 6, the actual values of the test data are plotted against the predicted values. The error of the model is obtained by subtracting the predicted values from the actual values of the time series and is plotted in Fig. 7.

Some of these results agree with the literature. Indeed, due to the capabilities of deep learning models to handle time series, these models are expected to outperform the other machine learning models (Talib et al., 2021). This happens also in our study: the deep learning models performed slightly better than the traditional machine learning models. The authors (Filgueiras et al., 2020) assessed three deep learning models (LSTM, CNN-LSTM, and ID-CNN) and two machine learning ones (ANN and RF) carrying out similar performances, even though CNN-LSTM exhibited slightly better performance. For the prediction of *AET*, Talib et al. (2021) experimented with LSTM and RF, but they performed differently depending on the number of predictors. Zhang et al. (2021) compared the deep learning models (temporal convolution neural network-TCN, deep neural network, and LSTM) and the machine learning ones (SVM and RF). According to their analysis, the TCN and LSTM models were better than others in estimating *AET* using temperature-based features. We noticed that hyperparameters play a crucial role in machine learning algorithms as they directly influence the behavior of training algorithms and significantly impact the performance of obtained models. As shown in Table 5, the performance of the models improved when BO was used to tune the hyperparameters of their learning algorithms. This result is supported by Yang and Shami (2020), who describes that choosing the optimal hyperparameter configuration for the machine learning algorithms directly influences the performance of the obtained models.

Concluding, two basic experiments are conducted by altering the input variables to predict the *AET*. One uses five input variables such as $NetCO_2$, *Sensible heat flux, air temperature, relative humidity*, and *wind speed* which are selected using the Pearson correlation, tolerance, and VIF methods. With these input variables, Grid search and BO hyperparameter selection methods are employed to compare the performance of machine learning models in the prediction the *AET*. The experimental results exhibited that the LSTM method outperforms the other machine learning models when both Grid search and BO hyperparameter tunning are used. However, the BO method performs

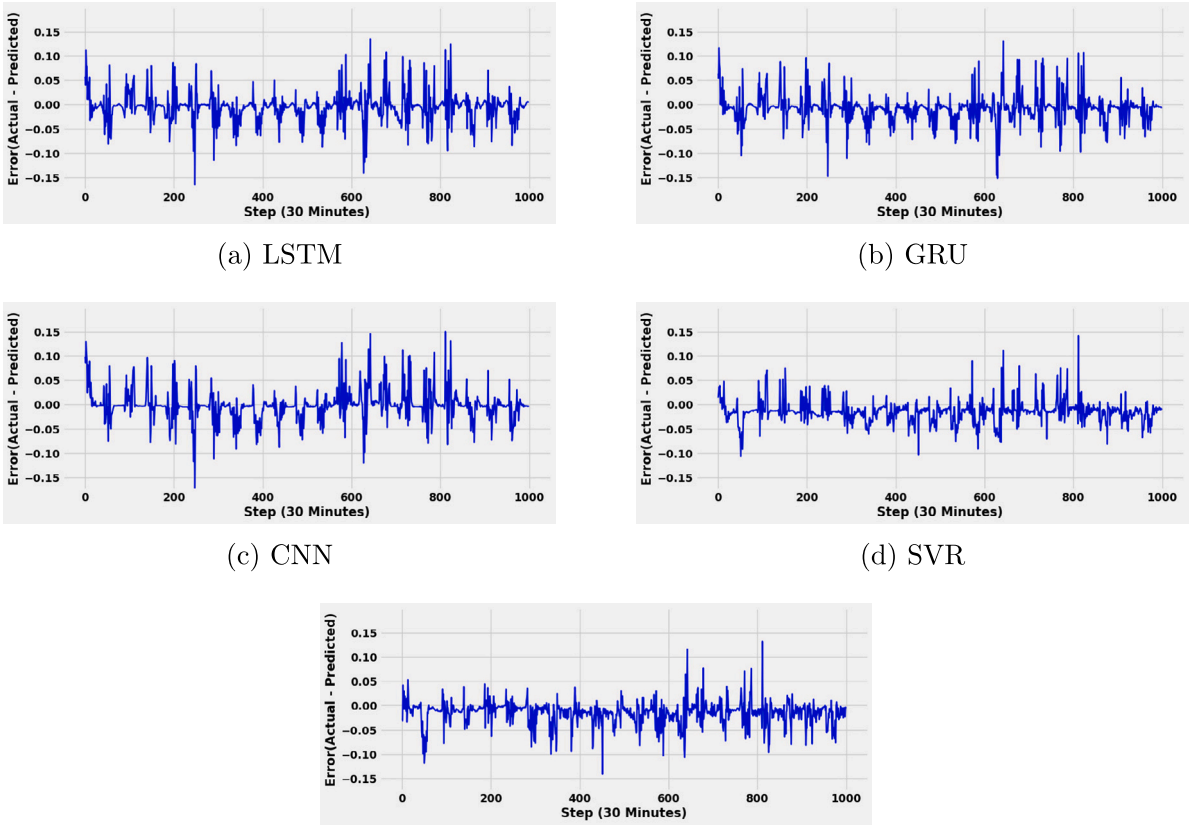

Fig. 7. The error measured between the time series of the actual values and predicted ones of the different models using four selected variables.

better than the Grid search. The second experiment uses more easily obtainable variables. We employed only the BO tuning method because it was more efficient than the Grid search. The obtained results confirmed anyway the outcomes of the first round of experiments. Also in the second round of experiments, LSTM outperforms the other machine learning models followed by SVR. Hence, *AET* prediction using the most easily measurable and obtainable meteorological variables shows the most promising results.

4. Conclusion

This paper addresses the challenges of forecasting *AET* for efficient water resources management. To overcome the difficulty and cost of direct measurements, the study examines the use of machine learning models, specifically three deep learning models (LSTM, GRU, and CNN) and two classical machine learning models (SVM, and RF) for *AET* prediction. The objective is to enhance the performance of machine learning to predict the *AET*. The grid search was applied to the machine learning for hyperparameter selection. In this work,we also considered BO to tune the hyperparameters of the learning algorithms. The study applied feature selection techniques to select relevant variables from a time series of *AET* and a meteorological dataset. The diverse sets of models and hyperparameter adjustments contribute to a comprehensive understanding of *AET* prediction, providing insights into potential improvement in forecasting accuracy.

Two groups of input variables were used to evaluate the performance of machine learning models for AET prediction. The first group is derived from 10 candidate features selected using feature selection techniques. The second group consists of manually selected input variables that are readily available and commonly accessible. Although some variables of the first group have a strong correlation with AET, they are expensive to measure and difficult to obtain at every local meteorological station. Therefore, these costly variables are excluded from the second group, and model performance is evaluated using only the most affordable inputs.

We employed the above-mentioned five machine learning models. This study showed that the deep learning methods outperform the classical machine learning models in forecasting *AET*. The LSTM model slightly outperforms the other models. However, the SVR and LSTM models showed comparable performance when using four input variables. When the number of input variables is decrased, the performance of the machine learning models also decreases, even though it shows promising performance. Among the deep learning approaches, the LSTM model performs better than the other two deep learning methods in all cases. Future work may involve further exploring hyperparameter tuning methods, considering additional variables for enhanced accuracy, and evaluating model performance under different datasets or conditions. Additionally, it may involve to design the *AET* prediction machine learning model using a minimal set of commonly available or easily measurable meteorological variables.

CRediT authorship contribution statement

Chalachew Muluken Liyew: Conceptualization, Design, Data curation, Software, Methodology, Analysis and interpretation of results, Draft manuscript preparation, Writing – review & editing. **Elvira Di Nardo:** Conceptualization, Design, Writing – review & editing. **Stefano Ferraris:** Conceptualization, Design, Data curation, Draft manuscript preparation, Writing – review & editing, Financial sourcing. **Rosa Meo:** Conceptualization, Design, Methodology, Analysis and interpretation of results, Draft manuscript preparation, Writing – review & editing.

Declaration of competing interest

We declare no conflict of interest.

Acknowledgments

The authors would like to acknowledge the funder of this paper. This publication is part of the project NODES which has received funding from the MUR – M4C2 1.5 of PNRR funded by the European Union - NextGenerationEU (Grant agreement no. ECS00000036). It was also partially funded by the PRIN 2022 project Snow Droughts Prediction in the Alps: A Changing Climate, Italy.

Data availability

Data will be made available on request.

References

Babaeian, E., Paheding, S., Siddique, N., Devabhaktuni, V. K., & Tuller, M. (2022). Short-and mid-term forecasts of actual evapotranspiration with deep learning. *Journal of Hydrology*, *612*, Article 128078. http://dx.doi.org/10.1016/j.jhydrol.2022.128078.

Barzegar, R., Aalami, M. T., & Adamowski, J. (2021). Coupling a hybrid CNN-LSTM deep learning model with a boundary corrected maximal overlap discrete wavelet transform for multiscale lake water level forecasting. *Journal of Hydrology*, *598*, Article 126196. http://dx.doi.org/10.1016/j.jhydrol.2021.126196.

Beven, K. (1979). A sensitivity analysis of the Penman–Monteith actual evapotranspiration estimates. *Journal of Hydrology*, *44*(3–4), 169–190. http://dx.doi.org/10.1016/0022-1694(79)90130-6.

Cristiano, P. M., Pereyra, D. A., Bucci, S. J., Madanes, N., Scholz, F. G., & Goldstein, G. (2016). Remote sensing and ground-based measurements of evapotranspiration in an extreme cold patagonian desert. *Hydrological Processes*, *30*(24), 4449–4461. http://dx.doi.org/10.1002/hyp.10934.

Dou, X., & Yang, Y. (2018a). Evapotranspiration estimation using four different machine learning approaches in different terrestrial ecosystems. *Computers and Electronics in Agriculture*, *148*, 95–106. http://dx.doi.org/10.1016/j.compag.2018.03.010.

Dou, X., & Yang, Y. (2018b). Modeling evapotranspiration response to climatic forcings using data-driven techniques in grassland ecosystems. *Advances in Meteorology*, *2018*, http://dx.doi.org/10.1155/2018/1824317.

E Lucas, P. d. O., Alves, M. A., e Silva, P. C. d. L., & Guimaraes, F. G. (2020). Reference evapotranspiration time series forecasting with ensemble of convolutional neural networks. *Computers and Electronics in Agriculture*, *177*, Article 105700. http://dx.doi.org/10.1016/j.compag.2020.105700.

Faraminñán, A. M., Degano, M. F., Carmona, F., & Rodriguez, P. O. (2021). Estimation of actual evapotranspiration using NASA-POWER data and support vector machine. In *2021 XIX workshop on information processing and control* (pp. 1–5). IEEE, http://dx.doi.org/10.1109/RPIC53795.2021.9648425.

Feng, J., Wang, W., Xu, F., & Wang, S. (2024). Evaluating the ability of deep learning on actual daily evapotranspiration estimation over the heterogeneous surfaces. *Agricultural Water Management*, *291*, Article 108627. http://dx.doi.org/10.1016/j.agwat.2023.108627.

Ferreira, A. d. N., de Almeida, A., Koide, S., Minoti, R. T., & Siqueira, M. B. B. d. (2021). Evaluation of evapotranspiration in Brazilian cerrado biome simulated with the SWAT model. *Water*, *13*(15), 2037. http://dx.doi.org/10.3390/w13152037.

Filgueiras, R., Almeida, T. S., Mantovani, E. C., Dias, S. H. B., Fernandes-Filho, E. I., da Cunha, F. F., & Venancio, L. P. (2020). Soil water content and actual evapotranspiration predictions using regression algorithms and remote sensing data. *Agricultural Water Management*, *241*, Article 106346. http://dx.doi.org/10.1016/j.agwat.2020.106346.

Flach, P. (2012). *Machine learning: the art and science of algorithms that make sense of data*. Cambridge University Press.

Gisolo, D., Bevilacqua, I., Gentile, A., van Ramshorst, J., Patono, D. L., Lovisolo, C., Previati, M., Canone, D., & Ferraris, S. (2024). Evapotranspiration of an abandoned grassland in the Italian alps: Modeling the impact of shrub encroachment. *Journal of Hydrology*, *635*, Article 131223. http://dx.doi.org/10.1016/j.jhydrol.2024.131223.

Gisolo, D., Bevilacqua, I., van Ramshorst, J., Knohl, A., Siebicke, L., Previati, M., Canone, D., & Ferraris, S. (2022). Evapotranspiration of an abandoned grassland in the Italian alps: Influence of local topography, intra-and inter-annual variability and environmental drivers. *Atmosphere*, *13*(6), 977. http://dx.doi.org/10.3390/atmos13060977.

Gisolo, D., Previati, M., Bevilacqua, I., Canone, D., Boetti, M., Dematteis, N., Balocco, J., Ferrari, S., Gentile, A., N'sassila, M., et al. (2022). A calibration free radiation driven model for estimating actual evapotranspiration of mountain grasslands (CLIME-MG). *Journal of Hydrology*, *610*, Article 127948. http://dx.doi.org/10.1016/j.jhydrol.2022.127948.

Granata, F. (2019). Evapotranspiration evaluation models based on machine learning algorithms—A comparative study. *Agricultural Water Management*, *217*, 303–315. http://dx.doi.org/10.1016/j.agwat.2019.03.015.

Granata, F., & Di Nunno, F. (2021). Forecasting evapotranspiration in different climates using ensembles of recurrent neural networks. *Agricultural Water Management*, *255*, Article 107040. http://dx.doi.org/10.1016/j.agwat.2021.107040.

Granata, F., Gargano, R., & de Marinis, G. (2020). Artificial intelligence based approaches to evaluate actual evapotranspiration in wetlands. *Science of the Total Environment*, *703*, Article 135653. http://dx.doi.org/10.1016/j.scitotenv.2019.135653.

Habtemariam, E. T., Kekeba, K., Martínez-Ballesteros, M., & Martínez-Álvarez, F. (2023). A Bayesian optimization-based LSTM model for wind power forecasting in the Adama district, Ethiopia. *Energies*, *16*(5), 2317. http://dx.doi.org/10.3390/en16052317.

Injadat, M., Salo, F., Nassif, A. B., Essex, A., & Shami, A. (2018). Bayesian optimization with machine learning algorithms towards anomaly detection. In *2018 IEEE global communications conference* (pp. 1–6). IEEE, http://dx.doi.org/10.1109/GLOCOM.2018.8647714.

Izadifar, Z., & Elshorbagy, A. (2010). Prediction of hourly actual evapotranspiration using neural networks, genetic programming, and statistical models. *Hydrological Processes*, *24*(23), 3413–3425. http://dx.doi.org/10.1002/hyp.7771.

Jiang, L., Islam, S., Guo, W., Jutla, A. S., Senarath, S. U., Ramsay, B. H., & Eltahir, E. (2009). A satellite-based daily actual evapotranspiration estimation algorithm over south florida. *Global and Planetary Change*, *67*(1–2), 62–77. http://dx.doi.org/10.1016/j.gloplacha.2008.12.008.

Kingma, D., & Ba, J. (2015). Adam: A method for stochastic optimization. In *International conference on learning representations*. San Diega, CA, USA.

Liu, Y., Zhang, S., Zhang, J., Tang, L., & Bai, Y. (2021). Assessment and comparison of six machine learning models in estimating evapotranspiration over croplands using remote sensing and meteorological factors. *Remote Sensing*, *13*(19), 3838. http://dx.doi.org/10.3390/rs13193838.

Liyew, C., Meo, R., Di Nardo, E., & Ferraris, S. (2023). Multivariate time series evapotranspiration forecasting using machine learning techniques. In *Proceedings of the 38th ACM/SIGAPP symposium on applied computing* (pp. 377–380). http://dx.doi.org/10.1145/3555776.3577838.

Mahmoud, S. H., & Gan, T. Y. (2019). Irrigation water management in arid regions of middle east: Assessing spatio-temporal variation of actual evapotranspiration through remote sensing techniques and meteorological data. *Agricultural Water Management*, *212*, 35–47. http://dx.doi.org/10.1016/j.agwat.2018.08.040.

Mastrorilli, M., Katerji, N., Rana, G., & Nouna, B. B. (1998). Daily actual evapotranspiration measured with TDR technique in Mediterranean conditions. *Agricultural and Forest Meteorology*, *90*(1–2), 81–89. http://dx.doi.org/10.1016/S0168-1923(97)00089-0.

Mateus, B. C., Mendes, M., Farinha, J. T., Assis, R., & Cardoso, A. M. (2021). Comparing LSTM and GRU models to predict the condition of a pulp paper press. *Energies*, *14*(21), 6958. http://dx.doi.org/10.3390/en14216958.

Munem, M., Bashar, T. R., Roni, M. H., Shahriar, M., Shawkat, T. B., & Rahaman, H. (2020). Electric power load forecasting based on multivariate LSTM neural network using Bayesian optimization. In *2020 IEEE electric power and energy conference* (pp. 1–6). IEEE, http://dx.doi.org/10.1109/EPEC48502.2020.9320123.

Pereira, D. G., Afonso, A., & Medeiros, F. M. (2015). Overview of Friedman's test and post-hoc analysis. *Communications in Statistics. Simulation and Computation*, *44*(10), 2636–2653.

Petneházi, G. (2019). Recurrent neural networks for time series forecasting. http://dx.doi.org/10.48550/arXiv.1901.00069, arXiv preprint arXiv:1901.00069.

Raghunathan, T. E., Lepkowski, J. M., Van Hoewyk, J., Solenberger, P., et al. (2001). A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey Methodology*, *27*(1), 85–96.

Ramirez, I. A. G., Calderon, A., Méndez, A., & Ortega, S. (2020). A novel tool for fast feature selection.

Senawi, A., Wei, H.-L., & Billings, S. A. (2017). A new maximum relevance-minimum multicollinearity (MRmMC) method for feature selection and ranking. *Pattern Recognition*, *67*, 47–61. http://dx.doi.org/10.1016/j.patcog.2017.01.026.

Shekar, N. S., & Nandagiri, L. (2016). Actual evapotranspiration estimation using a Penman–Monteith model. *International Journal of Advances in Agricultural and Environmental Engineering*, *3*, 161–164.

Talib, A., Desai, A. R., Huang, J., Griffis, T. J., Reed, D. E., & Chen, J. (2021). Evaluation of prediction and forecasting models for evapotranspiration of agricultural lands in the midwest US. *Journal of Hydrology*, *600*, Article 126579. http://dx.doi.org/10.1016/j.jhydrol.2021.126579.

Wang, X., Gao, B., & Wang, X.-S. (2022). Investigating the ability of deep learning on actual evapotranspiration estimation in the scarcely observed region. *Journal of Hydrology*, *607*, Article 127506. http://dx.doi.org/10.1016/j.jhydrol.2022.127506.

Wang, D., Zhan, Y., Yu, T., Liu, Y., Jin, X., Ren, X., Chen, X., & Liu, Q. (2019). Improving meteorological input for surface energy balance system utilizing mesoscale weather research and forecasting model for estimating daily actual evapotranspiration. *Water*, *12*(1), 9. http://dx.doi.org/10.3390/w12010009.

Wu, J., Chen, X.-Y., Zhang, H., Xiong, L.-D., Lei, H., & Deng, S.-H. (2019). Hyperparameter optimization for machine learning models based on Bayesian optimization. *Journal of Electronic Science and Technology*, *17*(1), 26–40. http://dx.doi.org/10.11989/JEST.1674-862X.80904120.

Yadav, H., & Thakkar, A. (2024). NOA-LSTM: An efficient LSTM cell architecture for time series forecasting. *Expert Systems with Applications*, *238*, Article 122333.

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, *415*, 295–316. http://dx.doi.org/10.1016/j.neucom.2020.07.061.

Yao, Y., Ye, Z., Bai, W., Kochan, O., & Mokhun, S. (2023). Time series prediction based on LSTM and modified hybrid breeding optimization algorithm. In *2023 13th international conference on advanced computer information technologies* (pp. 584–590). IEEE.

Ye, L., Zahra, M. M. A., Al-Bedyry, N. K., & Yaseen, Z. M. (2022). Daily scale evapotranspiration prediction over the coastal region of southwest Bangladesh: new development of artificial intelligence model. *Stochastic Environmental Research and Risk Assessment*, 1–21. http://dx.doi.org/10.1007/s00477-021-02055-4.

Zhang, C., Luo, G., Hellwich, O., Chen, C., Zhang, W., Xie, M., He, H., Shi, H., & Wang, Y. (2021). A framework for estimating actual evapotranspiration at weather stations without flux observations by combining data from MODIS and flux towers through a machine learning approach. *Journal of Hydrology*, *603*, Article 127047. http://dx.doi.org/10.1016/j.jhydrol.2021.127047.