12-2023

Statistical and Deep Learning Models for Reference Evapotranspiration Time Series Forecasting: A Comparison of Accuracy, Complexity, and Data Efficiency

Arman Ahmadi
University of California

Andre Daccache
University of California

Mojtaba Sadegh
Boise State University

Richard L. Snyder
University of California

Publication Information

Ahmadi, Arman; Daccache, Andre; Sadegh, Mojtaba; and Snyder, Richard L. (2023). "Statistical and Deep Learning Models for Reference Evapotranspiration Time Series Forecasting: A Comparison of Accuracy, Complexity, and Data Efficiency". *Computers and Electronics in Agriculture, 215*, 108424. https://doi.org/10.1016/j.compag.2023.108424

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Statistical and deep learning models for reference evapotranspiration time series forecasting: A comparison of accuracy, complexity, and data efficiency

Arman Ahmadi[a], Andre Daccache[a,*], Mojtaba Sadegh[b], Richard L. Snyder[c]

[a] *Department of Biological and Agricultural Engineering, University of California, Davis, CA 95616, USA*
[b] *Department of Civil Engineering, Boise State University, Boise, ID 83706, USA*
[c] *Department of Land, Air and Water Resources, University of California, Davis, CA 95616, USA*

ARTICLE INFO

Keywords:
Reference Evapotranspiration
Time Series Forecasting
Deep Learning
Prediction Accuracy
Data Efficiency
Computational Cost

ABSTRACT

Reference evapotranspiration (ETo) is an essential variable in agricultural water resources management and irrigation scheduling. An accurate and reliable forecast of ETo facilitates effective decision-making in agriculture. Although numerous studies assessed various methodologies for ETo forecasting, an in-depth multi-dimensional analysis evaluating different aspects of these methodologies is missing. This study systematically evaluates the complexity, computational cost, data efficiency, and accuracy of ten models that have been used or could potentially be used for ETo forecasting. These models range from well-known statistical forecasting models like seasonal autoregressive integrated moving average (SARIMA) to state-of-the-art deep learning (DL) algorithms like temporal fusion transformer (TFT). This study categorizes monthly ETo time series from 107 weather stations across California according to their length to better understand the forecasting models' data efficiency. Moreover, two forecasting strategies (i.e., recursive and multi-input multi-output) are employed for machine learning and DL models, and forecasts are assessed for different multi-step horizons. Our findings show that statistical forecasting models like Holt-Winters' exponential smoothing perform almost as well as complex DL models. Unlike statistical models, DL models generally suffer from low data efficiency and perform well only when enough data is available. Importantly, although the computational costs of most DL models are higher than statistical methods, this is not the case for all. Considering computational cost, data efficiency, and forecasting accuracy, our findings point to the superiority of the neural basis expansion analysis for interpretable time series forecasting (N-BEATS) architecture for univariate ETo time series forecasting. Moreover, our results suggest Holt-Winters and Theta methods outperform SARIMA – the most employed statistical model for ETo forecasting in the literature – in accuracy and efficiency.

1. Introduction

Time series forecasting is crucial in several domains of modern agriculture, like irrigation scheduling, crop modeling, and agricultural water management (Richetti et al., 2023). In addition to a comprehensive understanding of the present status of the system, making decisions usually require a reliable perception of the system's future. In other words, accurate forecasts facilitate sustainable management strategies and decisions (Samaniego et al., 2019). Evapotranspiration (ET) is an essential variable for irrigation management and scheduling. ET is the sum of evaporation from the soil and wet vegetation surfaces and transpiration through plant leaves and plays a prominent role in the global hydrological cycle (Pereira et al., 1999). ET rate is controlled by physical and biological factors and is plant specific. However, reference evapotranspiration (ETo) is a pure meteorological variable that relaxes the plant surface's controlling effects by assuming a hypothetical reference crop surface and standardized conditions (Allen et al., 1998; Diodato and Bellocchi, 2007), which is critical for agricultural water management. There are numerous studies in the literature focused on ETo forecasting. Examples are Gocić et al. (2015) study on soft computing methods for monthly ETo forecasting, Karbasi et al. (2022) weekly ETo forecasting using a hybrid deep learning model, Ferreira and da Cunha (2020) research on using deep learning models to forecast multi-step ahead daily ETo, and Chia et al. (2022) work on long-term

* Corresponding author.
E-mail address: adaccache@ucdavis.edu (A. Daccache).

https://doi.org/10.1016/j.compag.2023.108424
Received 28 July 2023; Received in revised form 2 November 2023; Accepted 6 November 2023
Available online 18 November 2023

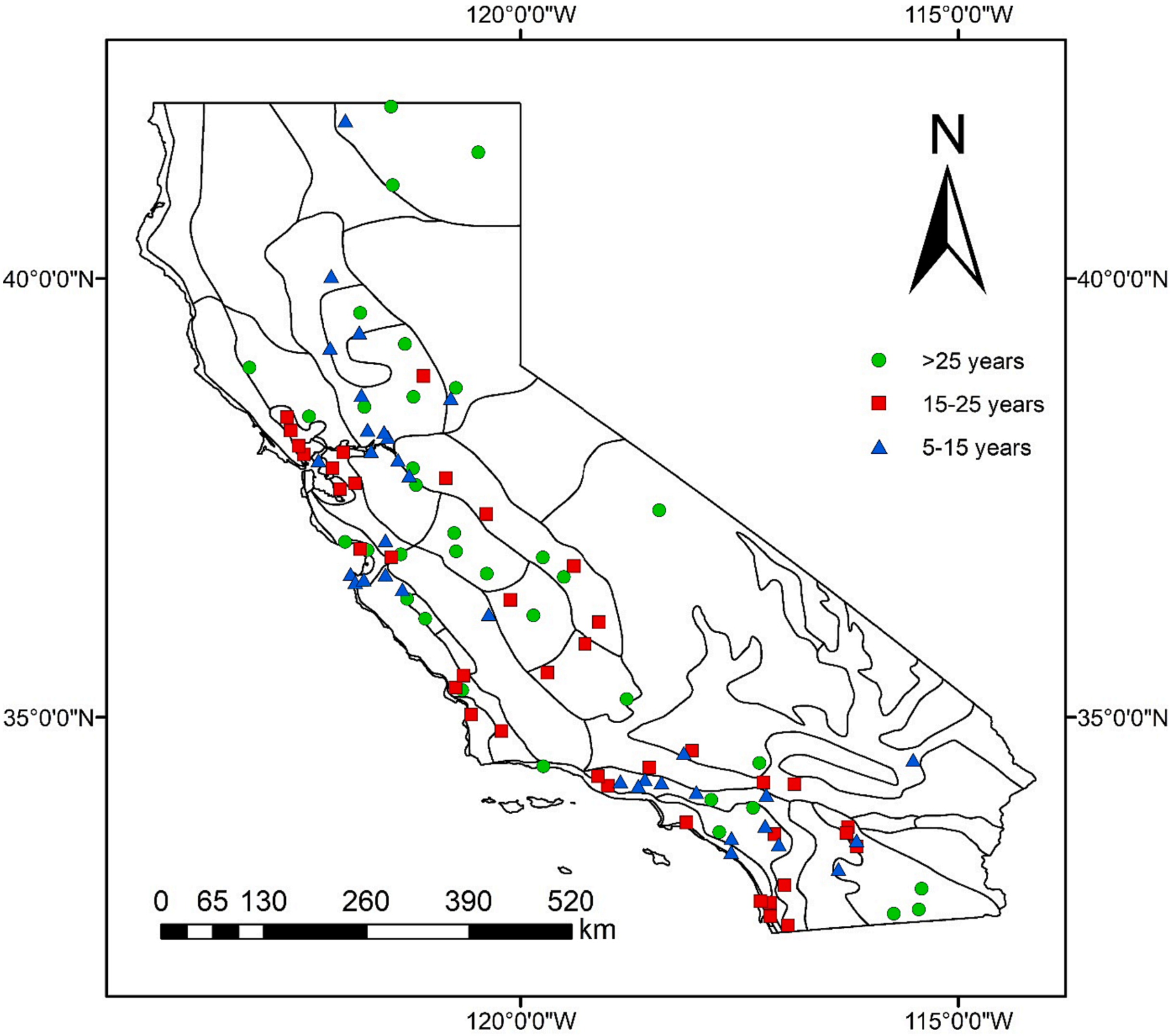

Fig. 1. Location map of CIMIS stations used in this study. Green circles, red squares, and blue triangles show stations with more than 25 years of data, stations with 15 to 25 years of data, and stations with 5 to 15 years of data, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

forecasting of monthly mean ETo using deep neural networks. Many recent research articles attest that ETo forecasting is a prominent case of agricultural time series forecasting (Ferreira and da Cunha, 2020). Moreover, a mature body of research is available in the literature on applying machine learning models for ETo estimation (Chia et al., 2021a; Chia et al., 2021b).

Natural systems are associated with internal randomness and inherent uncertainties that challenge forecasting their future states. From simpler linear models to cutting-edge deep learning (DL) models, numerous forecasting methodologies are available in the literature (Hyndman and Athanasopoulos, 2018), and they address modeling uncertainties differently. Although most of these models have been employed to forecast agricultural and meteorological time series (Ni et al., 2020; Hunt et al., 2022), their performance has rarely been systematically compared. This study aims to fill this research gap by analyzing ten forecasting models' complexity, data efficiency, and accuracy. To this end, we evaluate the models' performance in the case of monthly ETo forecasting in California. Having a significant seasonality and negligible trend in short periods, monthly ETo is a suitable archetype of agrometeorological time series. Moreover, with more than one hundred standardized weather stations, grave water availability, and quality challenges, California is a suitable case study for this research (Lund et al., 2018; Ahmadi et al., 2022).

Forecasting models evaluated in this study are either ubiquitous statistical forecasting methods employed in numerous studies for agrometeorological time series forecasting or cutting-edge DL models with promising performances over benchmark data sets and in data science competitions. These models are divided into three classes: 1) statistical models consisting of autoregressive integrated moving average (ARIMA), seasonal ARIMA (SARIMA), Holt-Winters' exponential smoothing, and Theta method; 2) machine learning (ML) models consisting of the light gradient-boosting machine (LightGBM); and 3) state-of-the-art DL models including neural basis expansion analysis for interpretable time series forecasting (N-BEATS), long short-term memory (LSTM), temporal convolutional network (TCN), Transformer model, and temporal fusion transformer (TFT). The models' performance is evaluated in a multi-dimensional framework for more in-depth investigation. For this purpose, weather stations are categorized based on their historical data availability, two multi-step ahead forecasting strategies are employed for ML and DL models, and forecasts are made for different forecasting horizons. Forecasting strategies used in this study are recursive and multi-input multi-output (MIMO) strategies. Contrary to the recursive strategy that forecasts one time step at each iteration, the MIMO strategy forecasts the entire forecasting horizon at

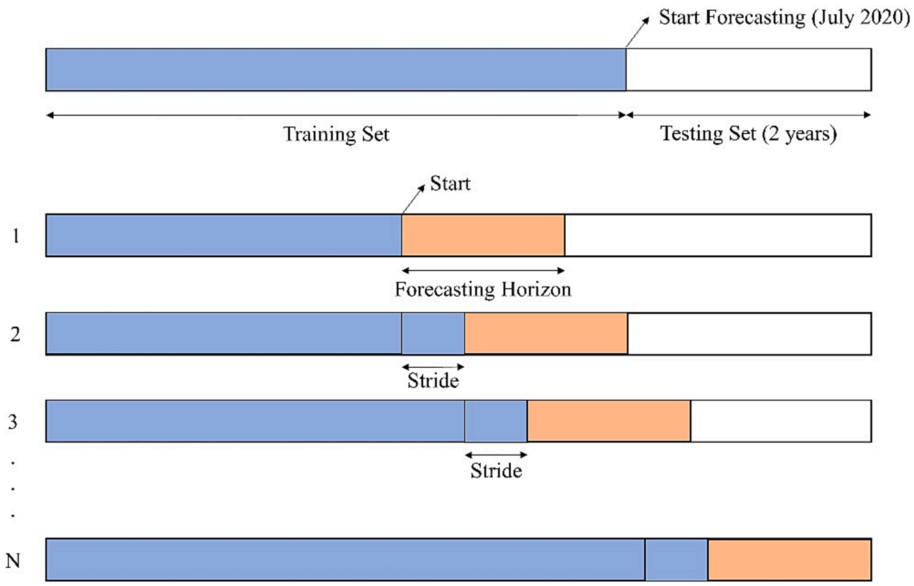

Fig. 2. Data splitting and forecasting time steps.

once.

The multi-dimensional analysis of this study goes beyond the typical comparison of performance measures in the literature and casts light on various aspects of agrometeorological time series forecasting, informing future applications. It is also worth mentioning that although some of the more well-known and older DL models, like LSTM, have been applied in agricultural and hydrological studies (Hunt et al., 2022), most of the more recent model structures, like N-BEATS and TFT, have had limited or no records of application in the hydrological and agricultural research. Implementing the state-of-the-art DL forecasting models, this study goes beyond analyzing the current trend in agricultural forecasting. It aims to introduce the advances of time series forecasting to the field of agrometeorology and to promote the current research trend in the field.

Here, we hypothesize that more complex DL models (i.e., deeper and wider architectures) generally outperform statistical methods (Makridakis et al., 2022). We also hypothesize that DL models encounter challenges in accurately forecasting future time steps for more recently established weather stations (e.g., stations with less than 15 years of data records), as more complex data-driven models tend to require more input data and be less data-efficient (Hassani et al., 2021). Additionally, we hypothesize that more complex DL models are computationally expensive and take longer to train. Furthermore, we hypothesize that ML and DL models trained with the MIMO strategy outperform models trained with the recursive strategy (Taieb et al., 2012). We test these hypotheses consistently across models. In doing so, we keep all the effective parameters (e.g., input data length) similar for all the models. Moreover, we analyze the potential effects of background factors (e.g., geographical characteristics and climatic conditions) to ensure the reliability of our findings.

2. Study area and dataset

In our case study of California, meteorological drivers of ETo (i.e., inputs of the Penman-Monteith equation, which are net radiation, soil heat flux, air temperature, vapor pressure, and wind speed) are measured in standardized weather stations throughout the state. These stations are managed under the *California irrigation management information system (CIMIS)*, which consists of over 145 automated weather stations to assist irrigators and water managers with planning and decision-making. CIMIS uses the Penman-Monteith equation and a modified version of Penman's equation to calculate ETo. Hourly weather data is used to calculate hourly ETo, which is subsequently added up over 24 h (midnight to midnight local time) to estimate daily ETo. Monthly ETo values reported by the CIMIS portal are the aggregation of daily values in metric units (mm). More information about CIMIS data and the Penman-Monteith equation can be found in Ahmadi et al. (2022).

For this study, monthly ETo data from 107 active CIMIS stations are acquired (https://cimis.water.ca.gov/). Stations are chosen according to their maintenance and condition records. Stations with unreliable or inconsistent data (e.g., stations with poor maintenance, non-grass reference surface, or inadequate irrigation) are eliminated from this analysis. All 107 stations used in this study have data available until the end of June 2022 without any missing values for monthly ETo. However, the stations have been established at different times; therefore, their data's start dates vary. To have a better understanding of the effect of historical data availability on the performance of the forecasting models, stations are categorized into three categories: 1) long records: stations with more than 25 years of data, 2) medium-length records: stations with 15 to 25 years of data, and 3) short records: stations with 5 to 15 years of data. There are 34, 38, and 35 stations in the first, second, and third categories, respectively. The oldest stations have data available from 1986. No further data preprocessing has been conducted on the raw ETo data.

Fig. 1 shows the distribution of the stations for each category. The zoning scheme in Fig. 1 refers to homogeneous zones with respect to the reference evapotranspiration. CIMIS divides California into 18 homogeneous zones with similar meteorological and evapotranspiration characteristics. More information about these zones' climatic characteristics and locations can be found in Table S1 and Figure S9 in the supplementary material. Readers are referred to Ahmadi et al. (2022) for more in-depth information about how ETo and its meteorological driving factors compare between these zones.

3. Methodology

3.1. Time series decomposition

An essential step in analyzing time series characteristics is decomposition. This technique splits the time series into three main components: trend, seasonality, and residual error (a.k.a. noise components). In this study, we used the *Statsmodels* library in Python to perform an additive time series decomposition (Equation (1):

$$Y_t = T_t + S_t + r_t \quad (1)$$

Where Y_t is the observed value at time t, and T_t, S_t, and r_t are the trend, seasonality, and residual components, respectively.

3.2. Forecasting strategy and horizon

This research focuses on univariate time series forecasting without using exogenous variables to forecast monthly ETo. Hence, the only input to forecasting models is monthly ETo in previous time steps. To perform one-step and multi-step ahead forecasting, we employed three forecasting horizons: one month ahead, three months ahead, and six months ahead. The last two years of data, July 2020 to June 2022, are used as the test set. For each station, all the forecasting models are trained with the data before July 2020, and after training, their performances are evaluated according to their forecasts for the test set. Having the same time frame as the test set for all the stations and models minimizes the seasonal and climatological biases in measuring forecasting accuracy.

Fig. 2 shows how the models are trained and used to forecast. The model is trained with the original training data to forecast the first horizon. The training data is expanded in the next step, and the stride is added. The model is now retrained with this expanded training set and forecasts the next horizon. This procedure continues until the model forecasts the last time step (i.e., June 2022). For all horizons, the stride is set to one time step (one month). Notably, we conducted retraining only for statistical and machine learning models. Since DL models require a long time to train, these models are trained only on the original training set and are not retrained at each step. However, the DL models use extended data at each step as input.

We employed two well-known strategies for multi-step ahead time series forecasting: recursive strategy and multi-input multi-output (MIMO) strategy. Recursive strategy, also called *iterated* or *multi-stage*, is the oldest and most intuitive forecasting strategy (Taieb et al., 2012). In this strategy, forecasting model *f* is trained to perform a one-step-ahead forecast:

$$y_{t+1} = f(y_t, \cdots, y_{t-k+1}) \quad (2)$$

Equation (2) represents a univariate recursive strategy that forecasts the variable of interest (y) at time step t + 1 using the same variable at k previous time steps. A recursive strategy can also be used for multi-step ahead forecasting. To do so, the model first forecasts the first step. Subsequently, the forecasted value is added to the input variables to forecast the next step, using the same one-step ahead model *f*. This procedure is continued until the whole horizon is forecasted.

Contrary to the recursive strategy, in the MIMO strategy, the forecasting function *F* is a multiple-output function that forecasts the entire horizon simultaneously:

$$[y_{t+H}, \cdots, y_{t+1}] = F(y_t, \cdots, y_{t-k+1}) \quad (3)$$

As shown in equation (3), in the MIMO strategy, the whole forecasting horizon, which consists of *H* time steps, is forecasted by one iteration of the *F* function. It is worth noting that there is no difference between recursive and MIMO strategies for one-step ahead forecasting. It should be noted that the MIMO strategy can be implemented for machine learning and DL models only. In other words, the statistical forecasting models of this study are all restricted to recursive strategy.

3.3. Time series forecasting models

We use *Darts*, a Python library for time series manipulation and forecasting (Herzen et al., 2022). Darts contains a variety of forecasting models, from statistical such as ARIMA to cutting-edge deep neural networks. Readers are referred to Herzen et al. (2022) for more information about this library.

3.4. Statistical forecasting models

3.4.1. (Seasonal) autoregressive integrated moving average (ARIMA and SARIMA)

The ARIMA model is one of the most popular linear models in time series forecasting (Contreras et al., 2003). ARIMA combines autoregressive (AR), differencing (I), and moving average (MA) features (Hyndman and Athanasopoulos, 2018). Differencing refers to computing the difference between consecutive observations to remove non-stationarity from time series. The autoregressive component is a linear model that forecasts the variable of interest using a linear combination of past values of the same variable. In other words, the AR model is a linear univariate forecasting model. The MA model, on the other hand, uses past forecast errors instead of past values of the variable of interest in a regression-like model. An ARIMA model has three hyperparameters, which are non-negative integers and need to be determined by the user: *p,* which is the order (number of time lags) of the AR model, *d* which is the order of differencing (i.e., the number of times the data have had past values subtracted), and *q* which is the order of the MA model (i.e., the size of the moving average window). ARIMA model is a non-seasonal forecasting model.

The seasonal ARIMA (SARIMA) model includes additional seasonal terms in the original ARIMA model. SARIMA has four additional hyperparameters, *P*, *D*, and Q (order of the seasonal component for the AR, difference, and MA models, respectively). The fourth hyperparameter of the SARIMA model is *m,* which is the periodicity or the number of time steps in a whole seasonal period. For monthly data, $m = 12$. More detailed information about the ARIMA and SARIMA models can be found in Hyndman and Athanasopoulos (2018).

This study used the *pmdarima* statistical library in Python to optimize ARIMA and SARIMA hyperparameters. We optimized hyperparameters for each station and used those station-specific parameters to train ARIMA and SARIMA models. From the *pmdarima* library, we used the *AutoARIMA* model, which identifies the optimal set of parameters for ARIMA and SARIMA models, settling on a single-fitted model. In calibration, we set the maximum p, q, P, and Q values to 5. As we work with monthly data, *m* was set to 12. The information criterion used to select the best model was the Akaike information criterion (AIC). We used alpha = 0.05 as the test level for statistical significance, the *Kwiatkowski–Phillips–Schmidt–Shin (KPSS)* unit root test to detect stationarity, and *Osborn-Chui-Smith-Birchenhall (OCSB)* as the seasonal unit root test. A stepwise algorithm outlined by Hyndman and Khandakar (2008) is used to optimize the model parameters. The limited-memory Broyden–-Fletcher–Goldfarb–Shanno (BFGS) algorithm with optional box constraints (L-BFGS) is employed as the optimization algorithm. Figures S1 and S2 in the supplementary material illustrate the distribution of optimized hyperparameters for different stations, categorized by their historical data availability. ARIMA and SARIMA models of the *Darts* library are wrapped around the *Statsmodels* Python module.

3.4.2. Holt-Winters' exponential smoothing

Exponential smoothing is a statistical forecasting method proposed in the late 1950s (Holt, 1957 (re-printed in 2004); Winters, 1960). Exponential smoothing is a univariate method that uses weighted averages of past observations to forecast future steps. These weights decay exponentially as the observations get older. In other words, the more recent the observation, the higher the associated weight (Hyndman and Athanasopoulos, 2018). More information about this method is available in Kalekar (2004) and Hyndman and Athanasopoulos (2018). The model used in this study is wrapped around Statsmodels Holt-Winters' exponential smoothing. We used an additive model for both trend and seasonality components. The seasonal period is set to 12, and the trend component is damped.

3.4.3. Theta method

The Theta model proposed by Assimakopoulos and Nikolopoulos

(2000) is a univariate forecasting method based on modifying the local curvature of the time series using a coefficient “Theta” (a real number) applied to the second differences of the data. Theta method decomposes the original data into two or more lines, extrapolates them using appropriate forecasting models, and then combines their predictions to obtain the final forecast. This study uses the 4Theta model, a modified version of the original Theta method (Spiliotis et al., 2020). Through a manual search, we chose Theta = 2 for this model. The seasonality period is set to 12, and the type of seasonality is multiplicative. The Theta lines are combined with an additive model, and the trend mode is linear. For more information about the Theta and 4Theta models, readers are referred to Assimakopoulos and Nikolopoulos (2000) and Spiliotis et al. (2020), respectively.

3.5. Machine learning model (LightGBM)

This study employs the LightGBM as a machine learning forecasting tool. LightGBM, initially proposed by Ke et al. (2017) and developed by Microsoft as a free and open-source framework, provides an efficient implementation of the gradient boosting algorithm and reduces memory usage. Gradient boosting is an ensemble method where ensembles are constructed from decision tree models. Models are fit through a gradient descent optimization algorithm, where the loss gradient is minimized as the model is tuned. In the machine learning literature and competitions, gradient boosting and decision tree-based models outperform other regression algorithms when applied to tabular data (Shwartz-Ziv and Armon, 2022). Moreover, it is shown in the literature that gradient boosting algorithms work similarly in terms of accuracy and runtime, while some studies point to the superiority of LightGBM (Al Daoud, 2019). Therefore, this study employs LightGBM as its machine learning model. We feed 12 previous time steps as the input of the LightGBM model. More information about the LightGBM method can be found in Ke et al. (2017) and Al Daoud (2019).

3.6. Deep learning models

We conducted a manual hyperparameter tuning for all deep learning models. We objectively searched different values for hyperparameters and chose a subset that resulted in the best-performing model. We also conducted a grid search for the dropout rate. Data from the Davis CIMIS station, one of the best-maintained CIMIS stations with data from 1986, is used for hyperparameter tuning. When two sets of hyperparameters had similar performances, we chose the simpler model (i.e., the model with fewer trainable parameters). To facilitate the comparison between different models, the input of all DL models is ETo in the past 12 months.

3.6.1. N-BEATS

N-BEATS is a deep neural architecture based on backward and forward residual links and a very deep stack of fully connected layers (Oreshkin et al., 2019). N-BEATS was originally developed in 2019 to solve the univariate time series forecasting problem. N-BEATS architecture is fast to train and demonstrates state-of-the-art performance for different datasets. For more information about N-BEATS architecture, readers are referred to Oreshkin et al. (2019).

In this study, we employed the generic architecture outlined in Oreshkin et al. (2019). In this architecture, we used four stacks, with four blocks in each stack. We used four fully connected layers preceding the final backcast-forecast forking layer in each block, with 16 neurons in each layer. The expansion coefficient dimension is set to five, and the rectified linear unit (ReLU) is used as the activation function of the encoder/decoder intermediate layer. The grid search showed that N-BEATS works best without dropout; therefore, the dropout probability was set to zero. We trained the model over 100 epochs with a batch size of 32.

3.6.2. Long short-term memory (LSTM)

LSTM, proposed by Hochreiter and Schmidhuber (1997), is a recurrent neural network (RNN). Numerous studies in agriculture and hydrology have employed LSTM for forecasting purposes (Ni et al., 2020; Ghasemlounia et al., 2021; Hunt et al., 2022). RNN models are generally suitable for solving problems with sequential input data like time series. However, vanilla RNN models struggle with remembering information for an extended period, which is called a long-term dependency problem. LSTM architecture is designed exclusively to avoid this problem, which is the main advantage of this model. More information about LSTM can be found in Hochreiter and Schmidhuber (1997) and Van Houdt et al. (2020). Our LSTM model consists of one recurrent layer with 12 features in the hidden state. The dropout is set to zero for this model. The LSTM model is trained over 1,000 epochs with a batch size of 8.

3.6.3. Temporal convolutional network (TCN)

Although convolutional neural networks (CNNs) are commonly associated with raster data, they can also be used for sequential data with the proper modifications. The TCN, presented by Bai et al. (2018), is a generic convolutional architecture designed for sequence modeling. In this study, we use dilated TCN for forecasting. Readers are referred to Bai et al. (2018) for more information about this model. Our model has a kernel size of 6 and 18 filters. The base of the exponent determining the dilation on every level is set to two. We used weight normalization of the model and a dropout rate of 0.1. With a batch size of 32, we trained the model for 1,000 epochs.

3.6.4. Transformer model

Transformer is a state-of-the-art DL model introduced by Vaswani et al. (2017). Following an encoder-decoder structure, the transformer architecture does not rely on recurrence and convolutions to generate an output. The core feature of its architecture is the *multi-head attention* mechanism. In the case of sequential data, a multi-head attention mechanism can jointly attend to information at different positions in the sequence, making Transformer an appealing architecture for time series forecasting. The mechanism is also highly parallelizable, which makes the Transformer architecture suitable to be trained with GPUs. More information about the Transformer model can be found in Vaswani et al. (2017). In our model, we set the number of features in the transformer encoder/decoder inputs to 16, with one encoder layer and one decoder layer. We used four heads in the multi-head attention mechanism. The dimension of the feedforward network model is set to 128. We used ReLU as the activation function of the encoder/decoder intermediate layer. According to the grid search results, the dropout rate is set to 0.1. The model is trained with a batch size of 32 and over 1,200 epochs.

3.6.5. Temporal fusion Transformer (TFT)

TFT is a cutting-edge DL architecture introduced by Lim et al. (2021) for interpretable multi-horizon time series forecasting. TFT is a novel attention-based architecture that uses recurrent layers for local processing and a self-attention layer for long-term dependencies. TFT can learn temporal relationships at different scales and utilizes specialized components to select relevant features. Readers are referred to Lim et al. (2021) for detailed information about this architecture. In this study, we set the hidden state size of the TFT architecture to 16 and the hidden size for processing continuous variables to 8. We used one layer for the LSTM encoder/decoder. We used four attention heads, where a multi-head attention query is applied to the future (decoder) part only. A gated residual network is used as the feedforward network. PyTorch mean squared error (MSE) is employed as the loss function for training. We trained the model over 700 epochs with a batch size of 32.

3.7. Performance measures

In this study, three deterministic performance measures are used to

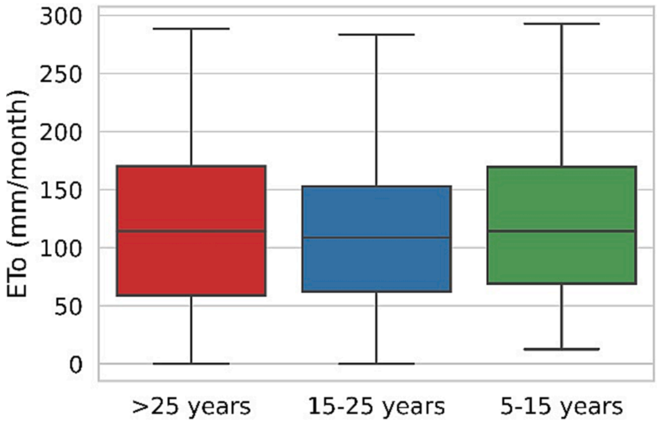

Fig. 3. Box plots of monthly reference evapotranspiration records of all the stations categorized by their data length.

evaluate the accuracy of forecasting models: root mean square error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(O_i - P_i)^2} \tag{4}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}(|O_i - P_i|) \tag{5}$$

$$R^2 = \left[\frac{\sum_{i=1}^{N}(O_i - \overline{O})(P_i - \overline{P})}{\sqrt{\sum_{i=1}^{N}(O_i - \overline{O})^2}\sqrt{\sum_{i=1}^{N}(P_i - \overline{P})^2}}\right]^2 \tag{6}$$

Where N is the number of time steps; O_i and P_i are observed and predicted monthly ETo values at i^{th} time step, respectively; $\overline{O}$ and $\overline{P}$ are the mean values of observations and predictions, respectively. Lower RMSE and MAE values and higher R^2 values indicate higher accuracies and better performances.

4. Results and discussion

4.1. Stations and time series characteristics

Stations from different categories are almost uniformly distributed over the study area (Fig. 1). Therefore, we hypothesize that no systematic bias is introduced to the results and the categorization caused by the stations' location. However, as Fig. 1 shows, there are no stations with 15 to 25 years of data in northern California. It should be noted that since the primary goal of the CIMIS program is to assist farmers with irrigation management, most of the CIMIS stations are located in farming/irrigation-oriented regions of California. Therefore, there are more stations in central and southern California and fewer in northern California. To test our hypothesis and confirm that the results are not affected by the geographical and climatic differences among categories, we analyzed the long-term monthly ETo records of the stations. As can be inferred from Fig. 3, no significant difference exists between the distribution of records from different categories. It should be noted that box plots in Fig. 3 represent all the observations from all the stations used in this study.

Fig. 4 depicts the results of time series decomposition for the Davis CIMIS station. It should be noted that decomposition is time series-specific, meaning only one time series can be decomposed at a time. Here we show decomposition of ETo time series for a well-maintained CIMIS station with adequately long historical data (Davis station).

Table 1
Runtime and number of trainable parameters of the forecasting models used in this study.

Model	Runtime (seconds)	Number of trainable parameters of deep learning models
ARIMA	168 (152 + 16)	–
SARIMA	266 (152 + 114)	–
Holt-Winters	4	–
Theta	1	–
LightGBM	2	–
N-BEATS	59	~ 20,700
LSTM	645	733
TCN	253	~ 4,300
Transformer	367	~ 12,100
TFT	473	~ 15,400

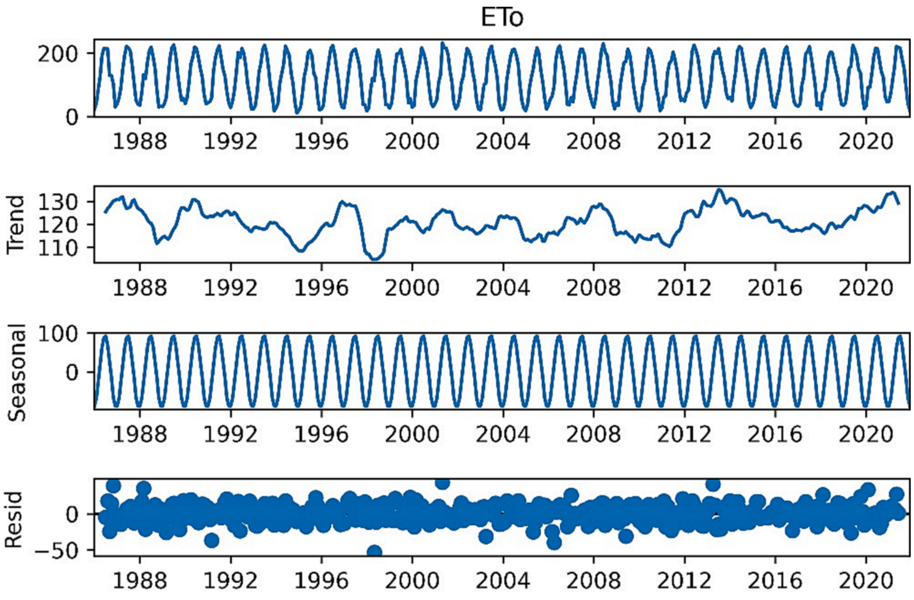

Fig. 4. Results of time series decomposition for Davis CIMIS station.

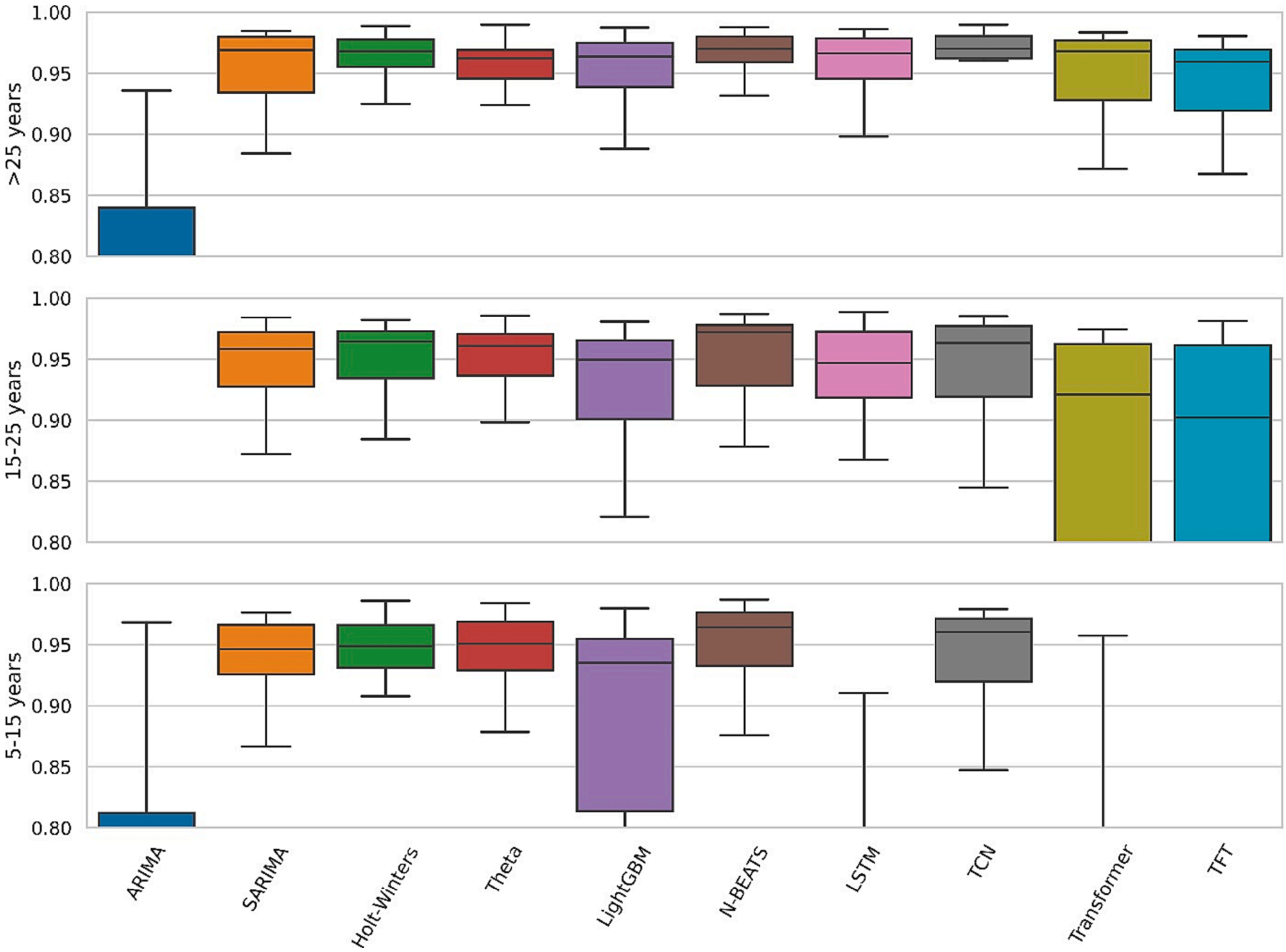

Fig. 5. R^2 results of forecasting models in the case of one month ahead reference evapotranspiration forecasting.

This station is located in the Central Valley, where many other CIMIS stations are placed. Expectedly, considerable seasonality is present in the time series (Fig. 4). Although Fig. 4 suggests an increasing trend in the minimum monthly ETo values for the Davis CIMIS station, the trend component does not control the observed time series. More information about California, the trends of ETo, and its meteorological driving factors can be found in Ahmadi et al. (2022).

4.2. Model complexity

To have a more in-depth comparison of forecasting models, in addition to their accuracy, we analyzed their complexity and computational cost. Table 1 presents the complexity of DL models by providing the number of their trainable parameters. Also, this table shows the runtime of all forecasting models as a measure of their computational cost. The runtime is based on the Google Colab platform without any hardware accelerator (e.g., GPU) and represents the time required for training the model on the Davis CIMIS station and making one-step ahead forecasts on the test set. In the case of ARIMA and SARIMA models, runtime consists of two parts: 1) optimizing the hyperparameters for Davis station using the pmdarima library (152 s); 2) training the model with optimized hyperparameters and making forecasts on the test set.

As expected, given their higher complexity, DL models generally have a higher runtime than statistical and machine learning models. However, Table 1 shows that N-BEATS has a significantly lower runtime than other DL and even ARIMA and SARIMA models. The main reason is that N-BEATS architecture is very fast to converge. As mentioned in the methodology section, we used only 100 epochs to train the N-BEATS model, whereas other DL models required a much higher number of epochs to reach minimum loss values. Notably, the number of trainable parameters in the N-BEATS model is not lower than in other DL models. Quite the contrary, Table 1 reveals that the N-BEATS architecture used in this study has the highest number of trainable parameters among all DL models. Our findings align with Oreshkin et al. (2019), demonstrating the computational efficiency of the N-BEATS model.

Unlike the N-BEATS model, LSTM architecture is proven very slow to train. Although LSTM has the lowest number of trainable parameters among the DL models, it has the most extended runtime (Table 1). According to Yu et al. (2019), the slow training of LSTM can be attributed to its backward propagation through time. The literature suggested some methods to speed up the convergence of LSTM training (e.g., a convex-based LSTM network introduced by Wang, 2017). Our study's findings also suggest no direct relationship between the number of trainable parameters of a DL model and its runtime. Based on these findings, we hypothesize that the overall architecture of a DL model is more important than merely the number of trainable parameters in the model's computational cost.

4.3. Forecasting accuracy

We used three measures of performance to evaluate the accuracy of forecasting models: RMSE, MAE, and R^2. We show R^2 results in the main text and other measures in the supplementary material. Since we have multiple time series with different scales in each category, and there is a meaningful effect of seasonality in the magnitude of monthly ETo at

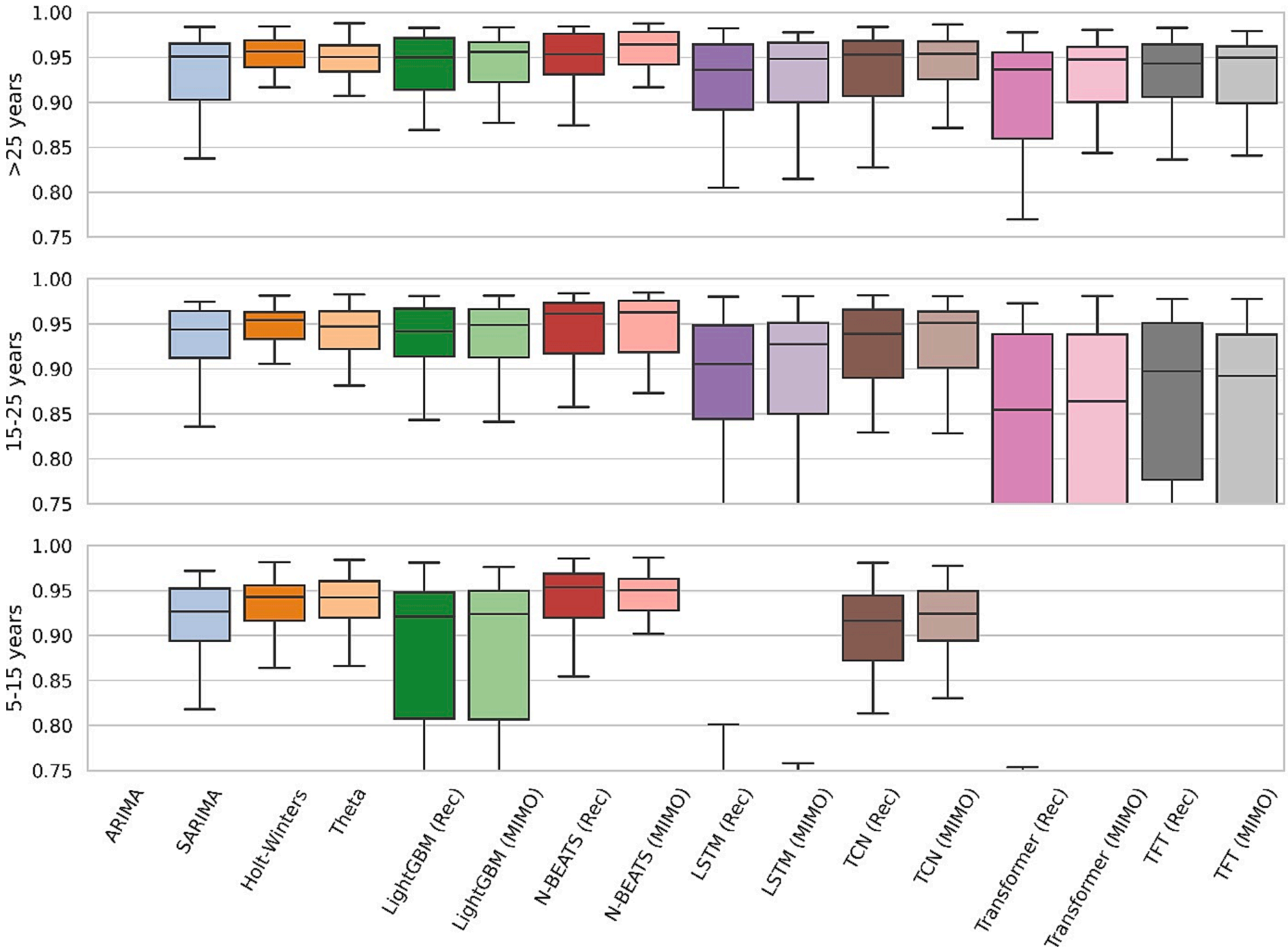

Fig. 6. R^2 results of forecasting models in the case of three months ahead reference evapotranspiration forecasting.

various timesteps, a scale-free accuracy measure like R^2 is preferred. According to Fildes et al. (2022), results of non-scale-free measures (e.g., RMSE) should be handled cautiously in the case of time series forecasting.

Figs. 5, 6, and 7 illustrate the models' accuracy for one month, three months, and six months lead time ETo forecasting, respectively. As mentioned earlier, there is no difference between recursive and MIMO strategies in one-step-ahead forecasting. Therefore, Fig. 5 shows a single box for each ML and DL model. The ARIMA model forecasts ETo less accurately than the other models, as its performance falls considerably short compared to other models (Figs. 5-7). This is mainly due to the lack of seasonality features in the ARIMA model. As Fig. 4 illustrates, there is a significant seasonality in our data, and it is no surprise that a model that does not consider seasonality is not a good choice for ETo data. On the other hand, the seasonal ARIMA (SARIMA) model results in accurate monthly ETo forecasts for various forecasting horizons and with different amounts of available historical data (Figs. 5-7). This finding aligns with Aghelpour et al. (2022) and Ashrafzadeh et al. (2020), who report on the goodness of the SARIMA model for monthly ETo forecasting.

As Figs. 5-7 show, LSTM, Transformer, and TFT models have difficulty working with smaller training sets (i.e., short data). In other words, these models are *data-hungry* and require larger training sets to work optimally. Transformer and TFT models are even more sensitive to the input data length, as their performance drops more severely than LSTM in the case of smaller training sets. Therefore, these models are suitable for forecasting agrometeorological variables only when enough input data is available. Recent literature offers techniques to make these models more *data-efficient* (e.g., Hassani et al. (2021)).

Contrary to the data-hungry models, N-BEATS and TCN show low sensitivity to the input data length. The most data-efficient DL model is N-BEATS, as its performance is least affected by the length of the training set. LightGBM model is also sensitive to the length of the data, but much less than more complex DL models. Statistical forecasting models do not show profound sensitivity to the training data length. Therefore, our results point to the overall data efficiency of statistical models.

Expectedly, all models perform better when forecasting the near future (e.g., one step ahead). In other words, the forecasting accuracy of the models drops when the forecasting horizons increase. Fig. 7 and figure S8 in the supplementary material suggest that N-BEATS is the best model for longer forecasting horizons, followed by Theta and Holt-Winters models. Our findings do not suggest a significant difference between the ML and DL models' accuracy under recursive and MIMO strategies. It can be inferred that neither of these strategies has a systematic advantage, at least in our case study.

Our findings generally point to the superior performances of simpler forecasting models compared to more complex DL models. Results illustrate that Theta and Holt-Winters methods work almost as well as the most accurate DL models while having much lower run time and complexity. We hypothesize that these models are more suitable for less complex time series, like ETo data. Another important note is that although numerous hydrological and agricultural studies employ the SARIMA model (Ashrafzadeh et al., 2020; Aghelpour et al., 2022), our study reveals that other statistical models outperform this model. This is even more important as we compare the higher computational costs of

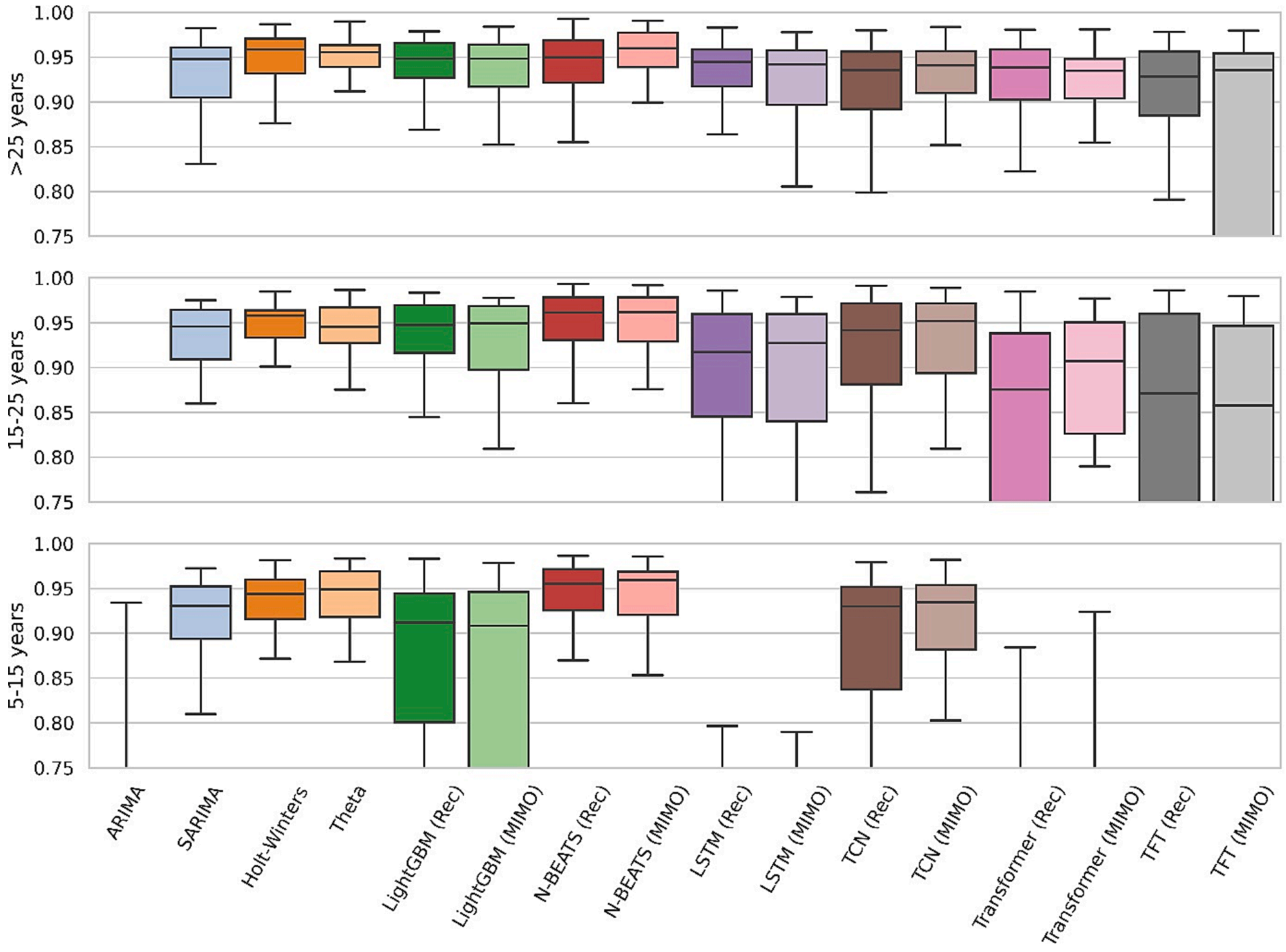

Fig. 7. R^2 results of forecasting models in the case of six months ahead reference evapotranspiration forecasting.

LSTM with other statistical models (i.e., Theta and Holt-Winters; see Table 1).

Considering the computational cost and the forecasting accuracy, N-BEATS is the best model among the models evaluated in this study. N-BEATS is very fast to train (Table 1), data-efficient, and can accurately forecast long horizons (Figs. 5-7). One reason for the superior performance of N-BEATS might be that this DL model was initially developed for univariate time series forecasting. In other words, this model works better than other DL models for our case because, inherently, it is the most appropriate architecture.

4.4. *Observed vs. Forecasted time series*

Fig. 8 depicts the forecasted time series of the N-BEATS (i.e., the best DL model) and Holt-Winters (i.e., the best statistical model) models against the observed values for different forecasting horizons. Davis, Auburn, and Woodland stations are chosen to represent stations with various data lengths (available data from 1986, 2005, and 2011, respectively). All these stations are placed in ETo zone 14 (Mid-central Valley) and close to each other. This proximity minimizes the effects of climate and geographic characteristics on the models' performance. Fig. 8 confirms the previous findings on the similarity of performances between the best DL model (i.e., N-BEATS) and the statistical model (i.e., Holt-Winters). This figure indicates that when monthly ETo values are increasing (when it is getting warmer), the forecasted values of both models tend to underestimate observed values.

On the other hand, when monthly ETo values are decreasing (i.e., when it is getting cooler), forecasted values tend to overpredict observed values. Again, this pattern happens for both N-BEATS and Holt-Winters models. This behavior can be attributed to DL and statistical models using past time series values to forecast and tend toward the averaged values. In the case of the Holt-Winters method, this is more evident, as this method uses weighted averages of past observations to forecast new values, while the exponential smoothing technique gives more importance (i.e., larger weights) to more recent observations. Given the similarity between the forecasted values of Holt-Winters and the DL model (Fig. 8), we speculate that a DL method might learn the same pattern in input data. Due to this tendency toward predicting average values of past observations, a postprocessing model that adds/subtracts residuals to the predicted ETo data based on the slope of predictions can be explored as an alternative to improve forecasts.

5. Summary and conclusion

This study evaluated the performance, computational cost, and complexity of well-known statistical and state-of-the-art machine learning and deep learning models for forecasting reference evapotranspiration (ETo). Monthly ETo in 107 standardized stations in California was used for this analysis. Stations were categorized according to historical data availability, and models were tested for various forecasting horizons. Recursive and MIMO forecasting strategies were evaluated for machine learning (ML) and deep learning (DL) models. Significant findings of this study and insights for future research include:

1. Complex deep learning models (e.g., LSTM and Transformer) are not data-efficient enough for agrometeorological forecasting. Given the

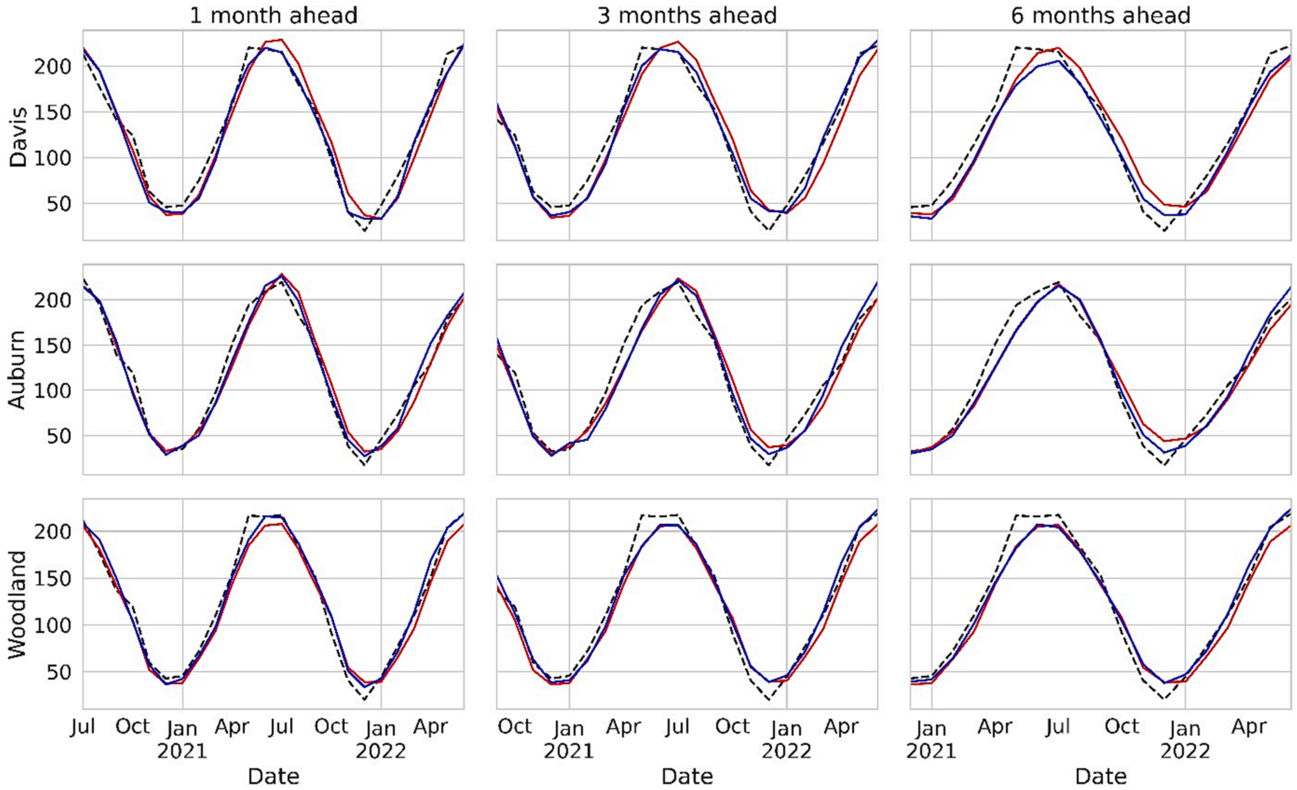

Fig. 8. Time series of observed and forecasted monthly reference evapotranspiration for three CIMIS stations. Dashed black lines, solid red lines, and solid blue lines depict observed values, Holt-Winters model forecasted values, and N-BEATS model forecasted values, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

scarcity of agrometeorological data and the unavoidable data gaps in time series measured by micrometeorological sensors, more research is needed on the data efficiency of DL forecasting models. One potential solution for this issue is developing pre-trained DL models that can be retrained for short-range agrometeorological data (aka, transfer learning; e.g., Raei et al., 2022). Furthermore, generative adversarial networks can potentially help alleviate this problem.

2. Simpler statistical forecasting models work as well or even better than state-of-the-art deep learning models. This might be due to the inherent simplicity of monthly reference evapotranspiration. Further studies are required to analyze the performance of these models for more complex agrometeorological time series.
3. N-BEATS was the best overall deep learning forecasting model. N-BEATS is very fast to cooverge, requiring only 100 epochs to train, while other DL models need a much higher number of training epochs (e.g., 1,000 in the case of LSTM and 1,200 for Transformer). N-BEATS is also very data-efficient; this model can forecast monthly ETo accurately even when less than fifteen years of historical data is available.
4. Although the most popular statistical method for agrometeorological time series forecasting is SARIMA, our findings reveal that Holt-Winters and Theta methods are more data-efficient, less computationally expensive, and generally more accurate than SARIMA. Future studies can evaluate their performance in other cases, for example yield forecasting.
5. Our findings reveal no significant difference between recursive and MIMO strategies. A more in-depth study is needed to analyze the effects of forecasting strategy on the performance of ML and DL models, primarily when covariates (e.g., from numerical weather predictions or other sensors) are used to inform ETo forecasting.

Some of the limitations of the current research that introduce opportunities for future studies include the following:

1. This research focused on monthly data with a clear seasonality component. More studies are needed to evaluate the performance of forecasting models for higher-frequency input data (e.g., daily data).
2. This study focused on univariate time series, while several agrometeorological cases can benefit from covariates and exogenous variables. According to the findings of the M5 competition, multivariate models with informative exogenous variables are expected to outperform univariate models (Makridakis et al., 2022). We hypothesize that introducing exogenous variables to deep learning models will boost their performance. However, it should be noted that statistical models are generally restrained to one variable. When exogenous variables are available, multivariate deep learning models are expected to outperform simpler statistical models. Future studies can test these hypotheses.
3. This study trains models for each station. However, deep learning models can benefit from *cross-learning*. Cross- or global-learning refers to learning from multiple series to extract information from the global data set (Makridakis et al., 2022). This is especially advantageous in agrometeorological forecasting, as the time series from various stations and sensors may share common characteristics (e.g., seasonality). For instance, ungauged regions with poor data history can benefit from neighboring stations and cross-learning strategies for accurate forecasting. Future studies can cast light on the advantages of cross-learning in agrometeorological forecasting.
4. This research focused on California as the case study. Although California is a climatically diverse case study with various ecosystems, more studies are required to evaluate the findings of this research in other climate conditions. However, as the agrometeorological time series are often similar to the time series used in this study in terms of a dominant seasonal component, we hypothesize that the findings of this research are relevant to other regions and climates.

CRediT authorship contribution statement

Arman Ahmadi: Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization, Conceptualization, Methodology. **Andre Daccache:** Conceptualization, Methodology, Writing – review & editing, Supervision. **Mojtaba Sadegh:** Conceptualization, Methodology, Writing – review & editing. **Richard L. Snyder:** Conceptualization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compag.2023.108424.

References

Aghelpour, P., Varshavian, V., Khodamorad Pour, M., Hamedi, Z., 2022. Comparing three types of data-driven models for monthly evapotranspiration prediction under heterogeneous climatic conditions. Sci. Rep. 12 (1), 1–19.

Ahmadi, A., Daccache, A., Snyder, R.L., Suvočarev, K., 2022. Meteorological driving forces of reference evapotranspiration and their trends in California. Sci. Total Environ. 849, 157823.

Al Daoud, E., 2019. Comparison between XGBoost, LightGBM and CatBoost using a home credit dataset. International Journal of Computer and Information Engineering 13 (1), 6–10.

Allen, R.G., Pereira, L.S., Raes, D., Smith, M., 1998. Crop evapotranspiration-Guidelines for computing crop water requirements-FAO Irrigation and drainage paper 56. Fao, Rome 300 (9), D05109.

Ashrafzadeh, A., Kişi, O., Aghelpour, P., Biazar, S.M., Masouleh, M.A., 2020. Comparative study of time series models, support vector machines, and GMDH in forecasting long-term evapotranspiration rates in northern Iran. J. Irrig. Drain. Eng. 146 (6), 04020010.

Assimakopoulos, V., Nikolopoulos, K., 2000. The theta model: a decomposition approach to forecasting. Int. J. Forecast. 16 (4), 521–530.

Bai, S., Kolter, J.Z. and Koltun, V., 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv: 1803.01271*.

Chia, M.Y., Huang, Y.F., Koo, C.H., 2021a. Improving reference evapotranspiration estimation using novel inter-model ensemble approaches. Comput. Electron. Agric. 187, 106227.

Chia, M.Y., Huang, Y.F., Koo, C.H., 2021b. Swarm-based optimization as stochastic training strategy for estimation of reference evapotranspiration using extreme learning machine. Agric Water Manag 243, 106447.

Chia, M.Y., Huang, Y.F., Koo, C.H., Ng, J.L., Ahmed, A.N., El-Shafie, A., 2022. Long-term forecasting of monthly mean reference evapotranspiration using deep neural network: A comparison of training strategies and approaches. Appl. Soft Comput. 126, 109221.

Contreras, J., Espinola, R., Nogales, F.J., Conejo, A.J., 2003. ARIMA models to predict next-day electricity prices. IEEE Trans. Power Syst. 18 (3), 1014–1020.

Diodato, N., Bellocchi, G., 2007. Modeling reference evapotranspiration over complex terrains from minimum climatological data. Water Resources Research 43 (5).

Ferreira, L.B., da Cunha, F.F., 2020. Multi-step ahead forecasting of daily reference evapotranspiration using deep learning. Comput. Electron. Agric. 178, 105728.

Fildes, R., Ma, S., Kolassa, S., 2022. Retail forecasting: Research and practice. Int. J. Forecast. 38 (4), 1283–1318.

Ghasemlounia, R., Gharehbaghi, A., Ahmadi, F., Saadatnejadgharahassanlou, H., 2021. Developing a novel framework for forecasting groundwater level fluctuations using Bi-directional Long Short-Term Memory (BiLSTM) deep neural network. Comput. Electron. Agric. 191, 106568.

Gocić, M., Motamedi, S., Shamshirband, S., Petković, D., Ch, S., Hashim, R., Arif, M., 2015. Soft computing approaches for forecasting reference evapotranspiration. Comput. Electron. Agric. 113, 164–173.

Hassani, A., Walton, S., Shah, N., Abuduweili, A., Li, J. and Shi, H., 2021. Escaping the big data paradigm with compact transformers. *arXiv preprint arXiv:2104.05704*.

Herzen, J., Lässig, F., Piazzetta, S.G., Neuer, T., Tafti, L., Raille, G., Van Pottelbergh, T., Pasieka, M., Skrodzki, A., Huguenin, N., Dumonal, M., 2022. Darts: User-friendly modern machine learning for time series. J. Mach. Learn. Res. 23 (124), 1–6.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.

Holt, C.C., 2004. Forecasting seasonals and trends by exponentially weighted moving averages. Int. J. Forecast. 20 (1), 5–10.

Hunt, K.M., Matthews, G.R., Pappenberger, F., Prudhomme, C., 2022. Using a long short-term memory (LSTM) neural network to boost river streamflow forecasts over the western United States. Hydrol. Earth Syst. Sci. Discuss. 1–30.

Hyndman, R.J., Athanasopoulos, G., 2018. Forecasting: principles and practice. Otexts.

Hyndman, R.J., Khandakar, Y., 2008. Automatic time series forecasting: the forecast package for R. J. Stat. Softw. 27, 1–22.

Kalekar, P.S., 2004. Time series forecasting using holt-winters exponential smoothing. Kanwal Rekhi School of Information Technology 4329008 (13), 1–13.

Karbasi, M., Jamei, M., Ali, M., Malik, A., Yaseen, Z.M., 2022. Forecasting weekly reference evapotranspiration using Auto Encoder Decoder Bidirectional LSTM model hybridized with a Boruta-CatBoost input optimizer. Comput. Electron. Agric. 198, 107121.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y., 2017. Lightgbm: A highly efficient gradient boosting decision tree. Adv. Neural Inf. Proces. Syst. 30.

Lim, B., Arık, S.Ö., Loeff, N., Pfister, T., 2021. Temporal fusion transformers for interpretable multi-horizon time series forecasting. Int. J. Forecast. 37 (4), 1748–1764.

Lund, J., Medellin-Azuara, J., Durand, J., Stone, K., 2018. Lessons from California's 2012–2016 drought. J. Water Resour. Plan. Manag. 144 (10), 04018067.

Makridakis, S., Spiliotis, E., Assimakopoulos, V., 2022. Predicting/hypothesizing the findings of the M5 competition. Int. J. Forecast. 38 (4), 1337–1345.

Ni, L., Wang, D., Singh, V.P., Wu, J., Wang, Y., Tao, Y., Zhang, J., 2020. Streamflow and rainfall forecasting by two long short-term memory-based models. J. Hydrol. 583, 124296.

Oreshkin, B.N., Carpov, D., Chapados, N. and Bengio, Y., 2019. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv: 1905.10437*.

Pereira, L.S., Perrier, A., Allen, R.G., Alves, I., 1999. Evapotranspiration: concepts and future trends. J. Irrig. Drain. Eng. 125 (2), 45–51.

Raei, E., Asanjan, A.A., Nikoo, M.R., Sadegh, M., Pourshahabi, S., Adamowski, J.F., 2022. A deep learning image segmentation model for agricultural irrigation system classification. Comput. Electron. Agric. 198, 106977.

Richetti, J., Diakogianis, F.I., Bender, A., Colaço, A.F., Lawes, R.A., 2023. A methods guideline for deep learning for tabular data in agriculture with a case study to forecast cereal yield. Comput. Electron. Agric. 205, 107642.

Samaniego, L., Thober, S., Wanders, N., Pan, M., Rakovec, O., Sheffield, J., Wood, E.F., Prudhomme, C., Rees, G., Houghton-Carr, H., Fry, M., 2019. Hydrological forecasts and projections for improved decision-making in the water sector in Europe. Bull. Am. Meteorol. Soc. 100 (12), 2451–2472.

Shwartz-Ziv, R., Armon, A., 2022. Tabular data: Deep learning is not all you need. Information Fusion 81, 84–90.

Spiliotis, E., Assimakopoulos, V., Makridakis, S., 2020. Generalizing the theta method for automatic forecasting. Eur. J. Oper. Res. 284 (2), 550–558.

Taieb, S.B., Bontempi, G., Atiya, A.F., Sorjamaa, A., 2012. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. Expert Syst. Appl. 39 (8), 7067–7083.

Van Houdt, G., Mosquera, C., Nápoles, G., 2020. A review on the long short-term memory model. Artif. Intell. Rev. 53 (8), 5929–5955.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Proces. Syst. 30.

Winters, P.R., 1960. Forecasting sales by exponentially weighted moving averages. Manag. Sci. 6 (3), 324–342.