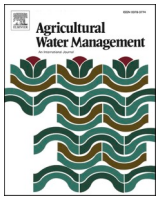
Agricultural Water Management 284 (2023) 108367

Contents lists available at ScienceDirect

Agricultural Water Management

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



New memory-based hybrid model for middle-term water demand forecasting in irrigated areas

R. González Perea [a,*], I. Fernández García [b], E. Camacho Poyato [a], J.A. Rodríguez Díaz [a]

[a] Department of Agronomy (Unit of Excellence María de Maeztu), University of Córdoba, Campus Rabanales, Edif.da Vinci, 14071 Córdoba, Spain
[b] Dept. of Electrical Engineering and Automatic, University of Córdoba, Campus Rabanales, Edif. da Vinci, 14071 Córdoba, Spain

ARTICLE INFO

Handling Editor - J.E. Fernández

Keywords:
Artificial intelligence
Irrigation
Pressurized water networks
Deep learning
Machine learning

ABSTRACT

The energy demand and their associated costs in pressurized irrigation networks together with water scarcity are currently causing serious challenges for irrigation district's (ID) managers. Additionally, most of the new water distribution networks in IDs have been designed to be operated on-demand complexing ID managers the daily decision-making process. The knowledge of the water demand several days in advance would facilitate the management of the system and would help to optimize the water use and energy costs. For an efficient management and optimization of the water-energy nexus in IDs, longer term forecasting models are needed. In this work, a new hybrid model (called LSTMHybrid) combining Fuzzy Logic (FL), Genetic Algorithm (GA), LSTM encoder-decoder and dense or full connected neural networks (DNN) for the one-week forecasting of irrigation water demand at ID scale has been developed. LSTMHybrid was developed in Python and applied to a real ID. The optimal input variables for LSTMHydrid were mean temperature (°C), reference evapotranspiration (mm), solar radiation (MJ m^{-2}) and irrigation water demand of the ID (m^3) from 1 to 7 days prior to the first day of prediction. The optimal LSTMHybrid model selected consisted of 50 LSTM cells in the encoder submodel, 409 LSTM cells in the decoder submodel and three hidden layers in the DNN submodel with 31, 96 and 128 neurons in each hidden layer, respectively. Thus, LSTMHybrid had a total of 1.5 million parameters, obtaining a representativeness higher than 94 % and an accuracy around of 20 %.

1. Introduction

Irrigation is regarded as one of the main adaptations to support crop production in response to population growth (FAO, 2016, 2015; Rodríguez Díaz et al., 2007). However, it must be a precision activity where the right amount of water is applied just when it is needed by the crop (Hedley et al., 2014). Thus, pressurized irrigation systems which are managed by irrigation districts (IDs) are one of the best options to increase water use efficiency (Daccache et al., 2014a, 2014b). Additionally, in last two decades, the flexibility in the irrigation operation by farmers has been increase because of the new pressurized irrigation networks (after modernization process). These new water distribution networks for irrigation make water available to farmers 24 h a day. Thus, farmers can decides every day its irrigation scheduling (how and when to irrigate) (Playán and Mateos, 2006; Rodríguez-Díaz et al., 2011; Tarjuelo et al., 2015). However, this increase in the availability of water by farmers makes the integrated management of the ID a complex task (the hydraulic systems must work under a wide range of hydraulic operation conditions). Managers of these IDs do not know when their users (farmers) are going to irrigate and at what time, making it difficult to optimize all the system (pumping stations, failures, repairs, energy purchases, etc.). Thus, the uncertainty associated with irrigation water demands in water distribution networks which are operated on demand (Plusquellec, 2009) and the uncertainty of the electricity market (In Spain, last deep change on 1st June 2021) are generating challenges in the daily management of irrigation districts, mainly in arid and semiarid regions such as Spain. In addition, an accurate prediction of water demand would allow for optimal network management measures to reduce energy demand. Nowadays, in the new age of digitalization, ICTs (Information and Communication Technologies) and intelligent algorithms based on Artificial intelligence (AI) are essential to increase and improve the precision agriculture management, particularly in the control of the water-energy management in pressurized irrigation systems (Kamilaris and Prenafeta-Boldú, 2018; Wolfert et al., 2017). Traditional strategies to save water and energy in ID such as critical point management, sectorization, etc. (Derardja et al., 2019; Fernández García et al., 2016;

* Corresponding author.
E-mail address: g72goper@uco.es (R. González Perea).

https://doi.org/10.1016/j.agwat.2023.108367
Received 25 January 2023; Received in revised form 4 May 2023; Accepted 11 May 2023
Available online 15 May 2023

González Perea et al., 2016; Rodríguez Díaz et al., 2012) frequently reduce farmers' water management flexibility, diminishing the positive aspects of the irrigation modernization process (on-demand irrigation). Thus, new water and energy optimization strategies ID are necessary, but without reducing the decision-making power of farmers, based on intelligent models which forecasts this farmer behavior. Traditionally, one of the main drawbacks to implementing new intelligent data-driven models is the availability and cost of obtaining this information. However, the recent revolution of the ICTs, as well as the development of new low-cost sensors capable to automatically collect on real time thousands of robust data and the exploitation of AI have modified the technological environment of the current irrigated agriculture. Based on machine learning from data, all these technologies offer new possibilities for water-energy nexus management in IDs leading to increased use efficiencies of these resources, making the IDs more sustainable, in environmental, social and economic aspects.

Based on AI techniques, water demand forecasting will allow the entire water distribution network (from pumping station to hydrant) to be adapted to operate as efficiently as possible, reducing the uncertainty caused by on-demand irrigation, optimizing all its resources. Thus, real water demand forecasting could be one of the main tools to improve IDs operation and help managers in the daily decision-making processes but without reducing farmers' degree of freedom.

AI, ICTs and bigdata have been applied to solve some forecasting issues (mainly of a time series nature) in precision irrigation, IDs and pressurized irrigation networks. Since it was first published in its full version in the early 2000s (Graves and Schmidhuber, 2005), Long Short-Term Memory (LSTM) neural network is the most widely used type of artificial neural network for solving time series forecasting problems. Its recurrent connection adds state or memory to this neural network and allow it to learn and harness the ordered nature of input time series. Thus, LSTM has become the most cited neural network of the 20th century. At ID scale, ground water level estimations and irrigation demand forecasting are two of the main topics addressed for water-energy optimization based on AI and bigdata. A new model based on LSTM neural network for water table depth forecasting in agricultural areas was developed by Zhang et al. (2018). This model was applied in five sub-areas of Hetao Irrigation District (China) using a time series of 14 years. Results showed that the developed model can serve as an alternative approach (as opposed of physical models) predicting water table depth, especially in areas where hydrogeological data are difficult to obtain. A combination of LSTM neural network with ARIMA (autoregressive integrated moving average) model was proposed by Sheikh Khozani et al. (2022) to predict ground water level Yazd-Ardekn Plain in Iran. The architecture of this hybrid model was optimized by the Salp Swarm Algorithm (SSA), sine cosine optimization algorithm (SCOA), particle swarm optimization algorithm (PSOA), and genetic algorithm (GA). Despite the lower values of MAE (mean absolute error) which ranged from 5 % to 47 %, general results indicated that an increased forecasting horizon reduced the accuracy of the model. This problem is regularly found in forecasting model based on memory such as LSTM and all its modifications.

No many works can be found for farmer behavior' forecasting and so the real irrigation demand in IDs. The estimation of the water demand in an ID can be approached at different spatial and temporal scales. At spatial scale, farm and irrigation district scale can be distinguished. On the other hand, at temporal scale, short (one day), middle (around a week), and long (several weeks) terms scales can be recognized. At present, there are not long-term forecasting models either at farm or water user association level. However, some forecasting models have been developed for the rest of the spatial and temporal scales.

At short-term time scale, Pulido-Calvo and Gutiérrez-Estrada (2009) developed a hybrid model combining DNNs (Dense Neural Networks), fuzzy logic (FL) and GA (Genetic Algorithm) to forecast one-day ahead daily water demand at irrigation district spatial scale. A major limitation of this model was the determination of the neural network architecture by trial and error. Consequently, the standard error prediction (SEP) of the model was around 25 %. This limitation was addressed by González Perea et al. (2015) which developed a new methodology to optimize the DNN architecture by GA to the short-term forecasting of daily irrigation water demand at irrigation district level. Results showed SEP values close to 10 % improving the accuracy obtained by other models to date. The short-term forecasting at farm level was addressed in previous works (González Perea et al., 2019, 2018, 2021), where three models addressed the forecast of the occurrence of the irrigation events (González Perea et al., 2019), the daily applied irrigation depth (González Perea et al., 2018) and the hourly distribution (according to electricity tariff) of the daily applied irrigation depth (González Perea et al., 2021). Forouhar et al. (2022) developed a hybrid framework by incorporating existing physical knowledge (conceptual model to understood factors leading to crop water needs using observation data) of the system into a data-driven model to predict the irrigation water demand up to 7 days ahead for an irrigation district in Victoria, Australia. However, although results showed that the integration of physical system understanding into data-driven models can improve the performance of the predictions, the accuracy of the results are not enough for an integrated management of water and energy resources in IDs. Gonzalez Perea et al. (2021) developed a model to predict the demand seven days ahead but at farm level. They combined AI techniques, satellite remote sensing and open-source climate data to optimize the irrigation water forecasting at farm level for a week in advance. However, although SEP values ranged from 17 % to 19 %, the one-week estimation was addressed as a single point, not distinguishing between days but for a whole week.

Consequently, the accuracy of short-term forecasting models for different spatial scales are good but are not enough to be considered in the day-to-day management of IDs. Short-term forecasting models do not have enough time resolution for a holistic management of the IDs. Thus, Longer term forecasting models would be much more useful for managers to know in advance how much water will be required and hence how much energy should be needed. This will involve a more efficient use of water and energy resources and, a cost-effective purchase of the electric energy. However, the already developed models are either not developed yet or the accuracy achieved is not adequate for its incorporation into the daily management of the ID. Thus, with the main objective of overcoming these limitations, in this work a new hybrid model combining FL, GA, LSTM encoder-decoder and dense or full connected neural networks (DNN) for the one-week forecast of irrigation water demand at irrigation district scale has been developed. This model, called LSTMHybrid, is based on artificial memory and has been developed in Python programming language (v3.9) (Van et al., 2009) and tested in a real ID.

2. Methodology

2.1. Study area and data source

The study area where the developed methodology has been applied is in Extremadura region, Southwest Span. Fig. 1 shows the location of this real ID called *Canal del Zújar Irrigation District (CZID)*. This ID consists of ten independent hydraulic sectors. The methodology is applied on sector II of this ID. *CZID* applied irrigation water through a pressurized irrigation network which is organized on-demand. *CZID* sector II convers a total irrigated area of 2691 ha and 191 hydrants. Tomato, rice and maize are the main crops covering more than 90 % of the total irrigated area. Each hydrant of CZID sector II has a flowmeter which automatically records the hourly volume of water applied. The complete cleaned dataset of the daily irrigation water demand contained a total of 1849 measures from 2015 to 2022 irrigation seasons. This dataset was randomly divided into three subsets: training dataset (70 % of the total measures of the original dataset), validation dataset (15 % of the total measures of the original dataset) and testing dataset (15 % of the total

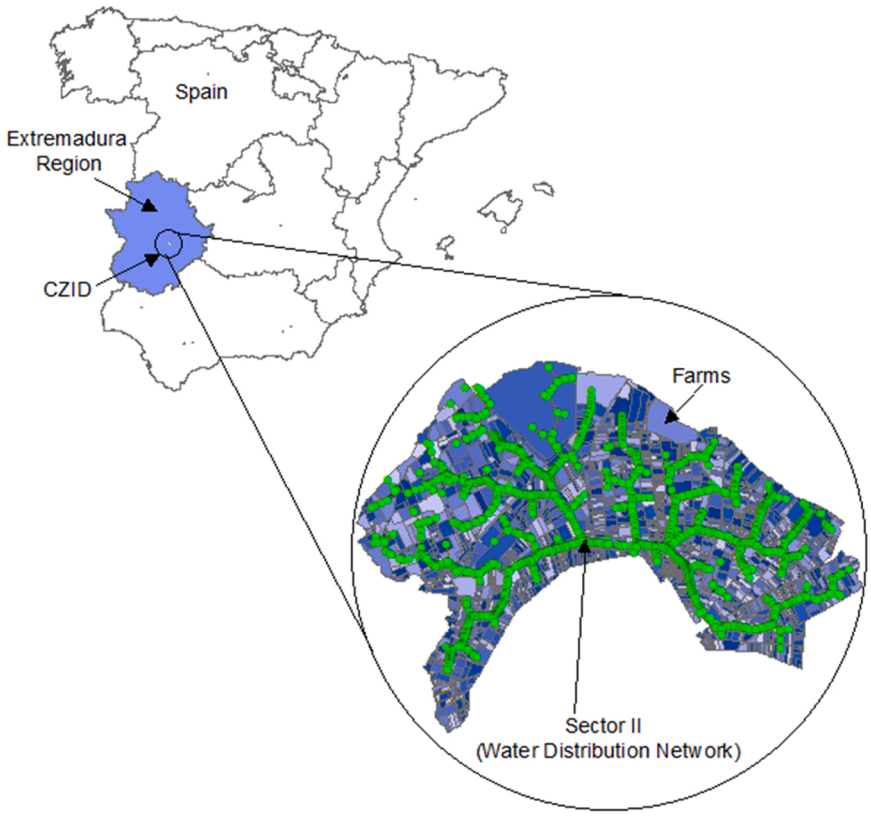


Fig. 1. Location of Sector II of the Canal del Zújar Irrigation District (CZID).

measures of the original dataset). The UTM coordinates of the closest agro-weather station are 252,468, 431,917 and 30 for x, y and zone, respectively. The average annual evapotranspiration and precipitation are 1296 mm and 390 mm, respectively. The average maximum and minimum temperatures range from 23 °C to 35.1 °C.

2.2. Problem approach

The stated problem focused on the development of a new methodology to determinate the optimal forecasting model of water demand at irrigation district level one-week ahead in advance.

To address this issue, the problem has been approached as time series forecasting problem. Many AI techniques, particularly in deep learning, can address time series prediction. However, due to the nature of their architecture, not all the deep learning models are specially designed for this issue. Typically, RNNs (Recurrent Neural Networks) are the family of deep learning models for processing sequential data, i.e. time series. Previous scientific works have highlighted that the most effective time series models used in practical applications are those based on LSTM neural networks (Afzaal et al., 2020; Ferreira and da Cunha, 2020; Yin et al., 2020). However, memory-based neural networks usually have accuracy problems as the forecasting time increases. To mitigate this middle-term memory leakage effect, the model developed in this work optimizes a hybrid architecture composed of an encoder-decoder architecture whose elementary cell includes a LSTM neural network and a DNN (Fig. 2). The hyperparameters of this hybrid architecture were optimized by the multiobjective genetic algorithm NSGA-II (Deb et al., 2002) and the identification of the forecasting model inputs were optimized by FL according to the methodology developed by (Lin et al., 1996).

2.3. Hybrid model architecture

Typically, LSTM neural networks as other RNNs can map an input sequence (time series) to an output sequence of the same dimension (fixed size vector). However, in many real applications such as water irrigation forecasting in IDs, inputs and outputs sequences are not necessary of the same length. For example, if the aim is to forecast irrigation water demand one week in advance, the input variables to the forecast model do not necessarily have the same temporal

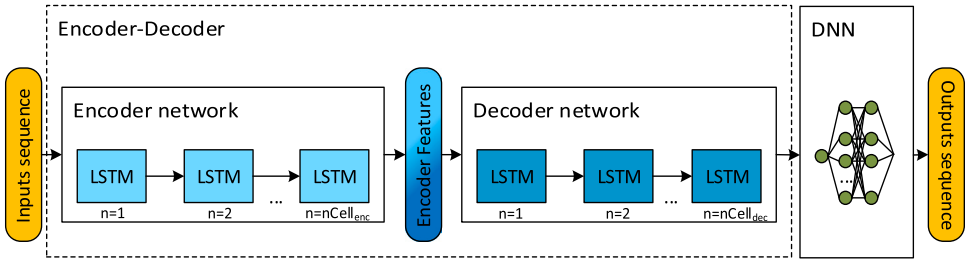


Fig. 2. Architecture of the hybrid model. LSTM: Long-short term memory model; DNN: Dense Neural Network model; $nCell_{enc}$: total number LSTM cells in the encoder network and $nCell_{dec}$: total number LSTM cells in the decoder network.

representativeness. Thus, precipitation could have a representative of two days backward, the amount of water applied a representative of four days previous, evapotranspiration of three days in advance, etc. This depends on the ID and the training set. Consequently, to generalize the methodology developed in this work, an LSTM encoder-decoder was used to address the fixed-length limitation of the LSTM neural networks. However, memory-based neural networks such as LSTMs have another major limitation related to a lack of memory and therefore a reduction in the accuracy of the estimations as the forecasting time increases. This effect is discussed in results section, but it is related to an effect of the chain rule. To reduce this major limitation of memory-based networks, a hybrid model composed of an LSTM encoder-decoder and a DNN was developed (Fig. 2). Thus, the LSTM encoder-decoder model was forced to learn only to extract the input features while the DNN model was responsible for interpreting these input features and providing the output (middle-term forecasting of irrigation water demand).

The entire hybrid model was developed with TensorFlow v2.9.0 (Abadi et al., 2015) and Keras v2.7.0 (Chollet, 2015) in python. Due to its high performance in most real-world applications, the optimization algorithm (training function) Adam (Adaptative momentum) was used for training the hybrid model (Kingma and Ba, 2015) with an exponential decay rate for the first and second moment of 0.9 and 0.999, respectively. Glorot uniform initialized algorithm (Glorot and Bengio, 2010) was used to initialize all synaptic weights because of its high performance (faster convergence) when Adams training function is used. The loss function used during the training process was the mean square error (mse). This loss function was used because of its ability to consider anomalous behavior of some specific days. Thus, the hybrid model will have the ability to learn not only from usual data but some typical anomalies in the ID (exceptional situations such as bank holidays, big rain events and so on…). Early stopping was used in this work as a form of regularization to avoid overfitting during training process.

2.3.1. LSTM encoder-decoder architecture

The LSTM encoder-decoder architecture is comprised of two models, the encoder to read and encode the input time series, and the decoder that reads the encoded input (*Encoder features*, Fig. 2) *t* and makes a one-step information extraction for each element in the output time series. As the forecasting was performed by the DNN model, this submodel (LSTM encoder-decoder) was only responsible of extracting the most relevant information from the input time series data. Thus, the responsibility of the LSTM encoder-decoder in the final forecast focused on, improving the main limitation of this kind of artificial neural networks such as lack of memory at certain times.

Both encoder and decoder are composed of elementary cells (Fig. 2). Each cell of both encoder and decoder is of the LSTM neural network type. The number of elementary cells establishes the density of the encode-decoder model. The number of cells of the encoder ($nCell_{enc}$) and the decoder ($nCell_{dec}$) is not the same necessarily. The higher the density of both encoder and decoder, the higher the capacity of adaptation and feature extraction. However, deeply dense encoder-decoder models often show problems in the learning process, both problems of memorization and training time and instability of the neural network. In LSTMHybrid, $nCell_{enc}$ and $nCell_{de}$ were optimized by the multi-objective genetic algorithm NSGA-II.

2.3.1.1. LSTM cell. Models based on Long-Short term memory neural networks are specially developed to solve time series forecasting issues. This kind of neural networks belong to the family of recurrent neural networks (RNNs) which adds memory to this architecture allowing to learn the ordered nature of time series as inputs. The state or memory occurs in its recurrent connections.

Fig. 3 shows the typical architecture of a LSTM cell. A LSTM cell consists of a set of gates which are recurrently connected blocks (memory blocks). Each LSTM cell is able to read, write and reset

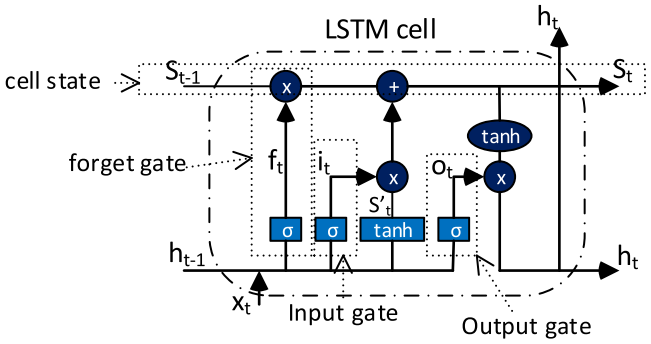


Fig. 3. General architecture of a LSTM cell. LSTM cell: Long-short term memory cell; S_{t-1}: cell state at time t-1; S_t: cell state at time t; ft: forget gate at time t: i_t: input gate at time t; o_t: output gate at time t; S'_t: candidate hidden state at time t; h_{t-1}: output from previous time step; h_t: output at time t and x_t input model at time t.

information along the training phase by its *input, output, forget* gates and the *cell state* (memory cell).

This kind of neural networks considers output from previous time steps (h_{t-1}) offering feedback to this LSTM cell. LSTM has a feedback loop at every node which allows information to move in both directions and so learning temporal patters of widely separated events. Additionally, *input, output and forget* gates decide which information is forwarded to the next node.

For a common LSTM cell architecture, the input gate (i_t), forget gate (f_t), cell state (S_t), and output gate (o_t) are described as follow:

The behavior of the input gate (i_t) can be characterized according to Eq. (1).

$$i_t = \sigma(x_t U^i + h_{t-1} W^i) \tag{1}$$

Where σ represent the sigmoid function; x_t is the input vector of the cell at time step t; U^i is the weight matrix that connects the inputs to the hidden layer; h_{t-1} is the hidden state from previous time step t-1 and W^i is the recurrent connection between the previous hidden layer and current hidden layer.

The forget gate, f_t, decides what to forget by a mechanism of sigmoid function according to Eq. (2).

$$f_t = \sigma(x_t U^f + h_{t-1} W^f) \tag{2}$$

Where U^f is the weight matrix that connects the inputs to the forget layer and W^f is the recurrent connection between the previous forget layer and the current forget layer.

The cell state (S_t) represents the "memory" of the LSTM cell (Eq. (3)) and information from the earlier time steps can travel to later time steps, reducing the effect of short-term memory.

$$S_t = \sigma(f_t S_{t-1} + i_t \widetilde{S}_t) \tag{3}$$

Where $\widetilde{S}_t$ is the candidate hidden state that is computed based on the current input and the previous hidden state (Eq. (4)) and S_{t-1} represents the internal memory at the time step t-1.

$$\widetilde{S}_t = \tanh(x_t U^s + h_{t-1} W^s) \tag{4}$$

Where tanh represents de tanh function; U^s is the weight matrix that connects the inputs to the candidate hidden layer and W^s is the recurrent connection between the previous candidate hidden layer and the current candidate hidden layer.

Finally, the output gate (o_t) which behavior is defined in Eqs. (5) and (6) defines the new cell state (S_t) and the hidden state at time step t.

$$o_t = \sigma(x_t U^o + h_{t-1} W^o) \tag{5}$$

Where U° is the weight matrix that connects the inputs to the output gate and W° is the recurrent connection between the previous candidate hidden layer and the current candidate hidden layer at output gate.

$$h_t = \tanh(S_t) \bullet o_t \quad (6)$$

The aim of the training process in LSTM cells and therefore in LSTM encoder-decoder model is to find the sets of U and W matrixes that minimize the global forecasting error. After finishing the training process, the forget gate can decide what is relevant to keep from prior steps. The input gate decides what information is relevant to add from the current step and the output gate determines what the next hidden state should be.

2.3.2. DNN architecture

DNN is a multilayer perceptron network (Rumelhart et al., 1986) responsible for the interpretation of each time step in the output time series provided by LSTM encoder-decoder and for giving the prediction values in its last layer. Fig. 4 shows a shallow DNN architecture which is mainly composed of three blocks: input layer, hidden layers and output layer. In LSTMHybrid, the output layer (and hence the DNN model) has been designed to forecast a single step in the output time series, not all days defined in the middle-term time scale. Consequently, the same DNN model has been used to process each time step provided by the LSTM encoder-decoder model. To achieve this, wrapped the DNN model in a time distributed wrapper that allows the wrapped layers to be used for each time step from the LSTM encoder-decoder model.

The block of hidden layers is frequently made up of two or more layers. Each layer is made up of several nodes or neurons, which are connected to the next layer's nodes. Similarly, to the LSTM encoder-decoder model, these connections are called synaptic weights and each layer has a synaptic weights matrix which connect the previous layer to the next one. In contrast to the LSTM encoder-decoder, all connections of the DNN model are fed forward, and therefore, the information is only allowed to move from a previous layer to the next one. During the training process all these synaptic weights are determined to minimize the global forecasting error. Each neuron of the hidden layers is activated by an activation function which defines the exited state of each neuron for each input vector. The activation function used in this work in all hidden layers is known as RELU (rectified linear unit activation function) (Eq. (7)), as it is one of the best activation functions in terms of performance (Isola et al., 2016; Karras et al., 2020). The number of neurons of the input layer is defined by the number of input variables. In this work, the number of neurons of the input layer was determined by the number of outputs of the decoder model.

$$output_{neuron} = \max(0, input_{neuron}) \quad (7)$$

Where $input_{neuron}$ and $output_{neuron}$ is the input and output values of each neuron of the hidden layers, respectively.

As with the LSTM encoder-decoder, both the number of the hidden layers and the number of neurons of each of them define the density of the DNN model and must be determined by each problem. These hyperparameters are often adjusted by trial and error, but this does not ensure the achievement of an optimal forecasting model. Thus, in this work, all this hyperparameters were optimized by the NSGA-II algorithm.

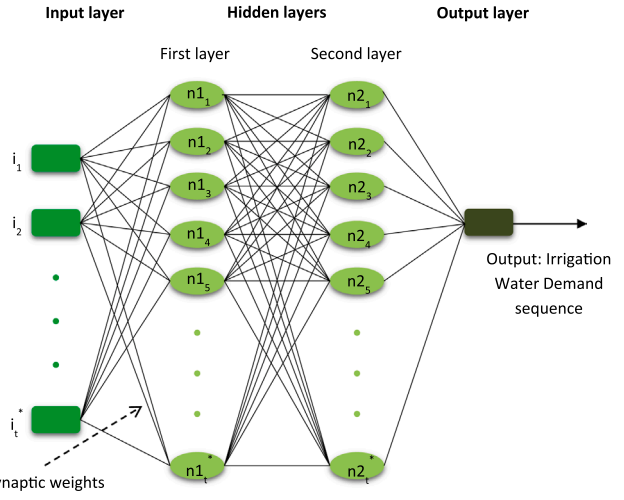


Fig. 4. Architecture of a DNN with two hidden layers. i_t: the total number of inputs of the DNN; $n1_t$ and $n2_t$: the total number of neurons of the first and second hidden layers, respectively.

2.4. Optimization of the hybrid model

The hybrid model proposed in this work was optimized by NSGA-II GA. Fig. 5 shows the flow chart of the optimization process of the hybrid model previously described. The optimization process starts by randomly generating a set of solutions. This initial set is called initial population and has a size of *iPop* solutions. Each of these solutions are known as individuals or chromosomes (*Ch*) of the initial population and are made up of a set of genes or decision variables (*numGEN*). In this work, each gene represents one hyperparameters of the hybrid model to be optimized.

Table 1 shows the genes considered in this work, as well as the value range and the data type that each variable can take during the optimization process. The first and second gene ($numGEN_1$ and $numGEN_2$) defined the number of LSTM cells of the encoder and decoder model of the LSTM encoder-decoder model, respectively. Thus, these two genes determined the density of the LSTM encoder-decoder model. The last three genes ($numGEN_3$, $numGEN_4$ and $numGEN_5$) determined the density of the DNN model defining the number of hidden layers as well as the number of neurons of theses layer. While $numGEN_4$ and $numGEN_5$ ranged from 0 to 200, $numGEN_3$ ranged from 1 to 200. Thus, the minimum and maximum number of hidden layers of the DNN model was 1 and 3, respectively.

After initialization of the initial population, each chromosome (each hybrid model with different architecture) was trained. After that, each of these trained models were tested and the objective function OF_1 and OF_2 were computed. In this work, OF_1 maximized the average of the R^2 (determination coefficient) of the output hybrid model (output time series of irrigation water demand). OF_2 minimized the average of the standard error prediction (SEP) (Ventura et al., 1995) Then, it was necessary to sort the initial population according to its aptitude, i.e., the forecast capacity of each hybrid model (objective functions). The best *iPop/2* chromosomes were then selected as the best individual of the initial population. In the remaining stages, the *iPop/2* chromosomes were modified (crossover and mutation), and a new population of *iPop* individuals was generated. The process was repeated for *numGEN* generations. Finally, the set of *iPop* optimal chromosomes (optimal hybrid models) obtained in the last generation defined the Pareto front.

In order to minimize the computational cost during the optimization process. All hybrid models are trained with restriction of training epochs. Once the optimization process is finished, the hybrid models that form the pareto front are re-trained without restrictions, obtaining the optimal values for all the synaptic weights of the gates of the LSTM encoder-decoder as well as the synaptic weights of the DNN model.

2.5. Identification of the hybrid model inputs

The optimal determination of the number of inputs to the hybrid model developed in this work is a key element. The input space reduction is essential both to achieve good generalization during the hybrid model production phase and to reduce the computational cost (memory and time) during the training phase. This dimension reduction must

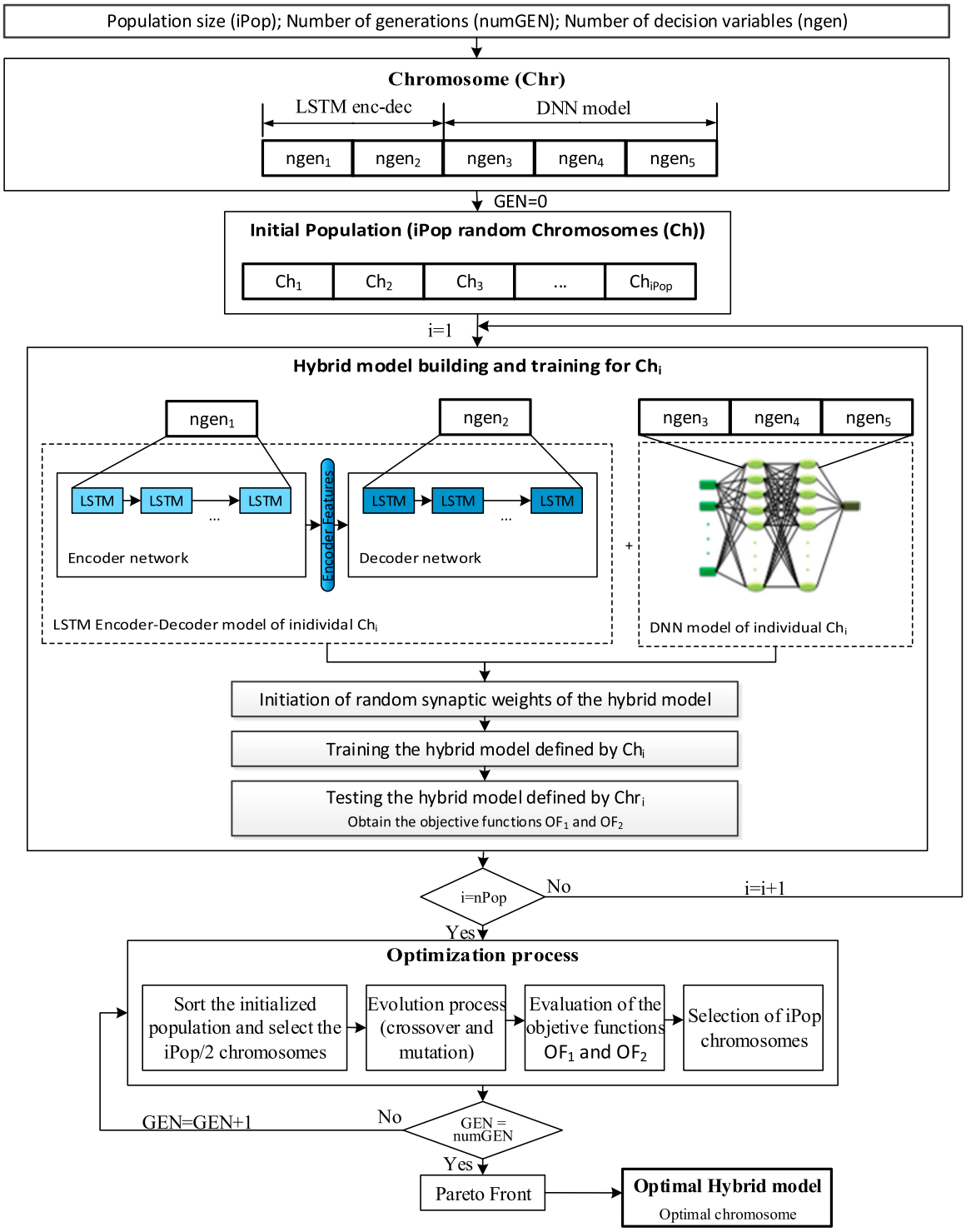


Fig. 5. Flowchart of the optimization process. *iPop*: size of the initial population; *numGEN*: total number of generations; *ngen*: number of gen or decision variables; $LSTM_{enc\text{-}dec}$: Long-short Term memory cells in the encoder and decoder model; *DNN model*: Dense Neural Network model; *GEN*: current generation; i: current individual or chromosome; OF_1: objective function 1 and OF_2: objective function 2.

identify the significant inputs from the set of possible inputs (potential inputs) for the forecasting hybrid model.

PCA (Principal Component Analysis) or PLS (partial least square) cardinal components methods are widely used to achieve this feature space reduction of the hybrid model. However, when the significant features resulting from these techniques are used in nonlinear models, very poor results are usually obtained (Lin et al., 1996). Therefore, in this work, fuzzy curves and fuzzy surfaces have been used to automatically identify the main features as inputs in the developed hybrid model by adapting the methodology developed by Lin et al. (1996). This methodology is described as follow:

For a potential input variable *v*, $PotI_v$, a fuzzy curve beam, cb_v, was created (Eq. (8)). Each curve determined the relationship between *v* and the output variable of the forecasting model (target variable), which in this work was the daily irrigation water demand (IWD) of the ID.

Table 1
Decision variables to be optimized during the optimization process of the hybrid model.

Decision variable	Range values	Type value	Description
$numGEN_1$	1–500	Integer	Number of LSTM cells of the encoder model.
$numGEN_2$	1–500	Integer	Number of LSTM cells of the decoder model.
$numGEN_3$	1–200	Integer	Number of neurons of the first hidden layer of the DNN model.
$numGEN_4$	0–200	Integer	Number of neurons of the second hidden layer of the DNN model.
$numGEN_5$	0–200	Integer	Number of neurons of the third hidden layer of the DNN model.

$$cb_v(PotI_v) = \frac{\sum_{k=1}^{M} \mathrm{IWD}_k \bullet \mu_{v,k}(PotI_v)}{\sum_{k=1}^{M} \mu_{v,k}(PotI_v)} \quad (8)$$

Where $\mu_{v,k}$ is the fuzzy membership function of point k in the space *v-IWD,* which relates the potential input variable v and the daily irrigation water demand of the entire ID and is defined by Eq. (9); M is the total number of points in the space $PotI_v$- IWD and IWD_k is the daily irrigation water demand of the entire ID in point k of the space $PotI_v$- *IWD*.

$$\mu_{v,k}(PotI_v) = \exp\left(-\left(\frac{PotI_{v,k} - PotI_v}{b}\right)^2\right) \quad (9)$$

Where $PotI_{v,k}$ is the value of $PotI_v$ at point k in the space PI_v-*IWD* and b takes a value close to two (Lin et al., 1996).

Hereafter, the mean square error values, *MSE*, of each cb_v curve were computed and sorted in ascending order according to Eq. (10). If there was a fully random relationship between the input variable v and the *IWD*, the fuzzy curves were flat, and the $MSEcb_v$ was large. Otherwise, $MSEcb_v$ took small values when the relationship in the space $PotI_v$-*IWD* was more significant.

$$\mathrm{MSE}_{cb_v} = \frac{1}{M} \sum_{k=1}^{M} (cb_v(PotI_{v,k}) - IWD_k)^2 \quad (10)$$

According to Lin et al. (1996a), a fuzzy surface is a space with a two-dimensional fuzzy curve. This fuzzy surface, $fs_{iv,ij}$, is defined in Eq. (11).

$$fs_{v,ij}(PotI_v, PotI_{ij}) = \frac{\sum_{k=1}^{M} IWD_k \bullet \mu_{v,k}(PotI_v) \bullet \mu_{ij,k}(PotI_{ij})}{\sum_{k=1}^{M} \mu_{v,k}(PotI_v) \bullet \mu_{ij,k}(PotI_{ij})} \quad (11)$$

Where $PotI_v$ and $PotI_{ij}$ are two potential input variables.

Then, like Eq. (10), the MSE was computed for each fuzzy surface, $\mathrm{MSE}_{fs_{v,ij}}$. Thus, fuzzy curves were initially used to rank all the potential input variables in ascending order. The PotI variable with the smallest $MSEcb_v$ was the most important input variable. Fuzzy surfaces were then used to find the independent input variables and to eliminate the related input in each step (20 % according to Lin et al., 1996). Therefore, in each step, new fuzzy surfaces were computed, and 20 % of the potential input variable with the largest $\mathrm{MSE}_{fs_{v,ij}}$ was eliminated.

Once the most important features were automatically identified, the final dataset composed of input variables- target was determined. Finally, the dataset was split into three subsets (training set, validation set and testing set) to achieve a good generalization of the hybrid model. This division of the original dataset was computed by a Monte Carlo algorithm maintaining the same mean and standard deviation in the three subsets as the original dataset (Ballesteros et al., 2016).

3. Results and discussion

The LSTMHybrid model was applied to CZID for the irrigation seasons from 2015 to 2022, with the aim of predicting the water demand at irrigation district level seven days in advance.

3.1. Inputs of the hybrid model

The significant inputs were determined by a combination of fuzzy curves and fuzzy surfaces according to Lin et al. (1996). The potential inputs evaluated and selected are shown in Table 2. A total of 18 potential inputs were assessed in a temporal frame which ranged from 1 to 10 days before the forecasting day, i.e. each of these 18 potential input variables were evaluated in a combination from 1 to 10 days (from *t-1* to *t-10*, in a step of 1 day, being *t* the forecasted day) previous to the forecasted day. After applying the methodology described in Section 2.5., only four input variables were finally selected: IWD, T_{ave}, Rad, ETo. The representativeness of all these input variables over the target variable (IWD) were maximum in a temporal frame of 7 days. Thus, the input space of the hybrid model consisted of four time series input variables. The time series ranged from t-1 to t-7, for the four input variables.

The hybrid model is designed to forecast the real behavior of the farmers within the ID but not the aggregation of the crop irrigation needs. The main difference (but not a small one) between the two approaches is the irrigation water management by each farmer. Logically, those farmers with better management of their irrigation systems show a smaller gap between the water applied and the actual crop needs. However, due to many other factors such as socio-cultural practices, farmers' perception based on their experience, or peculiarities of the irrigation system, there are farmers who are far from the best irrigation practices. According to the methodologies developed by González Perea et al. (2018, 2019) the wind chill temperature (determined mainly by temperature, wind speed and relative humidity) is essential to obtain a good estimation of the daily irrigation water demand in an irrigation district. This effect of the wind chill of each farmer seems to be blurred when aggregated to a larger spatial scale such as the entire irrigation district. This was highlighted by González Perea et al. (2015) who developed a forecasting system of the daily water demand at the irrigation district level. In that model, the farmer's wind chill did not

Table 2
Potential and selected inputs for the hybrid model.

Potential Inputs	Units	Description[(1)]
T_{max}	°C	Maximum temperature
T_{min}	°C	Minimum temperature
Tave[+]	°C	Average temperature
RH_{max}	%	Maximum relative humidity
RH_{min}	%	Minimum relative humidity
RH_{ave}	%	Average relative humidity
Rad[+]	MJ m^{-2}	Solar radiation
WS_{max}	m s^{-1}	Maximum wind speed
WS_{ave}	m s^{-1}	Average wind speed
Prec	mm	Rainfall
Eprec	mm	Effective rainfall computed according to Penman Monteith methodology
$Prec_{bin}$	-	Boolean variable that defines the occurrence or non-occurrence of precipitation. 0: no rainfall; 1: rainfall
Radnet	MJ m^{-2}	Net solar radiation
ETo[+]	mm	Reference evapotranspiration
IWD[+]	m^3	Irrigation water demand of the irrigation district
Holy	-	Boolean holidays. This variable is equal to 1 for bank holidays and vacation days and 0 otherwise.
Wday	-	Weekday
DOY	-	Day of the year

(1) All potential inputs were considered in a time frame from 1 to 10 days before the forecasting day.
[+]Selected inputs.

appear so clearly in the main input variables of the daily forecasting model (time series with a time frame of 2 days of irrigation water demand and reference evapotranspiration and average temperature and solar radiation for the day to predict).

Table 2 (selected inputs) shows that this tendency of dilution of the effect of the farmer's wind chill when the spatial scale increases is maintained what coincides with previous works' findings. Moreover, as the forecasting period increases, the previous days of the input variables considered are also longer. In this work, this time frame for the input variables was of seven days.

3.2. Optimization process

The density, which influences the learning and adaptation capacity of the hybrid model (both LSTM encoder-decoder and DNN), was optimized by the NSGA-II. The initial population consisted of 50 chromosomes(*iPop* = 50). The value of *numGEN* (number of generations) is determined by the stabilization of the two objective functions. This stabilization was reach at generation number 100 (*numGEN* = 100). The probabilities for mutation and crossover were set to 10 % and 90 %, respectively.

Fig. 6 shows the Pareto front which was obtained in the last generation of the optimization process as well as the initial population. Both populations (initial and Pareto front) were characterized by their objective function, OF_1 and OF_2. which were always computed over the test set. The aim of the optimization process carried out by the NSGA-II was the maximization of OF_1 and the minimization of OF_2. In this type of optimization problems where the two objective functions are opposed, it is usual to find an initial population located in the upper left corner (Fig. 6). However, when the initial population at generation 0 was randomly generated, the values of their objective functions were closer to the pareto front.

The LSTMHybrid model was based on the LSTM encoder-decoder which extracts the characteristics of the inputs and the DNN model which interpretates these characteristics to provide an estimation. The density of both submodels, so their capacity to learn, were optimized. However, other essential hyperparameters of the hybrid model such as the training function (Adam), activation function (RELU) and loss function (MSE) were previously fixed. These hyperparameters were previously set (according to previous research works' findings) to solve the problem addressed in this work (memory-based time series forecasting). Consequently, a certain part of the hybrid model was already optimized generating an initial population located in the region of the OF_1-OF_2 space showed in Fig. 6. This increased the performance of the genetic algorithm, reducing both the size of the initial population and the number of generations.

For the initial population, the values of OF_1 and OF_2 ranged from 67.69 % to 93.05 % and from 23.23 % to 52.69 %, respectively. The variation of both objective function values in this population was 4.97 % and 6.23 %, respectively. After optimization process, the population of the last generation (Pareto front) was in the right lower corner of the OF_1-OF_2 space (Fig. 6), usual behavior of a two-opposite objective functions Pareto front. The values of OF_1 and OF_2 for the Pareto front ranged from 93.39 % to 94.15 % and from 20.49 % to 25.29 %, respectively. However, the variance of this population was 0.17 % for OF_1 and 1.15 % for OF_2. Although the initial population was close to the Pareto front, the optimization process involved a smaller dispersion (variance) of the Pareto front compared to that of the initial population, also improving the values of the objective functions. In addition, when the input variables of a forecasting model are temporally correlated (as in this work where even the variable to be forecasted also appears as input variable of the hybrid model), its R^2 value is usually high. However, this effect does not appear in the estimation's accuracy, providing models that are not very accurate if they are not well optimized, although with a relatively high R^2 value. Thus, the Pareto front improved in the OF_2 direction much more than in the OF_1 direction, which means that the improvement in the accuracy was higher than in representativeness. In other words, as inputs were correlated because of their temporal correlation, the optimization of the density made it possible to find much more accurate hybrid models (although it also improved their representativeness).

Fig. 7 shows the histogram for the number of LSTM cells in the LSTM encoder-decoder submodel for the initial population (Fig. 7a) and the Pareto front (Fig. 7b). The number of LSTM cells of encoder and decoder is controlled by the gen $numGEN_1$ and $numGEN_2$, whose values could range from 1 to 500 (Table 1). As the initial population was randomly

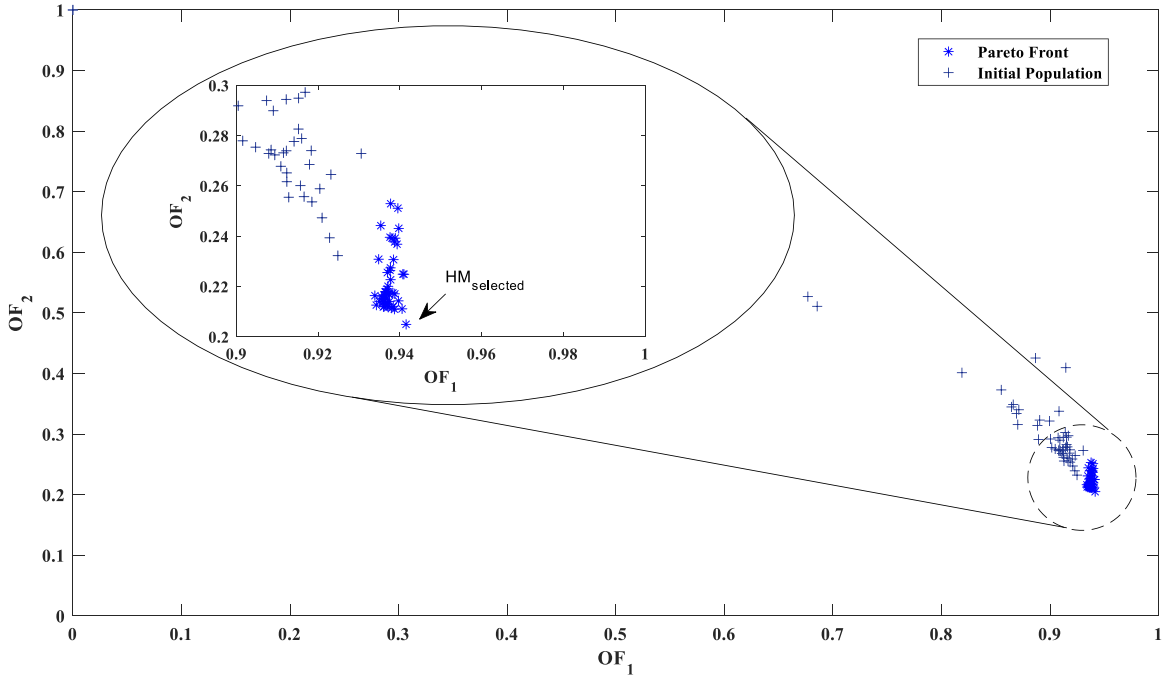


Fig. 6. Pareto front and initial population of the optimization process for the test dataset. $HM_{selected}$: hybrid model selected; OF_1: objective function 1 and OF_2: objective function 2.

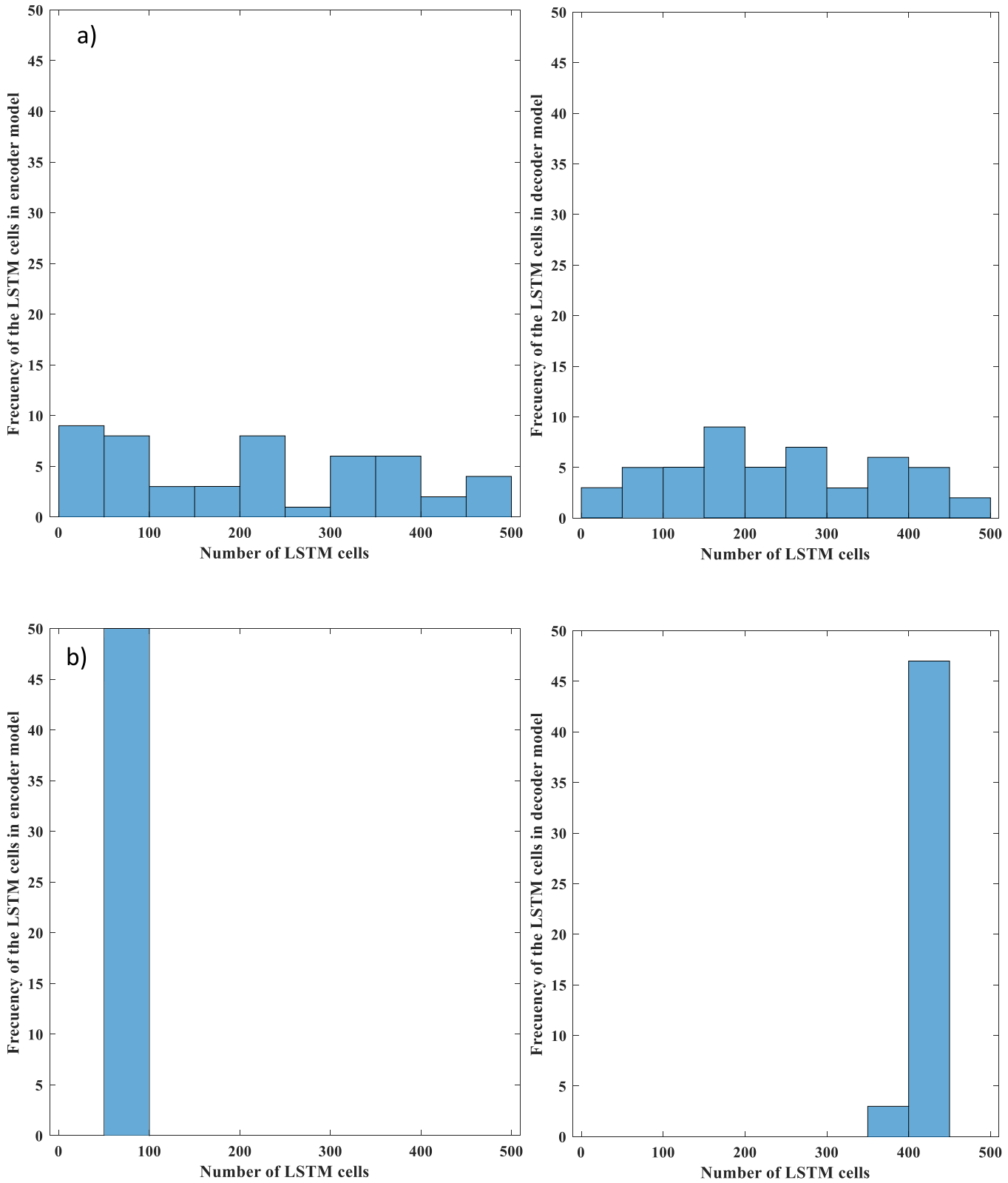


Fig. 7. Histogram for the number of LSTM cells in the encoder and decoder of the LSTM encoder-decoder submodel for the initial population (a) and the Pareto front (b).

created using a uniform distribution so that all values had the same probability of occurrence, the frequencies of the number of the LSTM cells were similar for encoder and decoder (Fig. 7a). After optimization process, the optimal LSTM cells in both encoder and decoder models were found (Fig. 7b). The number of LSTM cells of the encoder model for all individuals was 50 cells. Hence, all individuals with more or fewer LSTM cells than 50 were eliminated during the optimization process. With a density of 50 LSTM cells, for the irrigation district studied in this work, it is enough to encode all its input variables with losing relevant information. Although this methodology can be applied to other

irrigation districts and other input variables, the density of the encode model will be different and specific for each irrigation district. The density of the decoder model ranged from 400 to 450 LSTM cells in 45 individuals and from 350 to 400 LSTM cell in 5 individuals. However, all individuals of the Pareto front had a density in the decoder model which ranged from 391 to 410 LSTM cells. Therefore, the density of the decoder model is deeper that the encoder model. According to these results of the optimization process, while encoder model needed only 50 LSTM cells to encode the input information, decoder model required at least 391 LSTM cells to decode this information and pass it to the

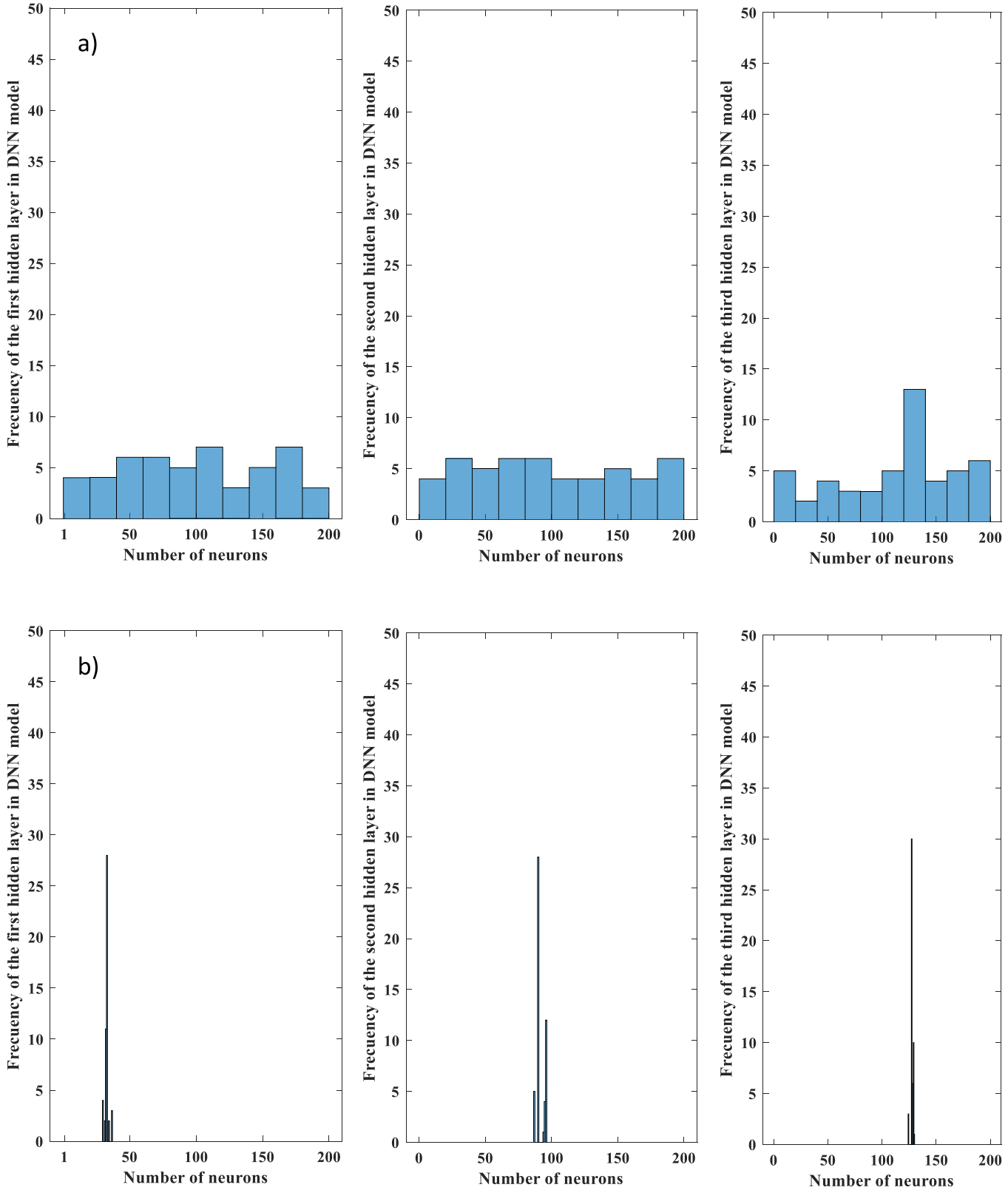


Fig. 8. Histogram for the number of neurons of the first, second and third hidden layer in the DNN submodel for the initial population (a) and the Pareto front (b).

interpreter (DNN submodel). This large difference in densities was due to the DNN submodel, which needed the information as atomized as possible to make an accurate estimation.

Similarly to the density of the LSTM encoder-decoder submodel, Fig. 8 shows the histogram for the number of neurons of the first, second and third hidden layers of the DNN model for both initial population (Fig. 8a) and Pareto front (Fig. 8b). The number of neurons of the first, second and third hidden layers were controlled during the optimization process by the genes $numGEN_3$, $numGEN_4$ and $numGEN_5$, respectively. While the number of the first hidden layer ranged from 1 to 200, the neurons of the second and third hidden layer ranged from 0 to 200, i.e. the minimum number of the hidden layers of the DNN submodel was 1, and maximum number of hidden layers in this submodel was 3. As the initialization of the first two genes, $numGEN_3$, $numGEN_4$ and $numGEN_5$ were randomly initialized with a uniform distribution. Thus, the density of the DNN submodel in the initial population was uniformly distributed between all their individuals (Fig. 8a). After optimization process, all individuals needed three hidden layers (no individual with 0 neurons in genes $numGEN_4$ and $numGEN_5$ were found) to optimize the estimations of irrigation water demand for 7 days ahead (Fig. 8b). No hidden layers in the individuals of the Pareto front showed neuron numbers close to the established lower and upper limits (minimum 0 or 1 neuron and maximum 200 neurons). If any of these distributions were close to the lower or upper limit, it could mean that the range of neurons of some hidden layer was not well selected. This would have required increasing this range to complete its optimization process properly. The number of neurons of the three hidden layers were gradually increased. Thus, all individuals showed a density in the first hidden layer that ranged from 30 to 37 neurons, getting 28 individuals with 33 neurons in this hidden layer. The density of the second hidden layer was increased in all individuals of the Pareto front ranged their neurons from 87 to 96. A total of 28 individuals of the Pareto front had a density of 90 neurons in its second hidden layer, although the density variance of this layer was higher than the other two hidden layers. The number of neurons in the third hidden layer ranged from 124 to 130. 30 individuals of the Pareto front had 127 neurons in this hidden layer.

According to results shown in Figs. 7 and 8, the LSTMHybrid model was made up of 50 LSTM cells in encoder model, around 400 LSTM cells in decoder model, three hidden layers in the DNN submodel with a range of neurons of 30–37, 87–96 and 124–130, respectively. The optimal hybrid model forecasted a time series of irrigation water demand at irrigation district spatial scale and middle term temporal scale (7 days ahead). This is a complex task, considering the uncertainty associated with the decisions that each farmer makes every day on his farm and the aggregation of all the farmers' behavior at ID level. Consequently, all the optimal hybrid models (those included in the Pareto front) found by the genetic algorithm were rather dense with millions of parameters in each of them. This fact gives an idea of the complexity of the problem to be solved, especially in comparison with the models shown in the introduction which had only several thousands of parameters.

3.3. Optimal hybrid model

After the optimization process, the optimal hybrid model was selected according to its objective function values ($HM_{selected}$). Thus, Fig. 6 shows the hybrid model with the higher OF_1 value (0.9415) and lower OF_2 value (0.2049) and so selected as the best optimal hybrid model. Table 3 shows the values of the decision variables ($numGEN_1$, …, $numGEN_5$) for $HM_{selected}$ as well as R^2 and SEP values. $HM_{selected}$ had 50 and 409 LSTM cells in the encoder and decoder models, respectively, of the LSTM encoder-decoder submodel. Three hidden layers of 31, 96 and 128 neurons were implemented in the DNN submodel. Hence, $HM_{selected}$ had a total of 1.5 million of parameters. With this density, the $HM_{selected}$ was able to forecast the daily irrigation water demand at a real ID, for 7 days ahead with an average representativeness value in the test dataset of 94.15 % and an accuracy of 20.49 %. To achieve these mean forecast values, the $HM_{selected}$ required four time series of input variables. These time series were composed of data from seven days, prior to the first forecast day of T_{ave}, Rad, ETo and IWD.

Table 3
Values of the decision variables for the optimal hybrid model as well as R^2 and SEP values of the test set.

	$HM_{selected}$
Number of LSTM cells of the encoder model	50
Number of LSTM cells of the decoder model	409
Number of neurons of the first hidden layer	31
Number of neurons of the second hidden layer	96
Number of neurons of the third hidden layer	128
R^2	94.15
SEP (%)	20.49

The average values of representativeness and accuracy of the HMselected were enough for a proper daily management of the irrigation district. Moreover, the resolution of the hybrid model allowed the almost real-time management of the energy demand of the water distribution network. When these results were disaggregated by forecast days, the R^2 and SEP values changed. R^2 decreased while SEP increased when the time horizon was higher. Fig. 9 shows the scatterplots between the observed (real IWD) and estimated IWD for seven days ahead (t-1, …, t-7) of the $HM_{selected}$ applied over the test dataset. Although the average of the R^2 value of $HM_{selected}$ was 94.15 %, the representativeness of the $HM_{selected}$ varied considerably as the forecast time increased. The R^2 value ranged from 97.06 % for *t-1* to 85.49 % for *t-7*. Similarly, the accuracy of the $HM_{selected}$ decreased as forecast time increased. The SEP value of the $HM_{selected}$ ranged from 16.42 % for *t-1* to 22.31 % to *t-7*. Thus, the $HM_{selected}$ forecast capability diminished when the forecast time increased.

It should be noted that the hybrid model developed in this work is based on memory. Although this type of model architecture is optimal for time series forecast, a very large number of LSTM cells are chained in both encoder and decoder models. The training process of the hybrid model used backpropagation method to update the neural networks weights in the direction in which the error function decreases most rapidly, the negative of the gradient. Therefore, the correction of all weights of the hybrid model (both LSTM encoder-decoder submodel and DNN submodel) is backward. When this backpropagation method is applied to architectures with memory, a memory leakage effect appears in the initial LSTM cells. This effect, caused as a vanishing of the error gradient and the chain rule, makes the hybrid model more insensitive to the inputs of the time series further back in time (the first to enter in the hybrid model), causing a loss of memory. In this work, the hybrid architecture attempted to reduce this effect by limiting the workload of the part of the hybrid model based on memory (LSTM encoder-decoder). The full workload was divided into the LSTM encoder-decoder and the DNN submodels. However, this 'Alzheimer's' effect of the neural networks based on memory is still present, albeit with a lower impact. Future work on this approach should look at new artificial neural network architectures currently under development, such as those based on attention (based on Transformer models) rather than memory, which eliminate this problem completely. This will improve the accuracy and representativeness of newly developed models.

4. Conclusions

Several works for short-term forecast of daily irrigation water demand at two spatial resolutions (farm and irrigation district levels) were previously developed. However, although some of them offered very accurate forecasts, the temporal resolution is not wide enough to properly manage an ID. In this work a novel methodology to develop a hybrid model based on memory to forecast the daily irrigation water demand at middle term time scale (7 days ahead) was presented. The hybrid model was optimized by the genetic algorithm NSGA-II to

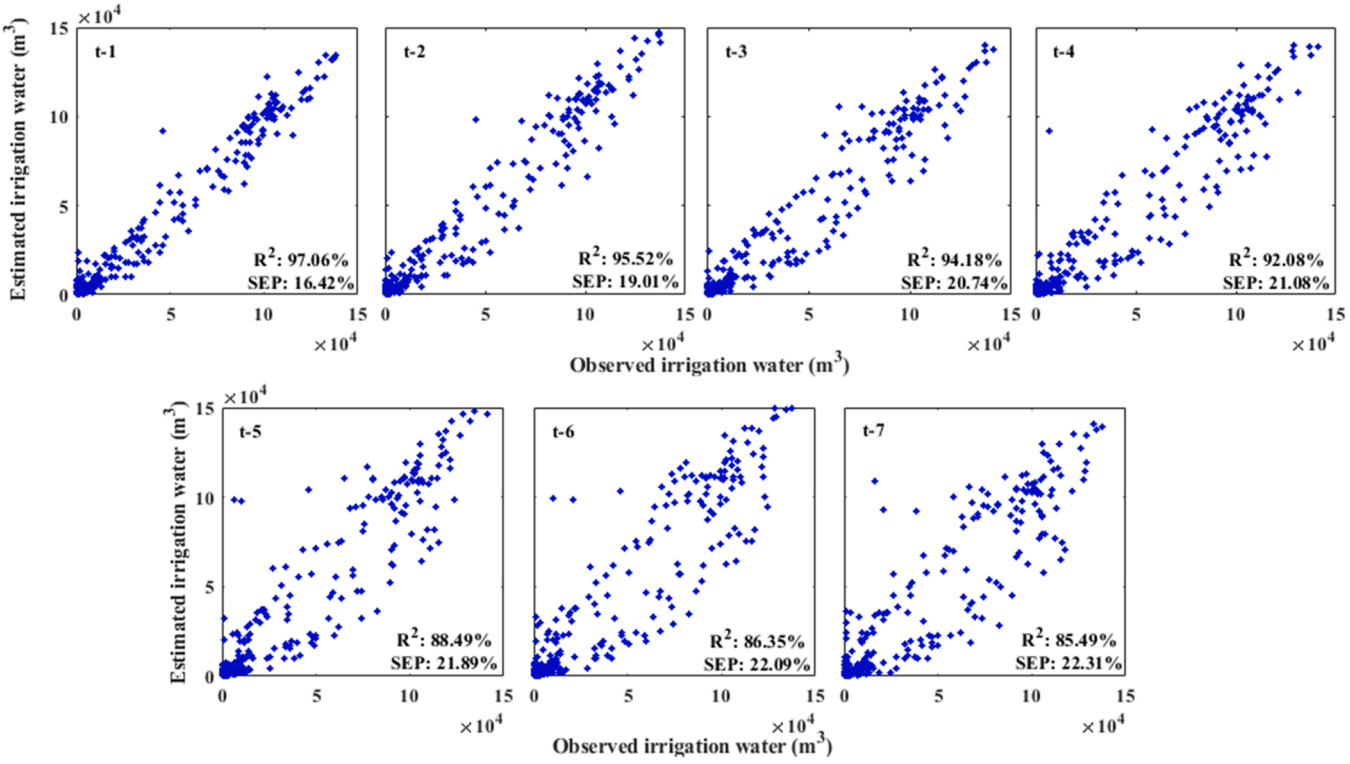


Fig. 9. Scatterplots between observed and estimated irrigation water demand for seven days ahead (t-1, …, t-7) by the optimal hybrid model for the test dataset.

maximize and minimize the representativeness and accuracy of their estimations applied to a real ID. The model inputs were also optimized by fuzzy curves and fuzzy surfaces, obtaining four input variables (T_{ave}, *Rad, ETo* and *IWD)* as times series in a frame of seven days for the developed hybrid model.

Results showed that the irrigation water demand forecast for seven days in advance in an irrigation district is a complex task, requiring a dense hybrid model. The hybrid model selected after the optimization process had more than 1.5 million of parameters. This hybrid model selected as optimum consisted of 50 LSTM cells and 409 LSTM cells in the LSTM encoder-decoder submodels and three hidden layers with 31, 96, 128 neurons, respectively, in the DNN submodel. The average representativeness of the model was greater than 94 % with an average accuracy in their estimations of around 20 %.

To date, the forecast model developed in this work is the most accurate at the spatio-temporal scale addressed. However, this hybrid model is memory-based and often shows memory leakage effects as the forecast time increases. Although an attempt was made to minimize this memory leakage effect by reducing the workload of the memory part of the forecast model (LSTM encoder-decoder submodel) and expanding with a DNN submodel, this 'Alzheimer' effect could be observed in the results obtained. Nevertheless, thanks to the hybrid model, this memory leakage effect is minimum. This model can be very useful for managers since it offers accurate predictions of water demand, which plays an essential role for water management and energy contracting in irrigation districts considering current and expected scenarios of water scarcity and high energy prices.

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

The authors do not have permission to share data.

Acknowledgments

This research was funded by Project PID2020-115998RB-C21 of the Spanish Ministry of Science and Innovation. We acknowledge financial support of the "Postdoctoral Researchers" program of the Junta-de-Andalucía Government and the Spanish Ministry of Science and Innovation, the Spanish State Research Agency, through the Severo Ochoa and María de Maeztu Program for Centers and Units of Excellence in R&D (Ref. CEX2019-000968-M).

References

Abadi, M., Agarwal, A., Barham, P., 2015. TensorFlow: large-scale machine learning on heterogeneous distributed systems. Netw.: Comput. Neural Syst. 1–19.

Afzaal, H., Farooque, A.A., Abbas, F., Acharya, B., Esau, T., 2020. Computation of evapotranspiration with artificial intelligence for precision water resource management. Appl. Sci. 10. https://doi.org/10.3390/APP10051621.

Ballesteros, R., Ortega, J.F., Moreno, M.Á., 2016. FORETo: new software for reference evapotranspiration forecasting. J. Arid Environ. 124, 128–141. https://doi.org/10.1016/j.jaridenv.2015.08.006.

Chollet, F., 2015. Keras Api documentation.

Daccache, A., Ciurana, J.S., Rodriguez Diaz, J.A., Knox, J.W., 2014a. Water and energy footprint of irrigated agriculture in the Mediterranean region. Environ. Res. Lett. 9. https://doi.org/10.1088/1748-9326/9/12/124014.

Daccache, A., Knox, J.W., Weatherhead, E.K., Daneshkhah, A., Hess, T.M., 2014b. Implementing precision irrigation in a humid climate – recent experiences and on-going challenges. Agric. Water Manag. 147, 135–143.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evolut. Comput. 6, 182–197. https://doi.org/10.1109/4235.996017.

Derardja, B., Lamaddalena, N., Fratino, U., 2019. Perturbation indicators for on-demand pressurized irrigation systems. Water 11. https://doi.org/10.3390/w11030558.

FAO, 2015. Climate change and food security: risks and responses.

FAO, 2016. FAO's work on climate change.

Fernández García, I., Montesinos, P., Camacho Poyato, E., Rodríguez Díaz, J.A., 2016. Incorporating the irrigation demand simultaneity in the optimal operation of pressurized networks with several water supply points. Water Resour. Manag. 30, 1085–1099. https://doi.org/10.1007/s11269-015-1212-7.

Ferreira, L.B., da Cunha, F.F., 2020. Multi-step ahead forecasting of daily reference evapotranspiration using deep learning. Comput. Electron. Agric. 178. https://doi.org/10.1016/j.compag.2020.105728.
Forouhar, L., Wu, W., Wang, Q.J., Hakala, K., 2022. A hybrid framework for short-term irrigation demand forecasting. Agric. Water Manag. 273. https://doi.org/10.1016/J.AGWAT.2022.107861.
Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks.
Gonzalez Perea, R., Ballesteros, R., Ortega, J.F., Moreno, M.Á., 2021. Water and energy demand forecasting in large-scale water distribution networks for irrigation using open data and machine learning algorithms. Comput. Electron. Agric. 188. https://doi.org/10.1016/j.compag.2021.106327.
González Perea, R., Poyato, E.C., Montesinos, P., Díaz, J.a.R., 2015. Irrigation demand forecasting using artificial neuro-genetic networks. Water Resour. Manag. 29, 5551–5567. https://doi.org/10.1007/s11269-015-1134-4.
González Perea, R., Camacho Poyato, E., Montesinos, P., Rodríguez Díaz, J.A., 2016. Optimization of irrigation scheduling using soil water balance and genetic algorithms. Water Resour. Manag. 30, 2815–2830. https://doi.org/10.1007/s11269-016-1325-7.
González Perea, R., Camacho Poyato, E., Montesinos, P., Rodríguez Díaz, J.A., 2018. Prediction of applied irrigation depths at farm level using artificial intelligence techniques. Agric. Water Manag. 206, 229–240. https://doi.org/10.1016/j.agwat.2018.05.019.
González Perea, R., Camacho Poyato, E., Montesinos, P., Rodríguez Díaz, J.A., 2019. Prediction of irrigation event occurrence at farm level using optimal decision trees. Comput. Electron. Agric. 157, 173–180. https://doi.org/10.1016/j.compag.2018.12.043.
González Perea, R., Poyato Camacho, E., Rodríguez Díaz, J.A., 2021. Forecasting of applied irrigation depths at farm level for energy tariff periods using coactive neuro-genetic fuzzy system. Agric. Water Manag. 256. https://doi.org/10.1016/j.agwat.2021.107068.
Graves, A., Schmidhuber, J., 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw. 18, 602–610. https://doi.org/10.1016/J.NEUNET.2005.06.042.
Hedley, C.B., Knox, J.W., Raine, S.R., Smith, R., 2014. Water: advanced irrigation technologies. Encycl. Agric. Food Syst. 5, 378–406. https://doi.org/10.1016/B978-0-444-52512-3.00087-5.
Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A., 2016. Image-to-Image Translation with Conditional Adversarial Networks.
Kamilaris, A., Prenafeta-Boldú, F.X., 2018. Deep learning in agriculture: a survey. Comput. Electron. Agric. 147, 70–90. https://doi.org/10.1016/J.COMPAG.2018.02.016.
Kingma, D.P., Ba, J.L., 2015. Adam: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, pp. 1–15.
Lin, Y., Cunningham III, G.A., Coggeshall, S. v, 1996. Input variable identification—fuzzy curves and fuzzy surfaces. Fuzzy Sets Syst. 82, 65–71.
Playán, E., Mateos, L., 2006. Modernization and optimization of irrigation systems to increase water productivity. Agric. Water Manag. 80, 100–116.
Plusquellec, H., 2009. Modernization of large-scale irrigation systems: is it an achievable objective or a lost cause. Irrig. Drain. 58, S104–S120.
Pulido-Calvo, I., Gutiérrez-Estrada, J.C., 2009. Improved irrigation water demand forecasting using a soft-computing hybrid model. Biosyst. Eng. 102, 202–218. https://doi.org/10.1016/j.biosystemseng.2008.09.032.
Rodríguez Díaz, J.A., Weatherhead, E.K., Knox, J.W., Camacho, E., 2007. Climate change impacts on irrigation water requirements in the Guadalquivir river basin in Spain. Reg. Environ. Change 7, 149–159. https://doi.org/10.1007/s10113-007-0035-3.
Rodríguez Díaz, J.A., Montesinos, P., Camacho Poyato, E., 2012. Detecting critical points in on-demand irrigation pressurized networks – a new methodology. Water Resour. Manag. 26, 1693–1713.
Rodríguez-Díaz, J.A., Pérez, L., Camacho, E., Montesinos, P., 2011. The paradox of irrigation scheme modernization: more efficient water use linked to higher energy demand. Span. J. Agric. Res. 9, 1000–1008.
Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature.
Sheikh Khozani, Z., Barzegari Banadkooki, F., Ehteram, M., Najah Ahmed, A., El-Shafie, A., 2022. Combining autoregressive integrated moving average with long short-term memory neural network and optimisation algorithms for predicting ground water level. J. Clean. Prod. 348. https://doi.org/10.1016/J.JCLEPRO.2022.131224.
Tarjuelo, J.M., Rodriguez-Diaz, J.A., Abadía, R., Camacho, E., Rocamora, C., Moreno, M.A., 2015. Efficient water and energy use in irrigation modernization: lessons from Spanish case studies. Agric. Water Manag. 162, 67–77. https://doi.org/10.1016/j.agwat.2015.08.009.
Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T., 2020. Analyzing and Improving the Image Quality of StyleGAN.
Van, Rossum, Drake, G., Fred, L., 2009. Python 3 Reference Manual. CreateSpace, Scotts Valley, CA.
Ventura, S., Silva, M., Pérez-Bendito, D., Hervás, C., 1995. Artificial neural networks for estimation of kinetic analytical parameters. Anal. Chem. 67, 1521–1525.
Wolfert, S., Ge, L., Verdouw, C., Bogaardt, M.J., 2017. Big data in smart farming – a review. Agric. Syst. 153, 69–80. https://doi.org/10.1016/J.AGSY.2017.01.023.
Yin, J., Deng, Z., Ines, A.V.M., Wu, J., Rasu, E., 2020. Forecast of short-term daily reference evapotranspiration under limited meteorological variables using a hybrid bi-directional long short-term memory model (Bi-LSTM). Agric. Water Manag. 242. https://doi.org/10.1016/j.agwat.2020.106386.
Zhang, J., Zhu, Y., Zhang, X., Ye, M., Yang, J., 2018. Developing a Long Short-Term Memory (LSTM) based model for predicting water table depth in agricultural areas. J. Hydrol. 561, 918–929. https://doi.org/10.1016/J.JHYDROL.2018.04.065.