Renewable Energy 145 (2020) 2178–2191

Contents lists available at ScienceDirect

Renewable Energy

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

Assessment of Deep Learning techniques for Prognosis of solar thermal systems

Camila Correa-Jullian [a, *], José Miguel Cardemil [a], Enrique López Droguett [a], Masoud Behzad [b, a]

[a] Mechanical Engineering Department, Universidad de Chile, Beauchef 851, Santiago, Chile
[b] Industrial Engineering School, Faculty of Engineering, Universidad de Valparaíso, Brasil 1786, Valparaíso, Chile

ARTICLE INFO

Article history:
Received 13 May 2019
Received in revised form
12 July 2019
Accepted 23 July 2019
Available online 27 July 2019

Keywords:
Deep learning
Performance prediction
Solar thermal systems

ABSTRACT

Solar Hot Water (SHW) systems are a sustainable and renewable alternative for domestic and low-temperature industrial applications. As solar energy is a variable resource, performance prediction methods are useful tools to increase the overall availability and effective use of these systems. Recently, data-driven techniques have been successfully used for Prognosis and Health Management applications. In the present work, Deep Learning models are trained to predict the performance of an SHW system under different meteorological conditions. Techniques such as artificial neural networks (ANN) recurrent neural networks (RNN) and long short-term memory (LSTM) are explored. A physical simulation model is developed in TRNSYS software to generate large quantities of synthetic operational data in nominal conditions. Although similar results are achieved with the tested architectures, both RNN and LSTM outperform ANN when replicating the data's temporal behavior; all of which outperform naïve predictors and other regression models such as Bayesian Ridge, Gaussian Process and Linear Regression. LSTM models achieved a low Mean Absolute Error of 0.55 °C and the lowest Root Mean Square Error scores (1.27 °C) for temperature sequence predictions, as well as the lowest variance (0.520 $°C^2$) and relative prediction errors (3.45%) for single value predictions, indicating a more reliable prediction performance.

1. Introduction

The use of solar thermal technologies for heating and cooling applications offers a significant development potential worldwide and can play an important role in energy planning, securing energy supply and fostering sustainable economic development. In particular, low-temperature hot water systems are considered the most mature of these technologies, as large scale projects have been developed since the 1960s [1]. In this regard, Solar Hot Water (SHW) systems can be used to directly supply hot water for residential or commercial needs. To meet low-temperature demands, non-concentrating flat plate (FPC) or evacuated tube collectors (ETC) are the most common solutions. SHW systems are frequently coupled to a thermal storage system, allowing to match solar radiation availability with useful energy production, as demand profiles are commonly higher during non-insolation hours. SHW systems are a commercially proven, efficient alternative for water heating applications, reducing fossil fuel energy consumption and their emissions [2]. Still, they can face performance issues when not correctly designed or appropriately integrated. For instance, SHW systems are usually combined with auxiliary heating components, such as gas or electrical boilers, which allows delivering the thermal load in cases of unavailability of solar radiation or depletion of thermal storage. However, the integration of such systems creates an additional difficulty to properly identify and locate faults in a particular component. Therefore, faults in SHW systems may go unnoticed until the components are too degraded for simple maintenance procedures, elevating the overall costs and complexity of the installation. Thus, inspection, continuous monitoring, and fault detection are of great importance for a good long-term performance of solar thermal systems [3].

Prognosis and Health Management (PHM) is an important issue in modern engineering systems, in which models are designed and used to detect anomalies, diagnose faults and predict future states of the system, for instance, the Remaining Useful Life (RUL) of a

* Corresponding author.
E-mail address: camila.correa.j@ug.uchile.cl (C. Correa-Jullian).

https://doi.org/10.1016/j.renene.2019.07.100
0960-1481/

component before failure, given the current operational conditions. In turn, these models can help to increase the system's availability and performance by managing maintenance issues [4]. In the past decades, several approaches have been proposed for this purpose, coupled with the development of Machine Learning (ML) applications in engineering systems to assist decision-making analysis. ML techniques introduced data-driven models which rely less on the physical analysis of the system, favoring data collection and processing instead [5]. A ML algorithm can learn patterns from the available data, extracting abstract relationships within the studied variables to classify or predict future values. In this context, Deep Learning (DL) approaches have become a widely-used tool in several applications of ML tasks, such as object detection, speech recognition and replication [6]. These architectures have matured to approach highly complex and non-linear tasks. These have also been combined in different applications, such as natural language processing and estimation of RUL in mechanical components [7,8]. DL frameworks have been used to model and predict different physical phenomena, such as weather patterns, health management, and medical diagnosis, among others [5,9,10], showing outstanding results. Notable examples of rotary mechanical components and lithium-ion batteries have been analyzed through measurable variables such as vibrations, acoustic emissions, temperature, state of charge and state of health, among others [11–13]. The success of deep architectures resides in their ability to obtain higher levels of abstraction from the input data. This property enhances its feature extraction capabilities and, due to its hierarchical architecture, it also helps the model to converge faster. The growing number of successful cases of these techniques' applications for risk and reliability assessment of mechanical systems may offer an additional development path for reliability assessment in thermal systems.

Machine Learning and DL algorithms have been applied to describe and predict the performance of thermal systems and solar energy applications. For instance, Artificial Neural Networks (ANN) have been applied to analyze thermal systems, particularly heat pumps, solar thermal and refrigeration technologies showing excellent results for performance prediction and detection of anomalies during operation of thermal systems [14]. Ghritlahre and Prasad [15] presented a literature review related to performance predictions of solar collector systems, describing over 30 different representations which assess ANN models for performance prediction for different configurations of thermal systems. Within the studies reported in this review, some notable examples are the performance analysis in FPC and ETC [16–18], the performance prediction of solar hot water systems [19–21], and the performance prediction of other thermal components, such as solar assisted heat pumps [22] and thermal energy storage systems [23]. Ghritlahre and Prasad highlighted the flexibility of the ANN models to interpret complex and incomplete data, while significantly reducing the calculation time required to predict performance and obtain intrinsic characteristics of the system. Different types of neural models such as Multi-Layered Perceptron (MLP), Radial Basis Functions (RBF) and Adaptive-Network-Based Fuzzy Inference System (ANFIS) were applied for performance prediction in the aforementioned works, successfully acquiring greater accuracy than other conventional, linear and nonlinear models.

In recent years, other complex deep architectures such as Recurrent Neural Networks (RNN), have been designed to focus on specific feature extraction techniques for temporal relationships. Long Short-Term Memory (LSTM) RNN have been successfully applied for the estimation of RUL in lithium-ion batteries [12]. Within the most common applications, RNN are particularly well adapted to analyze time series [6]. A major drawback of standard neural networks is the underlying assumption that the training and test examples are independent. Thus, even though ANN have reached outstanding sequencing results without directly modeling time, it has limited power over long-range dependencies [24]. RNN models are, in principle, able to create and process memories of arbitrary sequences of input data, mixing sequential and parallel information processing with efficiency [25]. The results of the aforementioned studies suggest that using deeper architectures to analyze temporal dependencies, such as RNN, could achieve more precise results, allowing a better understanding of the periodical behavior and evaluating the relationships between each component in the system. In the industry, the most common LSTM variant (known as Vanilla LSTM) has been used to estimate the RUL of different components, which has become one of the most challenging and significant decision-making indicators for security and maintenance issues [26].

For instance, in 2018 a comparative study of LSTM in forecasting day-ahead global horizontal irradiance with satellite data was presented [27]. The authors highlighted the fact that previous techniques for forecasting radiation values did not include the use of LSTM, whereas shallow ML algorithms (e.g. Support Vector Machine (SVM), Random Forest, among others) and standard ANN (Feed-Forward Neural Networks (FFNN), RBF) have been frequently used. This work aimed to obtain an accurate forecast for photovoltaic (PV) based energy plants, since PV power production, stability, and storage dimensioning is strongly characterized by fluctuating outputs influenced by instantaneous meteorological conditions. Predictive features are a critical issue for the electrical grid management; and, as shown recently, also for smart-grid applications: planning, storage system sizing and market participation of variable renewable energy sources [28]. However, as solar thermal systems present natural inertia, lower precision is required to obtain acceptable results compared to PV plants. Thus, there is a limited amount of research considering the application of these new techniques for performance diagnosis in thermal systems.

The present work proposes a Deep Learning-based framework for performance prediction for a SHW system based on temperature forecast under different meteorological conditions. We assess specialized DL algorithms which have not been previously applied for performance predictions in thermal systems, particularly SHW configurations. Considering the large and varied improvements observed in the application of DL techniques for PHM, algorithms which have achieved outstanding results analyzing temporal sequences, such as RNN and LSTM, are compared to the more widely-used ANN architectures for this purpose. Several of these models with different hyperparameter configurations are trained with nominal operational data (primarily, temperature measurements) of a SHW system. Their accuracy and precision are compared for predicting future instantaneous values, as well as for short sequences based on the Mean Absolute Error (MAE) and Root Mean Squared-Error (RMSE) scores and variance obtained. For the short sequence predictions, ten randomly selected 1-h long samples occurring during operational hours of the solar field are used for comparison. These results are compared to a naïve persistence model, as well as other regression methods.

Aiming to apply the algorithms to an actual system, the solar-assisted heating system located at the Universidad de Chile is used as a case of study. However, considering that the installation is relatively new, few information regarding the operational conditions is available. Therefore, in order to generate synthetic data the same approach used by Kalogirou et al. [21,29] was applied. By building a detailed simulation model in the Transient System Simulation Program (TRNSYS), a significant amount of synthetic data can be generated for training and validating the DL model. For building and validating the TRNSYS simulation, the nominal data from the manufacturer was employed, introducing design

temperatures, equipment sizes and capacities and the control scheme of each subsystem.

The subsequent sections of this article are organized as follows. Section 2 briefly describes the basics of the Deep Learning methods used in this work, aiming to explain the different variables that need to be taken into consideration for their training. Section 3 presents the configuration of the SHW system, its components, and operation logic. In Section 4, the Deep Learning framework for temperature sequence prediction is presented, containing a description of the simulation approach carried out in TRNSYS software for synthetic data generation and the proposed methodology regarding the design, training, and testing of the Deep Learning models. Results regarding both temperature prediction are discussed in Section 5. Finally, Section 6 concludes and highlights the future challenges for the proposed framework.

2. Deep learning methods

Machine Learning tasks are generally divided into supervised and unsupervised learning. In the case of supervised learning, the training of the model consists of an optimization process which seeks to minimize the error of the predicted output when compared to the label of the original data. After the training process, the model can interpret unused data input and replicate the learned behavior based on the internal representation obtained from the training data. The accuracy of the model will depend on the chosen architecture, its hyperparameters, the nature of the data and the learning process.

2.1. Artificial neural networks (ANN)

ANN are a common technique used in ML for supervised classification and regression tasks (i.e. the desired outputs are known beforehand and thus can be compared with the network's prediction). ANN are defined as sequential lineal regressions evaluated by non-lineal functions. Each regression is called a hidden layer and the number of outputs of each regression in a hidden unit. These are represented by a set of matrices and vectors (weights W and biases b) as the transitions between the input, hidden and output layers of a network, as shown in Fig. 1a. This relationship can also be seen in Equation (1), in which the output of an intermediate layer y_p is calculated by applying a non-linear transfer function σ to a pondered sum obtained from the input data $\boldsymbol{X}$ and the network's trainable parameters (W, b). Usually, a linear activation function is used for the output layer of the network to calculate the regression's result, whereas the non-linear functions of the intermediate layers applied in the present study are described in section 2.4. A frequent choice is the logistic sigmoid function defined in Equation (2), whose values exist between [0,1].

$$y_p = \sigma\left(W^T \boldsymbol{X} + b\right) \tag{1}$$

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2}$$

The training process is usually done by propagating error from the output to the input layers, adjusting the weights and biases within each layer to correctly represent the given data. This is known as backpropagation, by which these parameters are adjusted by chain rule and gradient descent techniques, as presented by Rumelhart et al. [30]. Depth is added to the basic ANN network by stacking several hidden layers, known as Multi-Layered Perceptron (MLP), where consecutive non-linear mathematical operations are applied to each layer. For an MLP (also known as Deep Neural Network or DNN), the input of each hidden layer is the output value of the previous one, and Equation (1) is rewritten as follows.

$$h_i = \sigma\left(h_{i-1}^T W_i + b_i\right) \tag{3}$$

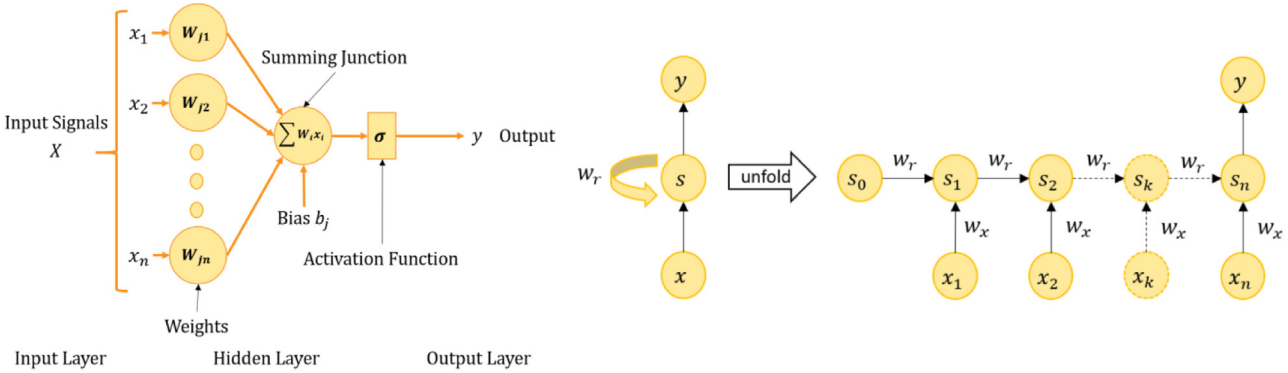
2.2. Recurrent neural networks (RNN)

However, in an RNN, each unrolled layer will receive the current input data x_t, as well as the previous hidden state of the node h_{t-1}. The output $y_{pred,t}$ will depend on the hidden value of the current time step h_t. The unfolded temporal structure is depicted in Fig. 1b. Each layer in the unfolded network represents a time step with shared weights for input, hidden and output data (W_x, W_r, W_y) and biases in hidden and output layers (b_h, b_y), which are expressed as:

$$h(t) = \sigma(W_x \cdot x_t + W_r \cdot h_{t-1} + b_h) \tag{4}$$

$$y_{pred,\,t} = g\left(W_y^T \cdot h_t + b_y\right) \tag{5}$$

Thus, the learning algorithm must be modified to a backpropagation through time (BPTT), which operates similarly to the gradient descent to adjust the network's weights [31]. However, for long-range dependencies, vanishing or exploding gradients may appear when propagating errors towards the inner layers [32], obstructing the learning process. These optimization problems



a) Basic structure of an MLP. b) Unfolded structure of an RNN.

Fig. 1. ANN and RNN methods details.

increase the computational demand for using these techniques. Thus, other architectures, as well as the use of regularization techniques (e.g. dropout, early stopping), have been proposed to address this challenge.

2.3. Long short-term memory recurrent neural network (LSTM)

Long Short-Term Memory (LSTM) RNN introduce a memory cell designed to overcome the issue of vanishing gradients, replacing the nodes in the hidden layers [33]. The classical architecture of LSTM-RNN consists of the interaction of gates or processes within a memory cell, each of which is independent RNN. Long-term memory is stored in what is called the "cell state" and is selectively updated with new input data stored in the short-term memory [33]. In Fig. 2, a schematic representation of Vanilla LSTM is shown. Here, three gates control the memory update from a previous state at timestep $t-1$ and the new input data at timestep t. Each gate has its own set of weights (W, U) and biases (b) of a regular RNN and use a logistic sigmoid activation function (see Equation (2)). As logistic sigmoid functions have values between [0,1], it represents the regulation of the flow of information through each gate. From left to right in the center cell of Fig. 2, there are the forget gate, input gate and output gate. Each cell has three inputs: the previous cell-state c_{t-1}, the previous output h_{t-1} and the new data x_t. There are two outputs from the cell: the present cell state c_t and the present output of the hidden state h_t representing the current predicted value.

When a new input x_t enters the memory cell, the forget gate will select the information to be thrown away from the previous cell-state c_{t-1}, represented as follows,

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{6}$$

The input gate, on the other hand, selects the information that will be updated while a new cell-state candidate is built from the input data. This is represented by the following equations.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{7}$$

$$a_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{8}$$

A combination from the forget gate (Equation (6)) with the previous cell-state c_{t-1} and from the input gate (Equation (7)) with the new candidate (Equation (8)) will create the new cell-state through Equation (9). Note that the new candidate is determined by a *tanh* activation function, which ranges between $[-1, 1]$ for normalization purposes.

$$c_t = f_t \cdot c_{t-1} + i_t \cdot a_t \tag{9}$$

Finally, the output gate selects which part of the cell-state obtained from Equation (9) will be used as a useful output h_t.

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{10}$$

$$h_t = o_t \cdot tanh(c_t) \tag{11}$$

Other variants to LSTM have been designed, such as Gated Recurrent Units (GRU). This simplified version merges the internal state with the hidden state and the forget gate and input gate [6]. In Ref. [26] it was shown that although LSTM performs better for short-term prediction and GRU for long-term prediction, the difference is not significant, and both outperform the standard RNN. Additionally, Bidirectional Recurrent Neural Networks (BRNN) introduces inputs from previous and future timesteps to predict outputs [34].

2.4. Hyperparameters and training process

Parametric models, such as DNN, are defined by a set of trainable and non-trainable parameters. The latter, also known as hyperparameters, define the basis of the architecture. These values must be selected based on the nature of the input data, the task the model is required to do and overall performance, among other conditions. To define a model, the meaning and relationships between hyperparameters must be considered. Depending on the nature of the data, a greater number of layers and units per layer increases the ability to map complex functions. Deeper architectures with a smaller number of units per layer are generally preferred to wide shallow layers for complex data mapping [35,36], such as the representation of physical phenomena. Both feature extraction and depth are fundamental for the model to obtain an accurate abstract representation in its internal latent space. From the learned latent space, the output of the model is extracted and its performance in different tasks depends on how similar or dissimilar is the test data compared with the training set. The output of each layer is determined by the previous inputs, weights, and biases, and the activation function used. For internal hidden layers, non-linear functions such as hyperbolic tangent (Equation (12)) or

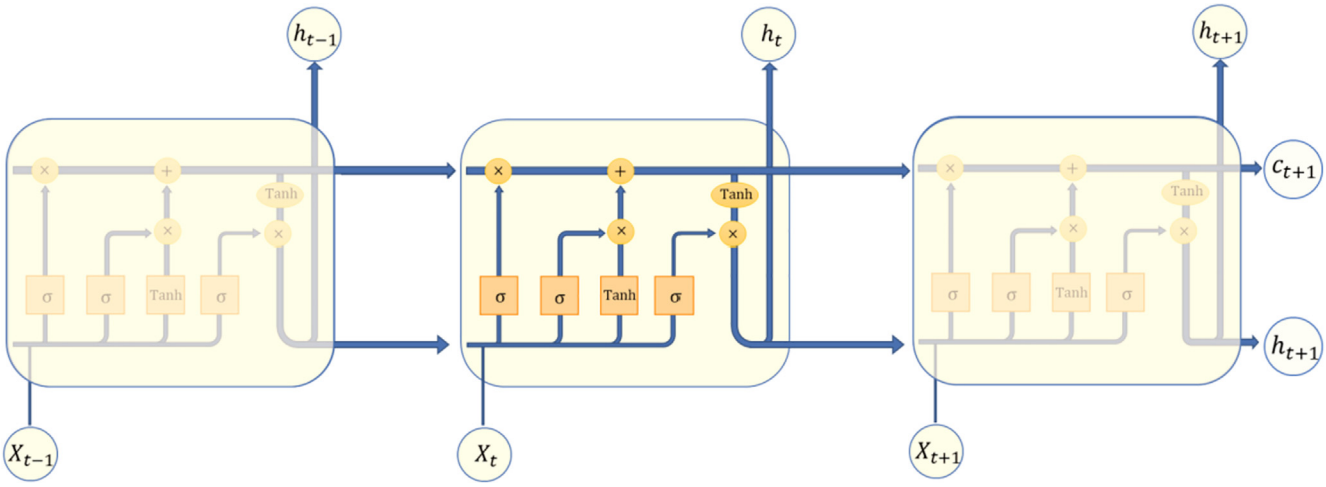


Fig. 2. Vanilla LSTM method.

Rectified Linear Units (ReLU) (Equation (13)) are frequent choices for RNN-based models.

$$tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{12}$$

$$ReLU(x) = \begin{cases} 0 \; for \; x < 0 \\ x \; for \; x \geq 0 \end{cases} \tag{13}$$

The training process is defined by the optimizer, learning rate and loss functions. These determine the method and rate with which the weights and biases are updated throughout the network by backpropagation. Weights can be randomly initialized from normal distributions, as well as other initialization techniques. Various optimizers exist, however, adaptive methods such as Adam [37] and RMSProp [38] obtain more stable results than other optimization techniques [39].

The training process is done through partitions of the training data into batches. This determines the number of random samples which are processed through the network per training iteration, balancing the trade-off between training time and loss of generalization. Each time all batches are passed forward and backward through the network, an iteration epoch is completed. The ideal number of epochs depends on the convergence speed of the model. Other configurations and hyperparameter settings can be used, such as regularization techniques, to increase convergence and reduce training times. The use of the Mean-Squared Error (MSE) as a loss function is frequent for supervised ML tasks, as shown in Equation (14). Consequently, the Root Mean-Squared Error (RMSE) is frequently used as an error metric for the predicted values. Searching for optimal hyperparameter values is usually time and resource-consuming process. As such, depending on the desired or acceptable precision for prediction values, a trial-and-error search is a viable alternative.

$$\text{MSE} = \sum_{i=1}^{n} \frac{\left(y_{real,i} - y_{pred,i}\right)^2}{n} \tag{14}$$

3. System description

The studied SHW system corresponds to a section of the climatization system of a building at the Physical and Mathematical Sciences Faculty of the Universidad de Chile, located in Santiago, Chile. The SHW system aims to supply the locker room's showers with sanitary hot water, where the nominal daily water demand is estimated as 24,000 L at 40 °C for 12 h. The solar-assisted heating system is comprised of two separate water circuits: a pre-heating section in which the heat input is delivered by solar collectors and a heat-recovery chiller; and a heating circuit in which the energy is supplied by heat pumps. The schematic diagram of the SHW system is presented in Fig. 3. Red lines represent the hot water flowing in the system, while the blue lines correspond to the cool water. The SHW is designed to preheat the sanitary water from 10 °C to 40 °C during the summer and to 30–35 °C during the winter. When the available solar radiation is not enough to reach these temperatures through the solar collectors, additional heat is provided by the heat-recovery chiller, which also regulates the indoor swimming pool's temperature as a separate heat load. From both sources, heat is driven to the 4 m^3 tanks serving as intermediate heat storage at 35–40 °C. The heating section provides the energy for reaching 60 °C, by means of four heat pumps, and subsequently, that hot water is stored in four tanks. Mains water is used to regulate the output temperature and as make-up water in the preheating section. Finally, the unused hot water reentries the system at the two first water tanks (Pre-Heat Tank 1-2 in Fig. 3). The mentioned components, along with flat plate type heat exchangers and centrifugal pumps are monitored in terms of temperature and operation status.

The solar field is composed of ETC heat pipe collectors, accounting for a total absorption area of 105.6 m^2. From the common inlet of the solar field, the flow is divided into thirteen collector strings connected in series, eight rows of three collectors and five rows of four collectors each. Each row is supported on metallic bases which maintain an inclination angle of 15° and a north orientation. In Tables 1 and 2, the reported optical and thermal properties under test conditions of the installed solar collectors (Hitek Solar NSC 58-30 model) are presented.

The Incidence Angle Modifier (IAM) values in Table 1 express how the optical properties, transmittance τ and absorptance α, of the solar collector's components vary depending on the incidence angle θ of the solar radiation on the collector's surface. As ETC has nonsymmetrical cover optical properties, a global $K_{\tau\alpha}$ must be considered depending on both the transverse θ_T and longitudinal θ_L incidence angles as shown in Equation (15) [40]. This, in turn, affects the solar collector's overall efficiency presented in Equation (16), where F_R represents the heat removal factor, U_L is the overall thermal loss coefficient, T_i is the fluid inlet temperature, T_a is the ambient temperature and $(\tau\alpha)_n$ is the transmittance-absorptance product at normal incidence angle [41]. The used operational parameters for the heat-recovery chiller are based on a Thermocold CWC Prozone 1320 Z C model.

$$K_{\tau\alpha}(\theta) = K_{\tau\alpha}(\theta_T) \cdot K_{\tau\alpha}(\theta_L) \tag{15}$$

$$\eta_i = F_R\left[G_T K_{\tau\alpha}(\tau\alpha)_n - U_L(T_i - T_a)\right] \tag{16}$$

4. Deep learning framework for temperature sequence prediction

The purpose of the proposed Deep Learning-based framework is to predict future values of the solar collector's outlet temperature based on the generated data from the TRNSYS simulation. The proposed framework is divided into three mains steps: data recollection, development of temperature prediction models and the assessment of the performance prediction power of different model configurations.

In Section 4.1, a brief description of the simulation approach in TRNSYS software is presented, considering the construction of the physical model based on technical information from the manufacturer. From this simulation, synthetic data was generated in nominal design conditions.

Section 4.2 describes the training process for the proposed models based on DNN, RNN and LSTM architectures, considering the nominal operating conditions of the SHW system. The design and selection of the network's hyperparameters are carried out by training and testing different configurations based on the mentioned architectures. To do so, the performance of DNN, RNN and LSTM architectures are compared under similar conditions and hyperparameters to identify strengths and weaknesses of each model.

Hence, the assessment of the predictive performance of each model is divided based on two predictions goals. First, the performance of each model is evaluated through the RMSE metric for single future temperature values. The second goal is to predict

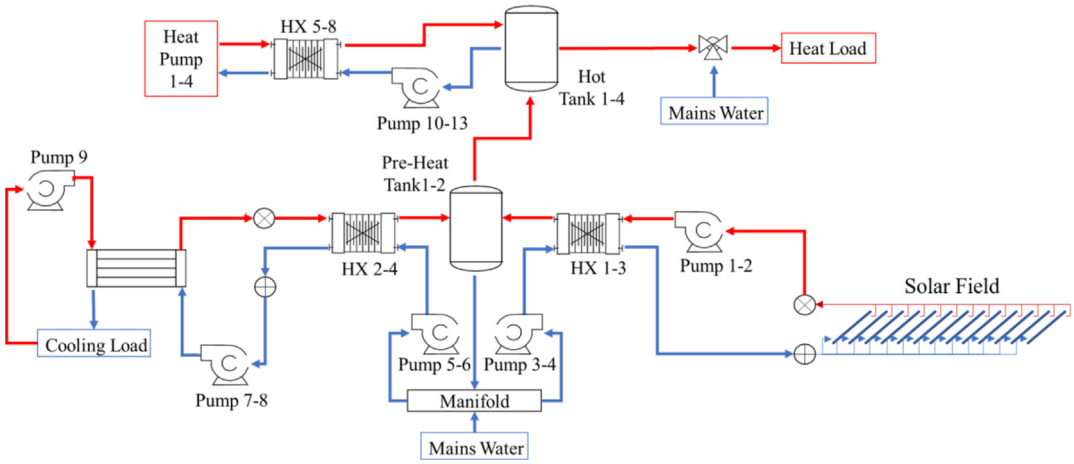


Fig. 3. Process Flow Diagram of the pre-heating section of the SHW system.

Table 1
Incidence Angle Modifiers (IAM) values for Hitek Solar NSC model.

	10°	20°	30°	40°	50°	60°	70°
$K_{\tau\alpha}(\theta_T)$	1.010	1.019	1.056	1.151	1.452	1.462	1.261
$K_{\tau\alpha}(\theta_L)$	0.999	0.994	1.018	0.974	0.952	0.913	0.833

Table 2
Thermal capacities of Hitek Solar NSC model.

η_0	0.618
a_1 [W/m^{2}K]	1.377
a_2 [W/m^{2}K]	0.018
Effective Thermal Capacity [kJ/m^2K]	5.684

several timesteps consecutively in a 2-h long time-window based on the previous prediction values, reproducing the sequences' temporal behavior during short future periods of time reporting both MAE and RMSE scores. These results are presented in Section 5.

4.1. Simulation approach and data collection

TRNSYS was selected for carrying out the physical simulations of the complete system, owing it to its modularity and open-source structure. TRNSYS operates with types or component blocks which are integrated to experimentally validated equations and calculation modes. Currently, there are more than three hundred operational types, allowing to simulate the operation of each component of the system using a specific TRNSYS module (Type). TRNSYS has been applied and validated in several applications related to solar energy systems, such as the systems considered in the present study. For instance, it was validated for representing the stratification of hot water storage systems, and also for large scale solar thermal systems for pool heating [42], and for different solar technologies as flat plate and heat pipe collectors [43], and even for large heating networks [44]. Finally, TRNSYS also has been successfully validated for representing HVAC systems [45]. Therefore, considering the capabilities of the software, a detailed model was built representing the operation of the heating system shown in Fig. 3, calibrating it by using data from manufacturers.

To properly simulate the system in TRSNSYS, the following data is considered:

- Meteorological data: solar radiation, ambient temperature, wind speed, and wind direction are required to properly characterize the environmental conditions in which the SHW system operates. A complete meteorological station is placed next to the solar collectors' array, where the radiation and other meteorological measurements were taken and processed and then introduced in the TRNSYS deck.
- SHW system information: design conditions and the main features of the system are defined, according to the information provided by the manufacturer. Also, the information about secondary equipment such as heat pumps, heat exchangers, heat accumulators, centrifugal pumps and the control system was also considered. Nevertheless, appropriate simplifications were introduced to reduce the computational cost:
  - o Mains water temperature is estimated based on numeric correlations presented in Ref. [46].
  - o Heat pumps are replaced by auxiliary water heaters with the same nominal heating capacities.
  - o The temperature control systems are simplified as the following:
    - If the temperature registered at the outlet of the solar collector field is higher than the average pre-heating tank temperatures, Pumps 1-2 are activated.
    - Hot water is dispatched from the pre-heating tanks to the high-temperature tanks if there is an active demand in the system.
    - Mixing valves are activated if the outlet temperature of the heat tanks is higher than 45 °C, in which case mains water is introduced before being dispatched.
  - o The system operates for 14 h on weekdays, between 7 a.m. and 9 p.m. During weekends the operation hours are reduced to 11, from 8 a.m. to 7 p.m.
  - o As no real-time measurements of the hot water demand profile were available, a weekly profile was drawn from estimations of user experience and the design conditions.

For each subsystem simulated in TRNSYS, a set of variables is monitored and extracted for analysis considering a sampling frequency of a minute. The extracted values consist basically of temperatures and control signals, leaving aside flow and energy consumption values due to the lack of experimental validation options for these variables.

The length of the simulation window depends on the availability of radiation and meteorological data which is fed into the TRNSYS model, consisting of the period between April 7th and September

22nd, 2018. The software then generates a minute-based simulation of the behavior of the monitored variables during the selected time-window. Although not experimentally validated, a comparison between the main temperatures obtained with the simulation and the design temperatures is presented in Table 3. Here, the overall difference accounts temperatures 19.5% higher in the simulation. This is an average difference of 7.7 °C, mainly caused by the higher participation of the solar field and the ideal operating conditions of the heat-recovery chiller. As seen in Ref. [43], when comparing modelled and measured data for a forced circulation SHW composed of ETC, the obtained percentage MAE for the collector outlet temperature, heat collected and heat delivered to the load were 18.4%, 16.8% and 7.6%, respectively. This, as well as the higher temperatures at the outlet of the heat-recovery chiller may explain the 29.3% difference between the simulated and designed average temperatures in the Pre-Heat Tank 1-2, as some previous works suggest that tank's outlet temperature reports errors lower than 5% for validated TRNSYS simulations [47]. Despite this difference, the TRNSYS model allows the generation of operational data isolated from external effects and unquantified degradation in the actual components which would generate unnecessary noise for this exploration of DL techniques for prediction.

For the prediction of the solar collector's outlet temperature the following data is considered as input variables: ambient temperature, the solar field's inlet temperature, the control signal of pumps 1-2, the inlet temperature at the heat exchangers HX 1-3 (as shown in Fig. 3) and previous values of the solar collector's outlet temperatures. These inputs are presented as matrices of size [features, simulation length] and then, averaged over a period of 6 min to account for the natural thermal inertia of the system and the variability of solar radiation, as minute-based measurements may suffer from additional uncertainties caused by atmospheric phenomena.

4.2. Proposed deep learning models

Different configurations of the methods based on DNN, RNN, and LSTM were trained and tested to assess and compare their performance under similar conditions for the prediction of the solar collector's outlet temperature. As mentioned in Section 2, the studied DL methods are defined by their hyperparameters. Some of these are the unit type, the number of units and layers, activation function, optimizer, learning rate, training epochs and batch size. Additionally, another tested configuration is the length of the sequence used as input to each model. The models are trained with sequences of one, three and seven days of continuous data to analyze the impact on each model's performance. Longer and more complex temporal relationships can be extracted in extended time-windows, regarding the availability of solar resource and hot water consumption patterns. As mentioned above, this length impacts the performance of RNN and LSTM-based models. The processed data has been averaged over 6 min, and thus a day is converted a data sequence of 240 timesteps, while three and seven days are converted to 720 and 1680 timesteps, respectively.

The timeseries data is reshaped into sequence matrices by a sliding window of N_f features and t_w timesteps. This sliding window of size $t_w \times N_f$ passes over the original dataset, extracting the input values $\boldsymbol{X}$ for each future predicted value. A sliding step $s_t = 1$ of the sliding window is used to extract continuous timeseries. The values mapped by the sliding window are then reshaped into matrices where each row represents a sequence of a whole time-window, x_t. Similarly, for the labels, the following timestep for the outlet temperature of the solar collector y_{t+1} are extracted as a vector resulting in a prediction horizon of 1 timestep, corresponding to 6 min in simulation time. For each resulting x_t time-window, the input data is a vector formed by the flattened matrix of $N_f = 5$ features and $t_w = \{240, 720, 1680\}$ timesteps.

Schematic representations of example architectures of DNN, RNN, and LSTM-based models are shown in Fig. 4a, Figs. 4b and 5, respectively. Here, N_h represents the number of hidden units in the MLP layer, while X, h and y are the input data and the output from the hidden layers and the output layer, respectively.

For the basic DNN or MLP model, the first layer receives the unfolded sequence of features during the entire time-window, which is then passed through two hidden layers to obtain the output value. Both the double-layered RNN and the LSTM incorporate an MLP as hidden layers prior to the output layer to obtain the desired predicted value.

The extracted dataset is randomly divided into training (64%), validation (16%) and testing sets (20%). Training and validation sets are normalized between [0,1]. The numbers of samples for training, validation and test sets per time-window size are presented in Table 4. The training process is carried out only on the training set and the validation set is used to assess if the model overfits the data. RMSE values are calculated over both the training and the validation subsets in each iteration epoch. After the training process is concluded, the test set is then used to evaluate the performance of the model with previously unseen data. RMSE and other metrics are used to evaluate the models' predictions under the test set, such as variance, explained variance score and R^2.

The processed data series were fed into the proposed architectures, considering also the hyperparameters presented in Table 5. An Intel®Core™ i7-6700K CPU @ 4.00 GHz ×8 with a TITAN Xp/PCIe/SSE2 Graphics card is used for training and testing the models.

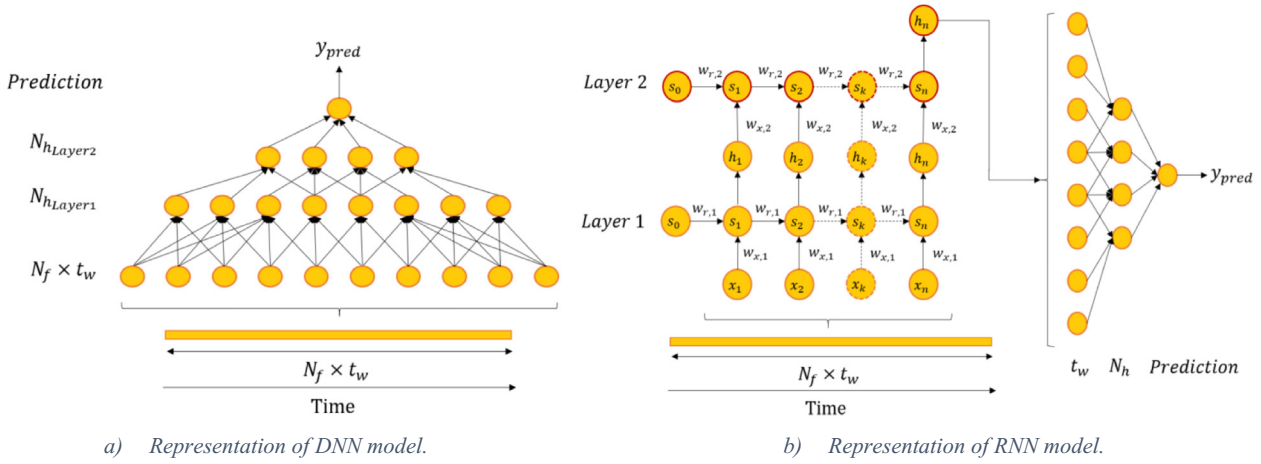
The testing of the models is divided into two phases. At first, single temperature values are predicted from t_w timesteps from the input data matrix of N_f features. The output of the model, y_{pred}, is compared to the corresponding label y_{t+1}. Although, as described in Section 5.1, the trained models yield low RMSE values, isolated temperature predictions do not yield enough information to determine if the sequential behavior of the timeseries has been effectively mapped into the latent space of the trained models. Thus, the ability to accurately predict longer sequences is also an interesting result. As a second evaluation of the prediction abilities of the proposed models, a sequence of ten timesteps (equivaling to a prediction horizon of 1 h in the future) will be consecutively predicted. When generating a sequence, the model will update the sliding temporal window over the data and thus will incorporate previous predictions' errors in future values. Thus, results may significantly vary from single value predictions depending on whether the trained models are able to incorporate the underlying sequential behavior of the temperature data.

Table 3
Comparison between design and simulated temperatures in the SHW system.

Temperature °C	Design	Simulated	Difference
Pre-Heat Tank	35.0	45.2	29.3%
HR Ch. Outlet	50.0	54.5	9.0%
Solar Inlet	37.0	45.0	21.7%
Solar Outlet	45.0	53.2	18.2%
		Average	**19.5%**

5. Results and discussion

This section presents the results obtained when assessing the DL architectures applied for the two tasks defined previously, temperature prediction as single values and as temperature sequences. Initially, various potential candidate DL models based on the hyperparameters shown in Table 5 are trained and then their



a) Representation of DNN model. b) Representation of RNN model.

Fig. 4. Schematic representation of proposed Deep ANN and RNN architecture.

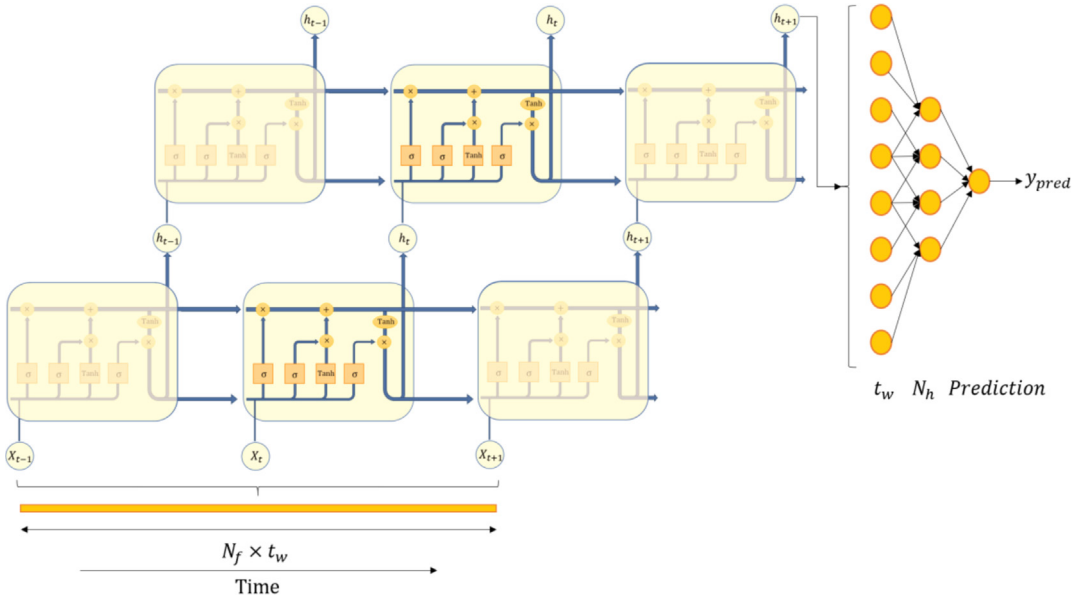


Fig. 5. Schematic representation of proposed Deep LSTM architecture.

Table 4
Number of samples per time-window length.

Dataset	N° Previous Days	Time-window Size	Train Set	Validation Set	Test Set
Solar Collector Outlet T°	1	240	5836	1460	1824
	3	720	17510	4378	5472
	7	1680	23654	5919	7393

Table 5
Tested hyperparameters and architectures.

Model	DNN, RNN, LSTM
Data Length	240, 720, 1680 timesteps
Number of layers	1-2-3
Number of RNN units	16, 32, 64, 128, 256
Number of MLP units	128, 256, 512, 1024
Activation Function	Tanh, ReLU
Optimizer	Adam, RMSProp
Epochs	50, 75, 100, 150, 200
Batch Size	32, 64, 128

performance is evaluated over the previously unseen test set. The validation set is only used to monitor the training process and was not used for training. Additionally, the results yield by the DL models are compared to traditional and naïve regression methods.

5.1. Temperature prediction

The first criteria to assess the performance of the trained DNN, RNN and LSTM-based models is to obtain low RMSE values with a small dispersion of the predicted temperature values under nominal operational conditions. From the models' point of view, considering the information currently available, an instantaneous

temperature prediction with an RMSE below 1 °C is not directly distinguishable from the thermal systems' natural variations. In Table 6, the obtained test errors and their related performance metrics are presented for different configurations, where a variance below 1 °C have been highlighted in bold font.

Low values of RMSE under 2 °C are obtained for most of the proposed models when trained to predict the solar collector outlet's temperature. The best performing model corresponds to the DRNN with three layers of [64,32,16] units each trained for the three-day temporal image dataset, obtaining a mean difference between the predicted y_{pred} and real y_{t+1} of 4.45%. However, the lowest relative difference of 3.45% is obtained by the DLSTM model comprised of [128,64] units, which is also trained for a three-day time-window. This relative error is calculated by averaging the difference between predicted and simulated temperature values, expressed as a percentage.

When compared to regular regression techniques, the temperature fluctuations at the outlet of the solar field are easily predicted within short-term windows, such as 6 min, due to natural thermal inertia. Statistic metrics are presented for Bayesian Ridge Regression (BR), Gaussian Process Regression (GPR), Linear Regression (LR) and Persistence (PR) methods [48–50]. These metrics are calculated under the same conditions and test sets as the DL models and shown in Table 7. However, as presented in the next section, the performance of these models is significantly reduced when longer temporal relationships are introduced; a result already seen in the context of solar radiation forecasting [50].

5.2. Sequence Prediction

To further analyze the difference between the proposed RNN and LSTM models, ten random samples from the original test set have been filtered to illustrate the temperature sequence prediction behavior. These time-windows are selected during the operational hours determined by the control system of the solar Pumps 1-2. Examples are presented in Figs. 6–8. This means that an extended horizon of, for example, ten timesteps, are predicted consecutively from the same original temperature sequence. The last predicted value is appended to the next time-window the model will process, coupled with the other simulated variables, replicating what would be an online monitoring scenario. Thus, the next value will not only carry its own uncertainty, but also the previous values' propagated uncertainty.

As mentioned before, the RNN-based models yielded lower test RMSE for single temperature predicted values. However, in Fig. 6 it can be seen that the RNN based models are not able to reproduce the data sequence correctly. For instance, in Sample 2, even though the temperature difference is low (3 °C at most), the predicted behavior is inverted, leading to an underestimation of the temperature values. For sample 10, however, the three-layered RNN model of [64,32,16] units outperforms the two-layered version of [64,32] units, implying the existence of a more abstract relationship within the data which deeper models can extract. As mentioned in Section 4.1, a greater number of layers can extract features from more complex data better than shallow architectures depending on the nature of this data and the length of the sequences to be learned.

Less surprisingly, due to the fact that DNN architectures are not specialized in time series analysis, Fig. 7 shows that these models do not predict the sequence properly, exhibiting a similar behavior than the RNN-based models with an error rate as high as 6 °C. On the other hand, the LSTM-based models achieved worse results than the RNN-based models predicted single temperature values; yet, as observed in Fig. 8, they exhibit better capabilities of following the temperature sequence with lower differences between the predicted and real values. The DLSTM-2 model, with [128, 64] units, achieves a maximum difference of 0.5 °C on Sample 2. Even though the model exhibits the same problems of desynchronizing the predicted peak values for Sample 8, the error is reduced to 1 °C.

Three of the best performing models, each for every architecture studied, are then selected to further explore their sequence prediction performance. The mean and standard deviation of the obtained MAE and RMSE values are presented in Table 8 for DNN, RNN and LSTM-based models for a 3-day length temporal frames. Note that, although the DNN-based model is outperformed by both the other models in terms of the mean RMSE and MAE obtained, it has a smaller standard deviation than the RNN model, while the LSTM model possesses the lowest RMSE score (mean RMSE of 1.27 °C, STD of 1.07 °C). These results also corroborate the previous results (see Table 6) that suggest that the LSTM-based model exhibits a better ability to predict the temporal sequence at hand, compromising between a slightly lower accuracy (0.55 °C MAE compared to 0.41 °C) but higher precision than the RNN-based models, as shown in Table 8.

The performance of these DL models is compared to other regression techniques in Table 9. These regression techniques were tested under the same configuration as the DL models for temperature sequence prediction. As a result, models which yield reasonable results for single timestep predictions (presented in Table 7) are outperformed by specialized deep learning algorithms as the prediction horizon increases. It must be noted that the PR model obtains the best short-term results in terms of MAE and RMSE, compared to the other methods. That effect may be a result

Table 6
Test errors and statistical metrics for trained models.

Model	Num. Days	Units	Test RMSE, °C	Test Error %	Variance, °C^2	R2	Explained variance
DNN	1	64–32	1.74	7.82	2.910	0.986	0.986
RNN	1	64	1.68	6.20	2.460	0.987	0.988
DRNN	1	64–32	1.28	5.50	1.221	0.993	0.993
LSTM	1	64	2.5	7.70	3.410	0.982	0.984
DNN-1	3	64–32	1.43	6.97	1.587	0.988	0.991
DNN-2	3	64-32-16	1.29	6.38	1.502	0.990	0.991
DNN-3	3	128–64	1.66	9.08	2.425	0.984	0.986
DRNN-1	3	64–32	0.92	7.10	1.516	0.991	0.991
DRNN-2	**3**	**64–32–16**	**0.89**	**4.45**	**0.619**	**0.996**	**0.996**
DLSTM-1	**3**	**64–32**	**1.42**	**4.33**	**0.618**	**0.996**	**0.996**
DLSTM-2	**3**	**128–64**	**1.47**	**3.45**	**0.520**	**0.997**	**0.997**
DLSTM-3	**3**	**128–64–32**	**1.66**	**5.99**	**0.898**	**0.993**	**0.995**
DNN	7	64-32-16	1.26	5.38	1.550	0.991	0.991
DRNN	**7**	**64–32**	**0.94**	**4.43**	**0.644**	**0.996**	**0.996**
DLSTM	7	64–32	1.38	6.10	1.510	0.991	0.991

Table 7
Test errors and statistical metrics for regression models.

Model	Num. Days	Test RMSE, °C	Test Error %	Variance, $°C^2$	R2	Explained variance
BR	**3**	**0.54**	**2.24**	**0.29**	**0.998**	**0.998**
GPR	3	0.57	2.48	0.32	0.998	0.998
LR	3	0.57	2.50	0.32	0.998	0.998
PR.	3	0.826	3.09	0.68	0.996	0.996

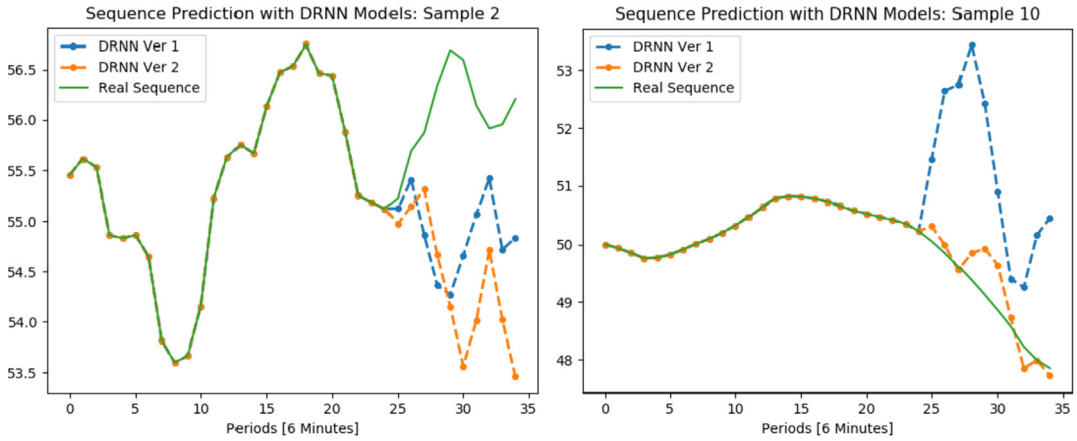


Fig. 6. Temperature sequence prediction examples with RNN-based model.

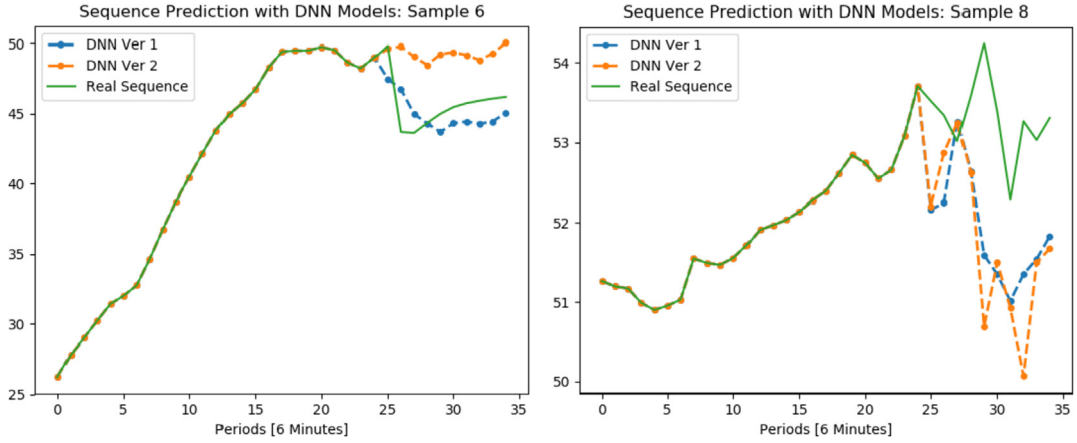


Fig. 7. Temperature sequence prediction examples with DNN-based model.

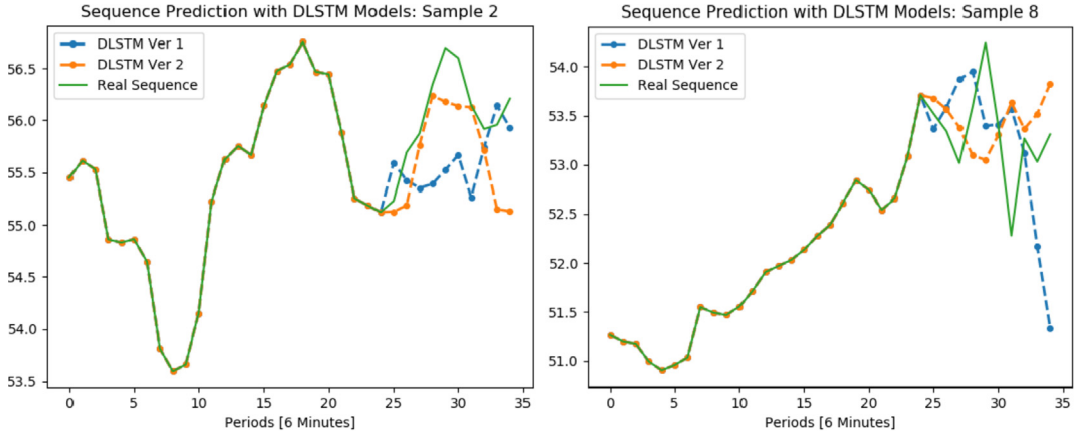


Fig. 8. Temperature sequence prediction examples with LSTM-based model.

Table 8
Sequence Reconstruction error by model for the solar collector outlet temperature and a 3-Day time-window.

Prediction Errors	DNN-2			DRNN-2			DLSTM-2		
Sample	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD	RMSE
1	3.39	5.35	6.33	2.11	4.28	5.46	0.57	3.99	4.03
2	2.55	0.92	2.71	1.66	0.94	1.77	0.46	0.49	0.67
3	0.74	0.50	0.89	−0.10	0.59	2.55	−0.75	0.36	0.83
4	0.90	0.31	0.95	0.28	1.53	1.29	0.08	0.43	0.43
5	−0.49	0.67	0.83	−0.13	0.40	0.04	0.03	0.55	0.55
6	0.62	1.57	1.68	−1.36	2.59	2.89	1.09	2.13	2.39
7	0.86	0.40	0.95	0.49	0.69	0.65	1.40	0.36	1.45
8	1.40	0.73	1.58	−0.32	1.06	0.30	0.12	0.88	0.89
9	1.17	0.56	1.29	0.16	0.41	0.26	0.80	0.43	0.91
10	0.62	0.69	0.92	−0.20	0.36	0.60	0.17	0.50	0.53
Average	**1.18**	**1.17**	**1.81**	**0.16**	**1.28**	**1.58**	**0.40**	**1.01**	**1.27**
STD			**1.60**			**1.68**			**1.07**
MAE	**1.27**			**0.41**			**0.55**		

Table 9
Sequence Reconstruction error for the solar collector outlet temperature and a 3-Day time-window.

Mean Errors	BR			GPR			LR			PR		
Sample	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD	RMSE
1	3.17	6.20	6.96	3.16	6.38	7.12	3.15	6.38	7.12	3.76	6.41	7.43
2	0.96	0.42	1.05	−0.27	0.23	0.36	−0.30	0.22	0.38	0.94	0.41	1.03
3	0.56	0.41	0.70	1.79	0.75	1.94	1.81	0.75	1.96	−0.81	0.79	1.13
4	0.5	0.29	0.58	0.65	0.32	0.72	0.67	0.32	0.75	0.24	0.21	0.32
5	−2.32	0.83	2.46	−3.13	1.18	3.34	−3.12	1.18	3.34	2.37	1.27	2.69
6	−8.64	3.47	9.32	−10.15	4.19	10.98	−10.17	4.20	11.01	−3.40	1.68	3.80
7	−0.81	0.23	0.84	−1.25	0.40	1.32	−1.26	0.40	1.32	1.62	0.77	1.79
8	−1.8	0.58	1.89	−2.21	0.90	2.39	−2.22	0.92	2.41	−0.41	0.47	0.63
9	0.58	0.51	0.77	0.62	0.46	0.77	0.62	0.45	0.77	3.53	1.26	3.75
10	0.91	0.14	0.92	1.45	0.35	1.49	1.47	0.35	1.51	−1.27	0.74	1.47
Average	**−0.69**	**1.31**	**2.55**	**−0.93**	**1.52**	**3.04**	**−0.94**	**1.52**	**3.06**	**0.88**	**1.40**	**2.40**
STD			**2.90**			**3.23**			**3.23**			**2.03**
MAE	**2.03**			**2.47**			**2.48**			**1.84**		

of the small difference in magnitude between temperature values in short-term timesteps and not a direct consequence of the model's performance. While naïve methods such as PR may offer a quick alternative for short-term forecasts, it depends on the data update frequency and would not be suitable for predicting or detecting unexpected changes in the system, such as early anomaly detection, in extended inter-hourly forecast horizons. Furthermore, they are not appropriate for long-term predictions due to the inertia usually observed in thermal systems and its dependence on external factors, such as meteorological conditions.

When comparing the best performing DL-based models during the operating hours of the SHW system, the DNN model tends to underestimate the outlet temperatures, while the RNN version tends to overestimate it. Although the LSTM models predict the general temperature tendencies in Samples 6, 8 and 10, they also present discrepancies with the simulated values, such as in Sample 2 shown in Fig. 9. Even if the magnitude of the error is small, this is preferable than unsynchronized predicted sequences compared to the actual data.

As mentioned before, while the RNN and LSTM models consistently perform better than the DNN model, the latter yields a lower variance for more than half of the samples tested. This result may explain the reason for more extensive use of DNN models compared to more complex models which require significant efforts to select adequate hyperparameters and training conditions, as well as longer training times. For the same length of temporal window provided, for instance three days, LSTM models require an average of 5.24 h to train, while RNN train in 3.25 h. This larger time is due to the larger number of epochs required for the LTSM training error to converge steadily. However, the results yielded by the LSTM model suggest that sequence predictions can effectively benefit from the use of this architecture, given the availability of long enough previous data sequences.

From the models assessed in the previous section, the DLSTM-2 is indicated as the best architecture since it displays the best performance under the stated criteria of a low mean and variance of the RMSE for instantaneous and sequence temperature predictions. The hyperparameters for the DLSTM-2 model are shown in Table 10.

Following the assessment of different models and hyperparameters, further analysis is required to increase the utility of this framework for future tool developments, such as anomaly detection. Actions regarding data preprocessing can be carried out to increase the model's effectiveness. As presented, the use of the whole previous day as input data is enough to predict the following timesteps' output accurately for DNN and RNN. However, LSTM requires longer sequences to reach comparable results. The improvement of the LSTM's performance with a three day-length temporal matrix reaches an RMSE below 1 °C, but the difference is not significant to when a seven-day sequence is analyzed.

Other lengths of time-windows could yield better interpretations of the temporal sequences, such as a five-day time-window representing the weekday period which matches the higher demand for hot water in the system. The use of a time-window coherent with the demand pattern may yield a better latent representation of the data, and subsequently, more accurate and precise predictions. Additionally, the training process could be carried out with data only from operating hours. This could reduce

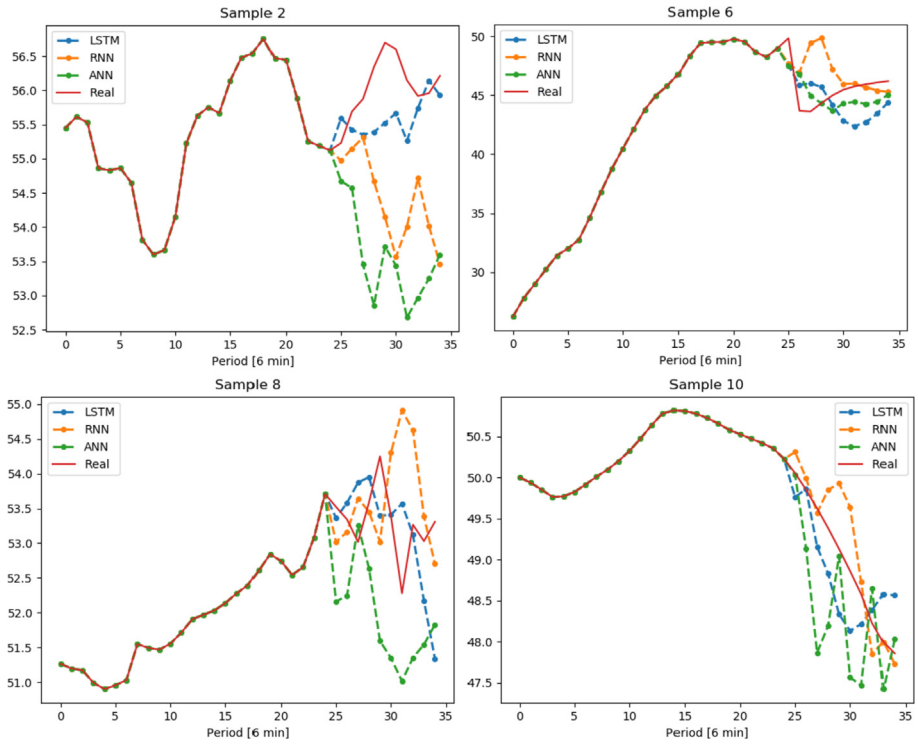


Fig. 9. Reconstruction samples for solar collector outlet temperature.

Table 10
Selected deep learning architecture.

Hyperparameter	Value
Architecture	LSTM
Units & layers	[128,64]
Activation function	Tanh
Optimizer	RMSProp
Learning rate	10^{-3}
Batch size	64
Epochs	150
Length of time-window	3 days
MLP hidden units	512

an unnecessary bias to lower temperatures that are more frequent within the simulated values than normal operating temperatures due to the daily trend of the solar resource.

6. Conclusions

This work presented an assessment of a Deep Learning-based performance prediction framework for a solar thermal system. In this context, several models with different hyperparameter configurations were proposed as potential candidates for temperature prediction in the system. Among the different architectures analyzed, the proposed LSTM model yields more accurate predictions than DNN model as well as more precise values than RNN for temperature sequence predictions. All three models outperformed the naïve persistence forecasts and other regression techniques, such as Bayesian Ridge, Gaussian Process and Linear Regression, for temperature sequence predictions. The assessment was applied considering a solar hot water heating plant located in Chile and using synthetic operational data, as an initial attempt for capturing the nominal behavior of the system. The physical model was built in TRNSYS software based on the nominal conditions.

The limitation of this framework is constituted by the synthetic data that was generated with a simulation platform. TRNSYS is a validated physics-based model that allows the incorporation of the real SHW installations technical specifications and quality meteorological measurement. Yet, the simplifications established in the simulation program, such as the restrictions on the control system and the ideal conditions of the auxiliary heat inputs, reduce the number of hours in which the solar field is operational. Nevertheless, the method presented herein shows two strategic advantages over raw experimental data for exploratory purposes. The first is related to the time required to monitor and register enough data of the system to accurately represent its behavior under different meteorological conditions. While the meteorological measurements are available, even for a short period of time, the operation of the system can be simulated in TRNSYS and obtain reliable results in terms of the thermal behavior. The second advantage is the possibility to isolate and observe nominal operational conditions of the system without the noise of external factors, such as maintenance issues, defective instrumentation and environmental factors, such as soiling.

The results show the strengths and shortcomings of an initial approach for performance prediction in a SHW system. The use of synthetic data allowed to isolate and study specific behaviors without temporal limitations and measurement uncertainties. Data collection is a vital step for developing deep learning-based models and the use of synthetic data also allows time-saving strategies to explore different alternatives before actual implementation in a real system. That approach is highly useful for conducting field assessments when the historical information is not available. However, its representativity of the thermal system is limited for anomalous scenarios. While further tuning of the model's hyperparameters is required, specialized architectures for time-series analysis, such as RNN and LSTM have proven to capture and predict temperature sequences better than DNN-based models. Moreover, other data-based metrics can be explored to assess the performance of these models by quantifying the uncertainty of the

experimental measurements, the simulation's results, and the model's predictions.

Declarations of interest

None.

Acknowledgments

The authors appreciate the financial support from CONICYT/FONDAP 15110019 “Solar Energy Research Center” SERC-Chile.

Nomenclature

ANFIS	Adaptive-Network-Based Fuzzy Inference System
ANN	Artificial Neural Network
BPTT	Backpropagation Through Time.
BRNN	Bidirectional Recurrent Neural Network
CNN	Convolutional Neural Network
DL	Deep Learning
DLSTM	Deep Long Short-Term Memory Recurrent Neural Network
DNN	Deep Neural Network
GP	Gaussian Process
IAM	Incidence Angle Modifier
LSTM	Long-Short Term Memory Recurrent Neural Network
ML	Machine Learning
MLP	Multi-Layer Perceptron
MSE	Mean Square Error
NN	Neural Network
PHM	Prognostics and Health Management
PV	Solar Photovoltaic technology
SVM	Support Vector Machine.
RMSE	Root Mean Squared Error
RBF	Radial Basis Function
RNN	Recurrent Neural Network
RUL	Remaining Useful Life.
SHW	Solar Hot Water system
TRNSYS	Transient System Simulation program

References

[1] International Energy Agency IEA, Technology roadmap: solar heating and cooling, Int. Energy Agency (2012) 50, https://doi.org/10.1007/SpringerReference_7300.
[2] H. He, T.P. Caudell, D.F. Menicucci, A.A. Mammoli, Application of Adaptive Resonance Theory neural networks to monitor solar hot water systems and detect existing or developing faults, Sol. Energy 86 (2012) 2318–2333, https://doi.org/10.1016/j.solener.2012.05.015.
[3] C. de Keizer, S. Kuethe, U. Jordan, K. Vajen, Simulation-based long-term fault detection for solar thermal systems, Sol. Energy 93 (2013) 109–120, https://doi.org/10.1016/j.solener.2013.03.023.
[4] G. Niu, Data-Driven Technology for Engineering Systems Health Management, Springer Singapore, Singapore, 2017, https://doi.org/10.1007/978-981-10-2032-2.
[5] S. Khan, T. Yairi, A review on the application of deep learning in system health management, Mech. Syst. Signal Process. 107 (2018) 241–265, https://doi.org/10.1016/j.ymssp.2017.11.024.
[6] H. Wang, B. Raj, On the origin of deep learning, J. Fish. Res. Board Can 13 (2017) 303–308. http://arxiv.org/abs/1702.07800.
[7] A. Ogawa, T. Hori, Error detection and accuracy estimation in automatic speech recognition using deep bidirectional recurrent neural networks, Speech Commun. 89 (2017) 70–83, https://doi.org/10.1016/j.specom.2017.02.009.
[8] S. Zheng, K. Ristovski, A. Farahat, C. Gupta, Long short-term memory network for remaining useful life estimation, 2017, IEEE Int. Conf. Prog. Health Manag. (2017) 88–95, https://doi.org/10.1109/ICPHM.2017.7998311.
[9] A. Alzahrani, P. Shamsi, C. Dagli, M. Ferdowsi, Solar irradiance forecasting using deep neural networks, Procedia Comput. Sci. 114 (2017) 304–313, https://doi.org/10.1016/j.procs.2017.09.045.
[10] G. Litjens, T. Kooi, B.E. Bejnordi, A.A.A. Setio, F. Ciompi, M. Ghafoorian, J.A.W.M. van der Laak, B. van Ginneken, C.I. Sánchez, A survey on deep learning in medical image analysis, Med. Image Anal. 42 (2017) 60–88, https://doi.org/10.1016/j.media.2017.07.005.
[11] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, J. Lin, Machinery health prognostics: a systematic review from data acquisition to RUL prediction, Mech. Syst. Signal Process. 104 (2018) 799–834, https://doi.org/10.1016/j.ymssp.2017.11.016.
[12] Y. Lui, G. Zhao, X. Peng, C. Hu, Lithium-ion battery remaining useful life prediction with long short-term memory recurrent neural network, Annu. Conf. Prog. Health Manag. Soc. (2017) 1–7.
[13] R. Zhao, R. Yan, J. Wang, K. Mao, Learning to monitor machine health with convolutional Bi-directional LSTM networks, Sensors (Switzerland) (2017) 17, https://doi.org/10.3390/s17020273.
[14] J. Korbicz, J.M. Koscienlny, Z. Kowalczuk, Fault Diagnosis Models, Artificial Intelligence, first ed., vol. 1, Springer-Verlag Berlin Heidelberg GmbH, Berlin, 2004 https://doi.org/10.1007/978-3-642-18615-8. Appliction vol. o.
[15] H.K. Ghritlahre, R.K. Prasad, Application of ANN technique to predict the performance of solar collector systems - a review, Renew. Sustain. Energy Rev. 84 (2018) 75–88, https://doi.org/10.1016/j.rser.2018.01.001.
[16] S.A. Kalogirou, S. Panteliou, A. Dentsoras, Artificial neural networks used for the performance prediction of a thermosiphon solar water heater, Renew. Energy 18 (1999) 87–99, https://doi.org/10.1016/S0960-1481(98)00787-3.
[17] W. Yaïci, E. Entchev, Performance prediction of a solar thermal energy system using artificial neural networks, Appl. Therm. Eng. 73 (2014) 1348–1359, https://doi.org/10.1016/J.APPLTHERMALENG.2014.07.040.
[18] E. Dikmen, M. Ayaz, H.H. Ezen, E.U. Küçüksille, A.Ş. Şahin, Estimation and optimization of thermal performance of evacuated tube solar collector system, Heat Mass Transf. Und Stoffuebertragung. 50 (2014) 711–719, https://doi.org/10.1007/s00231-013-1282-0.
[19] S.A. Kalogirou, Prediction of flat-plate collector performance parameters using artificial neural networks, Sol. Energy 80 (2006) 248–259, https://doi.org/10.1016/J.SOLENER.2005.03.003.
[20] C. Cetiner, F. Halici, H. Cacur, I. Taymaz, Generating hot water by solar energy and application of neural network, Appl. Therm. Eng. 25 (2005) 1337–1348, https://doi.org/10.1016/J.APPLTHERMALENG.2004.09.004.
[21] M. Souliotis, S. Kalogirou, Y. Tripanagnostopoulos, Modelling of an ICS solar water heater using artificial neural networks and TRNSYS, Renew. Energy 34 (2009) 1333–1339, https://doi.org/10.1016/J.RENENE.2008.09.007.
[22] M. Mohanraj, S. Jayaraj, C. Muraleedharan, Modeling of a direct expansion solar assisted heat pump using artificial neural networks, Int. J. Green Energy 5 (2008) 520–532, https://doi.org/10.1080/15435070802498499.
[23] Y. Varol, A. Koca, H.F. Oztop, E. Avci, Forecasting of thermal energy storage performance of Phase Change Material in a solar collector using soft computing techniques, Expert Syst. Appl. 37 (2010) 2724–2732, https://doi.org/10.1016/J.ESWA.2009.08.007.
[24] Z.C. Lipton, J. Berkowitz, C. Elkan, A critical review of recurrent neural networks for sequence learning, Proc. ACM Int. Conf. Multimed. MM '14 (2015) 675–678, https://doi.org/10.1145/2647868.2654889.
[25] J. Schmidhuber, Deep Learning in neural networks: an overview, Neural Netw. 61 (2015) 85–117, https://doi.org/10.1016/j.neunet.2014.09.003.
[26] Y. Wu, M. Yuan, S. Dong, L. Lin, Y. Liu, Remaining useful life estimation of engineered systems using vanilla LSTM neural networks, Neurocomputing 275 (2018) 167–179, https://doi.org/10.1016/j.neucom.2017.05.063.
[27] S. Srivastava, S. Lessmann, A comparative study of LSTM neural networks in forecasting day-ahead global horizontal irradiance with satellite data, Sol. Energy 162 (2018) 232–247, https://doi.org/10.1016/j.solener.2018.01.005.
[28] S. Leva, A. Dolara, F. Grimaccia, M. Mussetta, E. Ogliari, Analysis and validation of 24 hours ahead neural network forecasting of photovoltaic output power, Math. Comput. Simulat. 131 (2017) 88–100, https://doi.org/10.1016/j.matcom.2015.05.010.
[29] S. Kalogirou, S. Lalot, G. Florides, B. Desmet, Development of a neural network-based fault diagnostic system for solar thermal applications, Sol. Energy 82 (2008) 164–172, https://doi.org/10.1016/j.solener.2007.06.010.
[30] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, in: Readings Cogn. Sci, Elsevier, 1988, pp. 399–421, https://doi.org/10.1016/B978-1-4832-1446-7.50035-2.
[31] P.J. Werbos, Backpropagation through time: what it does and how to do it, Proc. IEEE 78 (1990) 1550–1560, https://doi.org/10.1109/5.58337.
[32] Y. Bengio, P. Simard, P. Frasconi, Learning long term dependencies with gradient descent is difficult, IEEE Trans. Neural Netw. 5 (1994) 157–166, https://doi.org/10.1109/72.279181.
[33] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735.
[34] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Trans. Signal Process. 45 (1997) 2673–2681, https://doi.org/10.1109/78.650093.
[35] A. Graves, A.R. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc, 2013, pp. 6645–6649, https://doi.org/10.1109/ICASSP.2013.6638947.
[36] Y. Levine, O. Sharir, A. Ziv, A. Shashua, Iclr2018, On the Long-Term Memory of Deep Recurrent Networks, 2017, pp. 1–23, http://arxiv.org/abs/1710.09431.
[37] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, AIP Conf. Proc. 1631 (2014) 58–62, https://doi.org/10.1063/1.4902458.
[38] T. Tijmen, H. Geoffrey E, Lecture 6.5- RMSprop: divide the gradient by a running average of its recent magnitude, COURSERA Neural Netw. Mach. Learn. 42 (2012).
[39] N. Reimers, I. Gurevych, Reporting score distributions makes a difference: performance study of LSTM-networks for sequence tagging, in: Proc. 2017

Conf. Empir. Methods Nat. Lang. Process., 2017, pp. 338–348, https://doi.org/10.18653/v1/D17-1035.
[40] S. Kalogirou, Solar Energy Engineering Processes and Systems, Second, Elsevier Inc., 2009, https://doi.org/10.1016/B978-0-12-374501-9.00014-5.
[41] John A. Duffie, W.A. Beckman, Solar Engineering of Thermal Processes Solar Engineering, Fourth, John Wiley & Sons, Inc., Hoboken, New Jersey, 2013, https://doi.org/10.1002/9781118671603.fmatter.
[42] E. Ruiz, P.J. Martínez, Analysis of an open-air swimming pool solar heating system by using an experimentally validated TRNSYS model, Sol. Energy 84 (2010) 116–123, https://doi.org/10.1016/j.solener.2009.10.015.
[43] L.M. Ayompe, A. Duffy, S.J. McCormack, M. Conlon, Validated TRNSYS model for forced circulation solar water heating systems with flat plate and heat pipe evacuated tube collectors, Appl. Therm. Eng. 31 (2011) 1536–1542, https://doi.org/10.1016/j.applthermaleng.2011.01.046.
[44] F. Bava, S. Furbo, Development and validation of a detailed TRNSYS-Matlab model for large solar collector fields for district heating applications, Energy 135 (2017) 698–708, https://doi.org/10.1016/j.energy.2017.06.146.
[45] J.F. Chen, Y.J. Dai, R.Z. Wang, Experimental and analytical study on an air-cooled single effect LiBr-H2O absorption chiller driven by evacuated glass tube solar collector for cooling application in residential buildings, Sol. Energy 151 (2017) 110–118, https://doi.org/10.1016/J.SOLENER.2017.05.029.
[46] J. Burch, C. Christensen, Towards development of an algorithm for mains water temperature, in: InterSolar 2007 Conf, 2007, pp. 5–10.
[47] S.A. Kalogirou, R. Agathokleous, G. Barone, A. Buonomano, C. Forzano, A. Palombo, Development and validation of a new TRNSYS Type for thermosiphon flat-plate solar thermal collectors: energy and economic optimization for hot water production in different climates, Renew. Energy 136 (2019) 632–644, https://doi.org/10.1016/j.renene.2018.12.086.
[48] J.E. Griffin, P.J. Brown, Some priors for sparse regression modelling, Bayesian Anal 8 (2013) 691–702, https://doi.org/10.1214/13-BA827.
[49] M. Guermoui, F. Melgani, C. Danilo, Multi-step ahead forecasting of daily global and direct solar radiation: a review and case study of Ghardaia region, J. Clean. Prod. 201 (2018) 716–734, https://doi.org/10.1016/j.jclepro.2018.08.006.
[50] P. Lauret, C. Voyant, T. Soubdhan, M. David, P. Poggi, A benchmarking of machine learning techniques for solar radiation forecasting in an insular context, Sol. Energy 112 (2015) 446–457, https://doi.org/10.1016/j.solener.2014.12.014.