Comparative analysis of neural networks techniques to forecast Global Horizontal Irradiance

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 25, 2021, accepted August 15, 2021, date of publication September 3, 2021, date of current version September 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3110167

Comparative Analysis of Neural Networks Techniques to Forecast Global Horizontal Irradiance

ALESSANDRO ALIBERTI[1], (Member, IEEE), DANIELE FUCINI[1], LORENZO BOTTACCIOLI[1,3], (Member, IEEE), ENRICO MACII[2], (Fellow, IEEE), ANDREA ACQUAVIVA[4], (Member, IEEE), AND EDOARDO PATTI[1,3], (Member, IEEE)

[1]Department of Control and Computer Engineering, Politecnico di Torino, 10129 Turin, Italy
[2]Interuniversity Department of Regional and Urban Studies and Planning, Politecnico di Torino, 10129 Turin, Italy
[3]Energy Center Lab, Politecnico di Torino, 10129 Turin, Italy
[4]Department of Electrical, Electronic, and Information Engineering "Guglielmo Marconi," Università di Bologna, 40126 Bologna, Italy

Corresponding author: Alessandro Aliberti (alessandro.aliberti@polito.it)

ABSTRACT Due to the continuous increasing importance of renewable energy sources as an alternative to fossil fuels, to contrast air pollution and global warming, the prediction of Global Horizontal Irradiation (GHI), one of the main parameters determining solar energy production of photovoltaic systems, represents an attractive topic nowadays. Solar irradiance is determined by deterministic factors (i.e. the position of the sun) and stochastic factors (i.e. the presence of clouds). Since the stochastic element is difficult to model, this problem can benefit from machine learning techniques, like artificial neural networks. This work proposes a methodology to forecast GHI in short- (i.e. from 15 min to 60 min) and mid-term (i.e. from 60 to 120 min) time horizons. For this purpose, we designed, optimised and compared four neural network architectures for time-series forecasting, respectively based on: i) Non-Linear Autoregressive, ii) Feed-Forward, iii) Long Short-Term Memory and iv) Echo State Network. The original data-set, consisting of GHI values sampled every 15min, has been pre-processed by applying different filtering techniques. Our results analysis compares the performance of the proposed neural networks identifying the best in terms of error rate and forecast horizon. This analysis highlights that the clear-sky index results the preferred filtering technique by giving greatly improvements in data-set pre-processing, and Echo State Network gives best accuracy results.

INDEX TERMS Solar radiation forecast, artificial neural networks, LSTM, ECHO, photovoltaic system, energy forecast, renewable energy.

I. INTRODUCTION

Nowadays, renewable energy is a very hustling research area. Finding viable, clean energy sources to replace fossil fuels, or at least to significantly decrease their usage in short to medium term, has become an extremely critical goal to achieve. On the one hand, air pollution, of which fossil fuels are a major contributor, is causing a real health crisis [1]. According to the World Health Organisation (WHO), air pollution is responsible for 7 million deaths every year, and 91% of the world population lives in places where air quality exceeds the limits mandated by the WHO itself [2]. On the other hand, greenhouse gas emissions from fossil fuels are also one of the major drivers of anthropogenic climate changes. According to a 2018 special report by the Intergovernmental Panel on Climate Change (IPCC), immediate action must be taken to limit the increase in global temperature to 1.5 °C and avoid the worst consequences of global warming [3]. For these reasons, renewable energy sources (RES) will have a key role in the future of our society. Among them, an important part is played by solar energy, which can be used to produce electricity exploiting photovoltaic (PV) systems. Nowadays, PV systems are rapidly spreading in our cities with strong economic, social and

The associate editor coordinating the review of this manuscript and approving it for publication was Baozhen Yao.

environmental impacts [4]. At the same time, the scientific community is researching new models and new optimisation methods to better manage these resources.

In a PV system, the power output of a PV panel is directly proportional to the solar irradiance, which in turn depends on various factors (e.g. latitude, season, and sky conditions). There are different components of solar irradiance, but the most important one for PV power generation is Global Horizontal Irradiance (GHI), which is the total irradiance on a horizontal surface [5]. GHI is related to Direct Normal Irradiance (DNI), which is the irradiance on a surface perpendicular to the sun, and Diffuse Horizontal Irradiance (DHI), which is the radiation from light scattered by the atmosphere.

To optimise smart grid operations and match power production, distribution and consumption efficiently and reliably, it is needed to know in advance the amount of energy produced by power plants [6], as well as energy consumption. However, one of the issues posed by some of the most popular renewable energy sources, like wind and solar energy, is their non-dispatchable and intermittent nature. A dispatchable energy source can be turned on and off when needed in a short amount of time, according to needs. This is not true for PV power systems. The sun only shines for a limited amount of hours during the day, depending on latitude and season, and the irradiance is also affected by clouds.

When integrating non-dispatchable Renewable Energy Sources (RES) into existing power grids, this intrinsic variability must be taken into account, particularly when the share of energy from these types of sources increases [7]. An exciting possibility is to integrate RES using smart grid technologies. A traditional power grid is centralised and implements a one-way communication, where the power is sent from the power plant to customers. In a smart grid, on the other hand, the process becomes distributed, and the consumer can also be an active user, giving feedbacks on electrical use that allow the grid to tune itself to provide better performance and guarantee better reliability. Examples of the application of smart grid management are Demand-Response (DR) [8], Demand Side Management (DSM) [9] and mobility based on electric vehicles [10]. DR refers to the changes in electricity consumption patterns by the user in response to fluctuations in power production by renewable energy sources and grid requirements, as well as for economic reasons like changes in the price of electricity. Instead, DSM refers to a set of actions aimed at efficiently managing the consumption of a grid, in order to reduce the costs incurred for the supply of electricity and for general system charges.

As stated above, the most popular RES are non-dispatchable and intermittent by nature. These features introduce problems in grid stability and efficiency which lead to limitations on the amount of these resources which can be effectively added to the grid. Thus, generation output by RES is driven by environmental and meteorological conditions and cannot respond to changes in demand. Solar energy, in particular, is determined both by deterministic (e.g. latitude, day of the year, hour of the day) and stochastic factors (e.g. effects of the atmosphere and weather conditions like cloud coverage). Consequently, the scientific community is pushing for innovation and optimization purposes in this scenario. Indeed, many studies are currently focused on new and optimized methods to forecast the Global Horizontal Irradiance (GHI), which strongly influences Photovoltaic (PV) production [11]–[14]. In this context, the main challenge is, therefore, to find a methodology to predict the power generated by a photovoltaic system accurately. Since PV energy generation is highly correlated to solar irradiance, it makes sense concentrate on predicting the latter, in particular GHI, and then use these predictions to calculate the expected energy production. Indeed, accurate forecasting of GHI is crucial to unlock the development of novel control strategies for smart grid management, e.g. DR and DSM, that aim at mitigating undesirable fluctuations introduced by RES (e.g. PV systems). For example in this scenario, a photovoltaic simulator such as the one proposed in [15], [16] could be employed using GHI predictions as a system input to estimate energy production of PV systems for the next short- and mid-term time-horizons. Thanks to DR and DSM policies, the energy demand can dynamically respond to changes in the energy generation, overturning the current paradigm where the generation responds to changes in demand.

Since solar irradiance is a physical phenomenon, a possibility could be to develop a physical model. The main problem with this approach is its complexity, mainly when modelling the stochastic atmospheric phenomena that determine the measured GHI on the surface. A more straightforward approach is based on time-series forecasting. The idea is to use previous values of the time-series we are interested in predicting, and/or one or more related series, one or more future values. Several studies were proposed in the literature to find physical and mathematical models to estimate and forecast solar radiation. Classical linear time-series models have been widely used [17]. Simple statistical models, for example, can be used but they might give sub-optimal results because solar irradiance is a complex nonlinear time-series. However, these studies have highlighted that these methodologies are not sufficient in the analysis and prediction of solar radiation due to the non-stationary and non-linearity characteristics [18]. To overcome these limits, a more robust approach is based on machine learning. One of the most used and studied applications of machine learning is that of artificial neural networks [19], [20].

In this paper, we propose a methodology for short- (i.e. from 15 min to 60 min) and mid-term (i.e. from 60 to 120 min) GHI forecast, with 15 min time-steps, exploiting state-of-the-art neural networks models in time-series scenario. The main goal is to obtaining accurate predictions to use as input for PV simulators, as demonstrated in our previous works [21], [22]. More accurate inputs allow even more robust PV simulations [16] and this leads to better analysis and management of the energy produced in operational contexts such as DR and DSM. In detail, we specifically designed and deeply optimised four architectures

respectively based on a Non-Linear Autoregressive (NAR), a Feed-Forward (FFNN), a Long Short-Term Memory (LSTM) and an Echo State Network (ESN) as they represent the most promising and suitable solutions in time-series predictions. In addition, we apply different data-set pre-processing and filtering techniques to identify which of them has better performances by comparing the prediction results of the different architectures. We repeated training, optimisation and test phases of all our neural network architectures for each pre-processing technique applied to the original data-set consisting of a time-series of GHI values sampled every 15 min. The first pre-processing technique aims at removing some sampling error in the original GHI data-set (hereinafter referred to as raw GHI data-set). Then, the second pre-processing technique applies a Tikhonov regularisation to smooth the time-series making the training phase easier. Finally, the third pre-processing technique is based on the clear-sky index, which is the ratio of the measured irradiance and the clear-sky irradiance. In this case, GHI samples are converted into clear-sky index values to remove the seasonal trend of the time-series. Both Tikhonov regularisation and clear-sky index are applied on the raw GHI data-set.

The rest of the paper is organised as follows. Section II performs a review of the scientific literature on the topic of solar irradiance forecasting. Section III illustrates the methodology proposed in this work. Section IV presents the experimental results by comparing the performance of the three approaches described. Finally, Section V discusses the concluding remark.

II. RELATED WORKS

The literature encloses several forecasting models for solar irradiance and PV power. Generally these models are divided into four main categories [20], [23] i) statistical models; ii) cloud imagery-based models; iii) numerical weather predictions (NWP) models and iv) hybrid models. As introduced in Section I, statistical models use previous values of the solar irradiance or PV power time-series to forecast the next values. For this reason, they represent the category of our interest to which we are inspired and compared. They, in turn, can be divided into linear and nonlinear models. However, according to [24], limitation of linear models is that they cannot take into account the non-linearity of many real-life time-series, including solar irradiance. As a result, according to the purpose of our work, this section investigates these models by highlighting merits and weaknesses.

Generally, linear methods represent the simplest forecasting model, often used as reference to evaluate other more complex [12]. They are based on the simple assumption that the forecasted value of the time-series is the same as the current value. One of the simplest linear models is the autoregressive (AR) [25]. A slightly more complex model is the autoregressive moving average (ARMA), which combines autoregressive and moving-average components [25]. The ARMA model can be extended including exogenous inputs (ARMAX). AR and ARMA models can be used to forecast stationary time-series. In a stationary process, the mean and the variance remain constant over time [17]. But processes like solar irradiance are non-stationary, so they must be transformed into stationary time-series, or different models should be developed. The ARIMA model (autoregressive integrated moving average) can be used for non-stationary time-series forecasting. Reikard [26] shows that ARIMA can give good short-term solar irradiance forecasting results. His experiments evaluate forecasting horizons of 5, 15, 30 and 60 min, and the ARIMA model not only outperforms simple AR models in all cases, but it also performs better than feed-forward artificial neural networks except for the shortest time horizon. This might be caused by the difficulty to train ANNs causing them to reach only a local optimum.

However, a limitation of linear models is that they cannot take into account the non-linearity of many real-life time-series, including solar irradiance. For this reason, nonlinear techniques for time-series forecasting have become very popular, and they have been extensively used for solar irradiance prediction [27]. Martin *et al.* [28] forecast half daily values of solar irradiance, i.e. ''accumulated hourly global solar irradiance from solar raise to solar noon and from noon until dawn for each day''. Since this time-series is non-stationary, two transformations are proposed, the clearness index, which is the ratio between the solar irradiance measured at ground level and the extraterrestrial irradiance, and the lost component, which is the difference of the same quantities. Different feed-forward neural networks configurations, in terms of the number of hidden layers, neurons, inputs, are tested, and the best one is selected for each weather station where the prediction model is evaluated. The results show that ANNs improve the forecasting accuracy of the reference persistence model and outperform a simpler linear AR model. Pedro and Coimbra [29] evaluate different PV power forecasting techniques. Among them, a feed forward neural network with one hidden layer of 20 neurons. The network has 13 inputs, which are fed with 13 previous values of the time-series. The forecasting is evaluated for 1 h and 2 h ahead. The neural network-based technique outperforms the other models evaluated in the study. Lauret *et al.* [30] instead compare several machine learning techniques. Simple persistence and AR models are used as reference. The GHI time-series are pre-processed by transforming them into clear-sky index series. For the shortest time horizons, the machine learning techniques perform better than the reference models for unstable conditions, while for clear-sky conditions, the AR model is also accurate. For longer time horizons, though, the machine learning models, including the feed-forward neural network, clearly outperform persistence and linear AR techniques. Rana *et al.* [31] use an ensemble of neural networks for short-term (from 5 to 60 min ahead) PV power forecasting for both the univariate and multivariate case. They test multiple ensembles E_i, where each ANN in the ensemble has i neurons in the hidden layer. Each ensemble is made of 20 networks. The final forecasting result is selected by taking the median of the 20 predictions. This

method obtains better results than the reference persistence model and another machine learning technique called support vector machine (SVM). McCandless *et al.* [32] develop a cloud regime-dependent forecasting technique based on feed-forward neural networks. Instead of using a single ''global'' neural network, different ANNs are trained and used for each cloud regime. To determine the cloud regime, a k-means algorithm is applied to the clearness index time-series. For the shortest time horizons (15 min) ANNs does not improve upon the reference persistence model, except for the most unstable sky conditions. For longer horizons (from 60 to 180 min), however, the ANNs outperform persistence. It is also shown that the regime-dependent forecasting always gives better results than a single global ANN. Monjoly *et al.* [33] use different multi-scale decomposition techniques to pre-process the GHI time-series, after transforming it into the clear-sky index. The different time scale components are then forecasted separately using separate ANNs or using a hybrid AR-ANN model. The results show that multi-scale decomposition significantly improves forecasting results, both using ANNs and the hybrid model. Bouzgou and Gueymard [13] design a Wrapper Mutual Information Methodology (WMIM) optimization approach by exploiting Extreme Learning Machine (ELM) regression technique to i) investigate the effect of the mutual information measure between the historical variables and the targeted future GHI value, and ii) select the best possible combination of historical variables from the existing time series. Experimental results highlight that the ELM model, combined with WMIM, provides the same performances of the more conventional Multi Layer Perceptron (MLP) but lower computing time.

On the other hand, another widely used neural network is the recurrent neural network, where feedback connections are added. The outputs of these networks also depend on their current state (memory), not only on the current inputs. This behaviour makes them very suitable for time-series analysis and forecasting. Among these, there are Long Short-Term Memory networks. These are recurrent neural networks using particular units (LSTM units) as nodes. These units can remember values for an arbitrarily long amount of time, and this behaviour makes them very suitable for time-series forecasting. Alzahrani *et al.* [34] use a deep (i.e. with more than one hidden layer) recurrent neural network with LSTM units for short-term forecasting of solar irradiance. The input time-series is sampled at a very high frequency (100 Hz). The advantage of this high-resolution time-series is that it can capture fast fluctuations. The results show that the deep LSTM has better accuracy than the reference feed-forward neural network. In [35] the authors exploit LSTM networks for 1 h ahead PV power forecasting. Different LSTM models are evaluated, and the best one, LSTM for regression with time steps, is selected and compared with other forecasting techniques, including multiple linear regression and feed-forward neural networks. The LSTM is shown to give more accurate results than the other models. Srivastava and Lessmann [36] compare LSTM with other established forecasting techniques in day-ahead GHI forecasting. They use satellite-derived GHI values and other atmospheric variables as inputs. In contrast to the majority of the studies in the literature, many different locations in several countries with different climates are taken into account, which makes it possible to assess the validity of the proposed model in different conditions. The LSTM-based approach is compared to a simple persistence model, a feed-forward neural network model and another machine learning model called ''Gradient Boosting Regression'' (GBR). The results show the superior performance of the LSTM compared to the other methods. Finally, Zang *et al.* [14] addresses short-term solar irradiance forecasting by exploiting spatio-temporal correlation model based on deep learning. In detail, they first apply a convolutional neural network (CNN) to extract spatial features. Then, they apply a LSTM network to extract temporal features from historical data. In this way, they successfully obtain spatiotemporal correlations to predict global horizontal irradiance one hour in the future.

Another interesting recurrent architecture is the Echo State Network. An ESN has a sparsely connected hidden layer, called ''reservoir'', with fixed connections and weights. The only weights that are learnt are those of the output connections. This property makes these networks easier to train compared to other recurrent architectures. Kmet and Kmetova [37] used an ESN for 24 h ahead solar irradiance forecasting, using the actual mean hourly values of irradiance and other meteorological variables like humidity and air temperature. The inputs of the network consist of 24 hourly values of the selected variables for the present day, and the outputs are the irradiance forecasting for the next day. The paper shows that this approach gives good results. In a previous study, on the other hand, Ruffing and Venayagamoorthy [38] found that in a real-world application results of an ESN-based solar irradiance forecasting model were not very promising. In a related field, Deihimi and Showkati [39] used an ESN for 1 h and 24 h ahead electric load forecasting. In this case, the results showed that the ESN has a good generalisation capability and can give very accurate results.

Table 1 summarizes and highlights the key features of the analysed literature solutions based on machine learning to predict GHI. All these works represent real milestones, however, as can easily be seen i) often a large number of hard-to-find data are used to predict the GHI trends (e.g. Surface Weather Observation, satellite-derived GHI, atmospheric variables, meteorological measurements); ii) very few works exploit the potential of filtering techniques to pre-process the original data-set with the aim of using it as input for the neural models (i.e. a data-set characterized by a trend that is easier to learn allows the implementation of leaner neural architectures), iii) few works are compatible with DR and DSM scenarios (i.e. 15 min time-steps) and iv) hybrid or assembled methodologies, rather than a single global architecture, are almost always used which implies a greater use of computational resources.

TABLE 1. Comparison with the major and latest literature review.

Solution	Input Dataset	Dataset Preprocessing Technique	Machine Learning Model	Sampling Time	Forecast Horizon
Proposed methodology	GHI trends	Tikhonov regularization; Clear-sky Index	NAR; FFNN; LSTM or ESN	15 min	from 15 min to 120 min
Bouzgou et al. [13]	GHI trends	x	WMIM ELM	5 min; 1 h	5-10-15-20-25-30 min; 1-2-3 h
Zang et al. [14]	Two-dimensional matrix composed of several meteorological parameter	x	Hybrid CNN-LSTM	1 h	1 h
Reikard et al. [26]	GHI trends	x	ARIMA	5 min; 15 min; 30 min; 60 min	5-15-30-60 min; 1-2-3-4 h
Sfetsos et al. [27]	GHI trends	x	FFNN; Radial basis function; Elman recurrent; ANFIS models	1 h	1 h
Martín et al. [28]	GHI trends	Clear-sky Index; Lost Component	ANN; ANFIS	half day	half day for the next 3 days
Pedro et al. [29]	PV trends	x	ARIMA; kNNs; ANN; GAs/ANN	1 h	1-2 h
Lauret et a. [30]	GHI trends; Weather Observation	Clear-sky Index	AR; NN; GPs; SVM	average of 1 h (raw GHI is sampled every 6 sec)	1-2-3-4-5-6 h
Rana et al. [31]	PV trends; GHI; temperature; humidity; cloud cover; wind speed;	x	ANN ensemble; SVR	average of 5 min (raw GHI is sampled every 1 min)	from 5 min to 1 h
McCandless et al. [32]	GHI trends, Surface Weather Observation; Satellite Data	x	Different FFNN for each cloud regime	average of 15 min (raw GHI is sampled every 1 min)	15 min; 60 min; 120 min; 180 min
Monjoly et al. [33]	GHI trends	Clear-sky Index	Hybrid structure based on multiscale decomposition	1 h	1 h
Abdel-Nasseret et al. [35]	PV trends	x	LSTM	1 h	1 h
Srivastava et al. [36]	Satellite-derived GHI; Atmospheric variables	x	Global ANN	not applicable	1 day
Ruffing et al. [38]	GHI trends	x	ESN	average of 1 min (raw GHI is sampled every 3 sec)	from 30 min to 270 min
Deihimi et al.[39]	Load and Temperature	x	ESN	1 h	from 1 h to 24 h

With respect to presented literature solutions to forecast solar radiation, our work proposes an optimised methodology for the short- (i.e. up to 60 min) and mid-term (i.e. from 60 to 120 min) GHI forecast exploiting and comparing four neural networks specifically designed and optimised. We applied three different data-set pre-processing and filtering techniques to identify which of them has better performances by comparing the prediction results of the different architectures. As a first analysis, we gave in input to our neural networks the raw GHI data-set where we applied some basic filters to clean the data-set itself by removing possible errors, such as lack of data or storage error due to sensor sampling. Then, as a second analysis, we applied the Tikhonov regularisation technique to the very same raw data-set, which smooths the time-series trend-making easier the training of our neural networks. Finally, as a third analysis, we converted the raw data-set, consisting of GHI samples, into clear-sky index values, thus removing seasonal trends of the time-series. For a fair comparison, all the data-sets at our disposal are used as input on all the best neural architectures identified experimentally. The use of pre-processing techniques, together with the capillary optimisation of neural structures, allows us to increase the prediction time horizon with an acceptable error rate. Moreover, compared to state-of-art solutions, the optimisation of neural architectures from specifically transformed data-sets allows us to obtain leaner structures at the computational level without affecting the prediction accuracy.

To overcome the state-of-the-art limitations and to highlights our scientific contribution, we propose a novel and reliable methodology to forecast GHI in short- and mid-term exploiting neural networks models suitable for time-series analyses. Through this research work we aim at i) obtaining accurate GHI predictions to be compliant with smart grid management, as previously discussed; ii) obtaining an optimized methodology for the short- (i.e. from 15 up to 60 min.) and mid-term (i.e. from 60 to 120 min.) GHI forecast and, iii) assessing the impact of different pre-processing techniques in relation to different state-of-art neural models in a GHI forecasting scenario.

For this purpose, we optimized the methodology right from the pre-processing of the data-set until the choice of the best performing neural model operating with time-series, suitable in this operative context. Consequently, we optimized the methodology right from the pre-processing of the data-set until the choice of the best performing neural model operating with time-series, suitable in this operative context. In brief, the most important operations we performed are:

- exploiting three different data-set pre-processing and filtering techniques (i.e. Raw data-set, Tikhonov, and Clear-sky index). To the best of our knowledge, we used the Tikhonov regularisation for the first time in the field of energy forecasting;
- designing, deeply optimizing and comparing the performance of four neural network architectures based on i) Non-linear Autoregressive Neural Network, ii) Feed Forward Neural Network, iii) Long Short-Term Memory and iv) Echo State Neural Network. In the literature, they represent the best solution for time-series analysis. Our objective, as mentioned above, is to design and optimise highly-specialised models for the GHI forecast in the short- and mid-term (i.e. from next 15 to 120 min). This allows more accurate GHI forecasts ready to be used as input to PV simulators (e.g. [15], [16]) to unlock new energy scenarios and policies, such as DR, DSM and electric vehicles energy management;
- mixing together the investigated pre-processing techniques and the neural models. The results analysis shows that the clear-sky index approach is the most successful, giving the most accurate results, particularly for mid-term predictions and the Echo State Network results

the neural architecture that best performs in terms of prediction accuracy.

III. METHODOLOGY

Generally, when approaching applications based on neural networks, it is common practice to follow a strict and precise procedure. In [40], the author provides a comprehensive procedure to identify a dynamical system. This procedure consists of four steps as detailed in Figure 1.

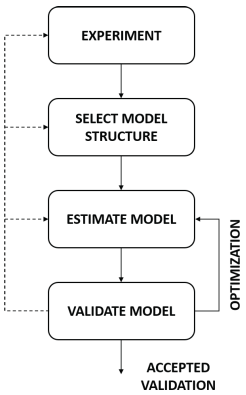


FIGURE 1. System identification procedure.

The first step, *Experiment*, corresponds to the problem analysis. Generally, this step approaches the problem by identifying the main characteristics and expected goals to start collecting a reliable data-set. Then, the available data must be divided into two different data-sets: Training- and Test-set, respectively. These data-sets are used in the training and test phases of the neural network. In this work, we propose a methodology for short- and mid-term GHI forecasts. Therefore, we exploit 5 years of GHI measurement sampled every 15 minutes for training and 1 year for test (see Section III-A). We also applied three different pre-processing techniques to the input data-set to obtain better performances in terms of prediction accuracy and computation level. The first necessary pre-processing technique aims to avoid missing and inconsistent measurements, i.e. removing errors due to sensor sampling, obtaining the so-called raw Data-set. Then, we applied two alternative filters to resulting raw data-set: i) Tikhonov regularization and ii) clear-sky index conversion. The Tikhonov regularization technique smooths the time-series trend avoiding possible spikes. While converting GHI time-series into clear-sky index values removes all possible seasonal trends. In both cases, we trained all our neural networks again, and then we optimized their architectures. Each of the three resulting data-sets has been split into Training- and Test-set (i.e. data never used during the training).

The second step, *Model Structure Selection*, allows generally to identify the correct neural network model to use. This step is crucial because the use of the wrong neural model can affect the expected results [41]. We design and compare four state-of-art neural networks in time-series scenario: i) Nonlinear Autoregressive Neural Network (NAR), ii) Feed-Forward Neural Network (FFNN), iii) Long Short-Term Memory Network (LSTM) and iv) Echo State Network (ESN) (see Section III-B). Moreover, we consider also both one-step and multiple steps predictions for each selected model. Making a one-step prediction means taking GHI samples at times $t, t-1, t-2, \ldots, t-n$ to predict GHI at time $t+1$, i.e. in 15 min, since that is the distance of two consecutive samples in the data-set. Instead, for multi-step prediction, we evaluate two different techniques, iterative and multi-output, respectively. In the iterative approach, the artificial neural network has a single output, so it can only predict one-step ahead. For subsequent steps, the predicted value for time $t+1$ is used as one of the inputs for the prediction at time $t+2$, and so on. In the multi-output approach the network has n output nodes, giving the prediction for $t+1, t+2, \ldots, t+n$ in parallel.

Once the network model is identified, the network is first implemented and then trained. This step is called *Model Estimation*. In time series scenario, training a neural network is needed to provide i) the vector containing desired output data, ii) the number of regressors to define the prediction, iii) the vector containing the weights of both input-to-hidden and hidden-to-output layers and iv) the data structure containing the parameters associated with the selected training algorithm. Finally, the training phase produces a training error, which represents the network performance index [42]. To fair compare their performances in forecasting GHI, we trained all our neural networks with the very same Training-set, and we tested them with the very same Test-set.

The *Model Validation* step validates the trained network. Generally, validating a network allows evaluating its capabilities. In time-series predictions, the most common validation method consists of analyzing the residuals (i.e. prediction errors) by cross-validating the test set. This method allows performing a set of tests, including also the auto-correlation function of the residuals and the cross-correlation function between controls and residuals. This analysis provides the test error, that is an index considered as a generalization of the error estimation. This index should not be too high compared to training error. If this happens, the network could overfit the training set. Generally, if the network overfits the training set, the selected model structure contains too many weights. The structure is then subjected to the *Network optimization and final validation*. The process requires to return in the *Estimate Model* step to change and redefine some structural parameters by optimizing the whole architecture. For this purpose, the unnecessary weight must be pruned. Consequently, once the new weights are given, the network architecture must be re-validated. This is an iterative process as highlighted in Figure 1 (see dotted arrows). Generally, leading back to model estimation means that the problem has several local minima and finding the global minimum is not easy. Leading back to model structure selection means that the neural structure does not fit for purpose. Indeed, this is usually oversized. Thus, it is common to apply some

pruning techniques. Generally, an initial model structure that is large enough to describe the system is determinate, and it is then reduced gradually until the optima structure is achieved. Finally, leading back to the experiment phase implies that certain regimes of the operating range are not reflected in the data-set; thus, additional tests are needed to acquire more information about the missing regimes. The whole process of model estimation, validation and optimisation is reported in Section III-B and Section IV where the best neural structure identified for each model and the prediction results are described respectively.

Figure 2 summarizes the overall process. The rest of this section will describe in-depth the proposed methodology.

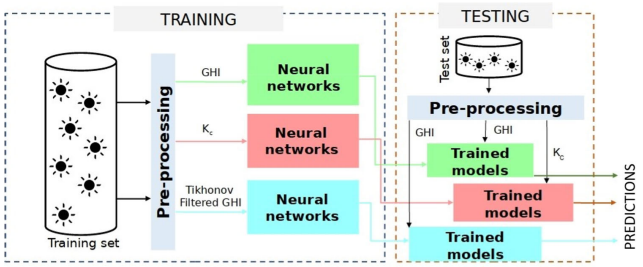


FIGURE 2. Methodology outline.

A. DATA-SET AND DATA TRANSFORMATION

We exploit a data-set of 6 years GHI measurements, sampled every 15 minutes from a meteorological station in Turin [21]. This data-set has been subjected to necessary some pre-processing. First, GHI can never be negative, so any negative values were set to 0. Then, comparing the raw GHI values in the data-set with the generated clear-sky values (I_{cs}), some of them were higher than the corresponding I_{cs}. This is probably due to some sampling error by the sensor or, in some cases, to some short-term cloud enhancement effects [43]. Since most of these peaks usually occur when the solar zenith angle is big [44], which is also when the reliability of the sensor is lower, it was decided to filter these anomalous peaks, so for each $GHI > I_{cs}$, GHI was set equal to I_{cs}.

After these basic pre-processing, we divide the data-set into Training- and a Test-set as follows: 5 years for Training (2010-2014) and 1 year for Test (2015), with 175 296 and 35 038 samples, respectively. Then, we exploit about the 10% of the Training-set for validation purpose, in the training and model validation phase.

Furthermore, we introduced two data-set transformations to improve prediction performance: i) Tikhonov regularization and ii) clear-sky index.

1) TIKHONOV REGULARIZATION

Generally, raw GHI data is characterized by many sudden peaks and variations. For this reason, we decide to try smoothing the data to make the training of the neural networks easier. The disadvantage of this approach is that some information about the variability of the phenomenon will necessarily be lost. However, the potential benefit is that the networks could more easily follow the trends in the data, particularly on medium or long term predictions. This is a trade-off, meaning that the choice of this approach might depend on the required prediction horizon and the application for which the predictions are needed. Then to smooth the original data, we exploit the Tikhonov regularization [45]. This technique is used for time-series analysis and predictions in other domains, like glucose level prediction [46].

The filtered signal is given by:

$$\hat{y} = U_d \omega \tag{1}$$

In Equation 1, ω is the N-dimensional first derivative of the input signal, while U_d is the integral operator matrix (Equation 2).

$$U_d = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 1 & 0 & \dots & 0 & 0 & 0 \\ 1 & 1 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & \dots & 1 & 0 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 & 1 \end{bmatrix} \tag{2}$$

To calculate ω, the function $f(\omega)$ (Equation 3) needs to be minimized.

$$f(\omega) = \|y - U_d\omega\|^2 + \lambda_d^2 \|L_d \omega\|^2 \tag{3}$$

In Equation 3 L_d is the second derivative operator matrix, while λ_d is the regularization parameter, set to 3000, in accordance to [46]. An example of the results of the filtering can be seen in Figure 3. This figure shows twelve consecutive days in the year on which the three main weather conditions occur: i) sunny day, ii) cloudy day, and iii) rainy day.

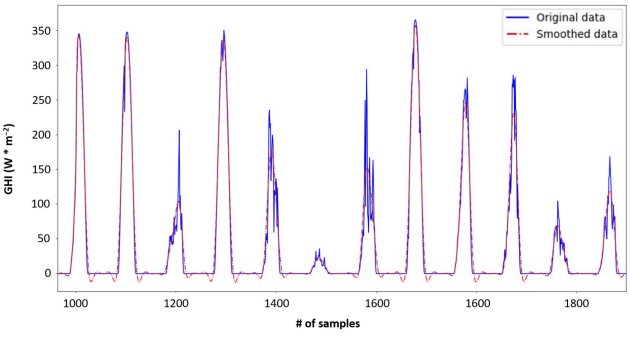


FIGURE 3. Original data vs. smoothed data.

Filtered data will be used for training the networks. Since the filter eliminates some peaks and spikes in the GHI data, giving a smoother signal, it should be easier for the networks to approximate it, potentially increasing the generalization capability of the model. Once the networks are trained, the original unfiltered GHI data will be used for testing, as shown in Figure 2.

2) CLEAR-SKY INDEX

The solar irradiance time-series exhibits a seasonality component and is therefore non-stationary. Some authors assert that neural network can work well even with non-stationary time-series, given enough training data [27]. Others prefer to transform the solar irradiance into a stationary series [29], [33]. For a stationary series, statistical properties like mean and variance are constant in time. This should make it easier to predict than a non-stationary series. For this purpose, the clear-sky index (K_c) is used in literature. K_c is the ratio between the expected irradiance under clear sky conditions and the measured one. For this work, after evaluating the performance of the network with the original GHI data, we chose to repeat the experiments using the clear-sky index, defined in Equation 4, where I_m is the measured irradiance, and I_{cs} is the calculated clear-sky irradiance.

$$K_c = \frac{I_m}{I_{cs}} \tag{4}$$

The K_c series was then used to train the networks. Since $I_{cs} \geq I_m$, then $0 \leq K_c \leq 1$, so it is not necessary to scale the input data. To make predictions, K_c values were used as inputs, then the predicted values (i.e. the outputs of the network were multiplied by the corresponding clear-sky values in order to obtain the GHI predictions) that could then be compared with the expected values.

B. PREDICTION MODEL BUILDING

Between the solutions we propose and compare, one of the most commonly used is the Feed-Forward Neural Network (FFNN). Contextually, we propose a Nonlinear Autoregressive Neural Network (NAR) architecture also based on the Multilayer Perceptron like the FFNN. In addition, we propose a Long Short-Term Memory Network (LSTM) architecture, often used for time-series forecasting and successfully applied to GHI prediction in recent studies [34], [35]. Finally, we propose an architecture based on the Echo State Network (ESN), which has shown promising results in time-series forecasting [37]. In the following subsections, we will present the neural architectures considered in this study. For each of them, we will describe the properties and strengths, giving particular emphasis to the hyperparameters taken into consideration and properly investigated. In Section IV, instead, we will present and detail all the network configuration w.r.t. the exploited data-set. This is because the optimization of the architecture strictly depends on the data-set under consideration.

1) NONLINEAR AUTOREGRESSIVE NEURAL NETWORK

The Nonlinear Autoregressive Neural Network is an ANN that extends a traditional linear autoregressive model. It is particularly suitable for non-linear time-series that report unexpected spikes and fleeting, transient periods [47]. The general structure is shown in Figure 4. Thus the model can be described as:

$$y_{t+1} = F(y_t, y_{t-1}, y_{t-2}, \ldots, y_{t-n}) \tag{5}$$

where F is an unknown non-linear function and at time t the network is fed with the n regressors of the signal y.

Progressively, we determined the hyperparameters of the network, in particular the number of both regressors and units in the hidden layer. Since there is no rule to determine the best number of regressors mathematically, the choice was made by trial-and-error, going from 2 up to 20 regressors. Regarding the units in the hidden layer, we overestimate the initial number selecting 30 hidden units. This because, in this methodology, we can adopt pruning functionality that allows to eliminate superfluous weights and determine the best network configuration [21]. Once the parameters were selected, the network training was performed using the Levenberg-Marquardt algorithm [48]. Consequently, we have pruned the obtained network with the Optimal Brain Surgeon algorithm [49], and we trained the network again before making the inference for the predictions.

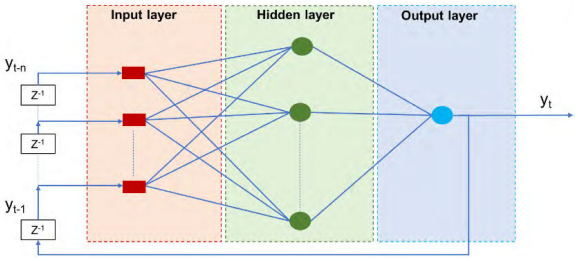


FIGURE 4. Nonlinear autoregressive neural network.

2) FEED-FORWARD NEURAL NETWORK

The third neural network is based on Feed-forward Neural Network (FFNN). Also based on the Multilayer Perceptron, it is characterized by a dense fully connected layer, where information only moves from one side to the other.

In FFNN models, generally, the output of each node is calculated using an activation function applied to the weighted sum of the inputs. The activation function is usually a nonlinear one (e.g. a common choice being the hyperbolic tangent (tanh)). Considering a MLP with n inputs, one hidden layer with m units and one output, the output of the network can be modelled as follows:

$$\hat{y} = F(\sum_{i=1}^{m} W_i f(\sum_{j=1}^{n} w_{ij} u_j + w_{i0}) + W_0) \tag{6}$$

In equation 6, F and f are the activation functions for the output and hidden layer respectively, W_i and w_{ij} are the weights between hidden and output layers and between input and output layers, W_0 and w_{i0} are the biases, and u_j are the inputs.

In a preliminary phase, we considered different architectures with different hidden layers. We found that the best compromise between prediction and computation accuracy is an architecture with two hidden layers. Once the model has been chosen, we determined the hyperparameters, i.e. number of regressors and activation function. For the hidden layers, we have opted for the hyperbolic tangent (tanh)

activation function, since it is a common choice and gives good results [50]. For the output layer, instead, we have chosen a linear activation function. As for the number of inputs and units in the hidden layer, there is no established mathematical technique to choose the best parameters, also in this case, we opted for a trial-and-error approach. First, we arbitrarily decided to select the number of units as two times plus one the number of inputs for networks with a single hidden layer. Then, we investigated the number of regressors from 2 to 20, evaluating the performance for each case. The optimization algorithm used for training is the Adaptive Moment Estimation (Adam optimizer) [51]. This algorithm is closely related to two other optimization techniques, Root Mean Square Propagation (RMSProp) and Adaptive Gradient Algorithm (AdaGrad), combining their features together. To avoid the phenomenon of overfitting, we have used the early-stopping technique [52]. In practice, during the training phase, training is stopped when there is no improvement in the validation set for a few steps. The additional benefit of early-stopping is the significant reduction of the training times.

3) LONG SHORT-TERM MEMORY NEURAL NETWORK

The Long Short-Term Memory Neural Network (LSTM) represents an evolution of a canonical recurrent neural network developed to solve the ''vanishing gradient'' problem [53]. This is a problem that arises during the training of such neural networks with backpropagation methods. These architectures are particularly suitable in the prediction of time-series because, thanks to their structure, they are able to preserve the error that can be backpropagated through time and layers. By maintaining a more constant error, they allow recurrent nets to continue to learn over many time steps. Since the LSTM was proposed, different variations of the architecture were developed [54]. The typical LSTM unit is composed of a cell, an input gate, an output gate, and a forget gate. The LSTM unit structure is shown in Figure 5.

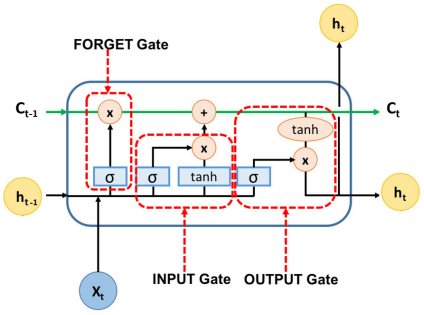


FIGURE 5. Long short-term memory unit.

The cell state, represented by the top horizontal line in Figure 5, contains information that is passed on to the next cell and can be modified by the gates. First, the forget gate decides what information is kept or thrown away from the cell state. The sigmoid activation function outputs a value between 0 and 1, where 0 means to forget the previous state completely and 1 to keep it as is. Then, new information can be stored in the cell state. This is done by the input gate, which is composed of a sigmoid and a tanh layer. Finally, the output of the LSTM unit is determined by the output gate based on the cell state.

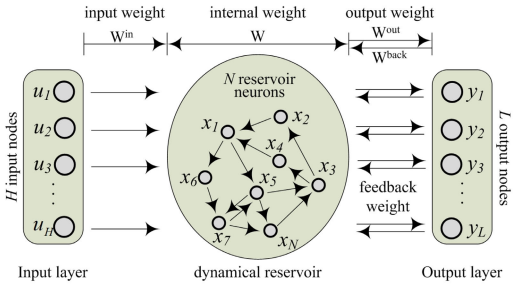


FIGURE 6. Echo state network.

Mathematically, LSTM cell computes its short-term state h_t, its long-term state c_t and its output y_t at each time step t basing the following set of equations expressed in vectorial:

$$\begin{aligned} i_t &= \sigma(W^{xi}x_t + W^{hi}h_{t-1} + b_i) \\ f_t &= \sigma(W^{xf}x_t + W^{hf}h_{t-1} + b_f) \\ o_t &= \sigma(W^{xo}x_t + W^{ho}h_{t-1} + b_o) \\ g_t &= tanh(W^{xg}x_t + W^{hg}h_{t-1} + b_g) \\ c_t &= f_t \otimes c_{t-1} + i_t \otimes g_t \\ y_t &= h_t = o_t \otimes tanh(c_t) \end{aligned} \tag{7}$$

where W^{xi}, W^{xf}, W^{xo} and W^{xg} are the weight matrices of the connections to the input vector x_t, W^{hi}, W^{hf}, W^{ho} and W^{hg} are the weight matrices of the connections to the previous short-term state vector h_{t-1} and b_i, b_f, b_o and b_g are the bias terms.

For our implementation, we used an LSTM layer for the hidden part of the networks, while the output layers are simple dense layer, as for the FFNN. As already discussed for the FFNN in Section III-B2, the same trial-and-error approach was adopted to determine the number of regressors, hidden units and hidden layers. The Adam optimizer was again used for training, and the same early-stopping technique was also used to avoid overfitting.

4) ECHO STATE NETWORK

The Echo State Network (ESN) is a recurrent neural network composed by an input layer, a recurrent hidden layer called ''reservoir'' and an output layer (see Figure 6). ESNs are an implementation of so-called ''reservoir computing'' [55]. The main idea is to have a fixed, random, sparsely connected recurrent layer, and a readout layer, connecting the reservoir to the output. In the simplest architecture, these output connections are the only trainable ones. It is also possible to add direct trainable connections from input to output, bypassing the reservoir, and feedback connections from output to reservoir [56].

The reservoir must satisfy the ''echo state property'' (ESP), which guarantees that the effects of the initial conditions vanish as time passes. It has been empirically observed that the ESP usually holds when the spectral radius of the reservoir weight matrix is lower than one [56]. However, this condition is neither sufficient nor necessary. Stricter conditions have been determined by Yildiz *et al.* [57]. Jaeger [56] gives the following definitions for the ESN reservoir state (Equation 8) and output (Equation 9).

$$\mathbf{x}_{t+1} = f(\mathbf{W}\mathbf{x}_t + \mathbf{W}^{in}\mathbf{u}_{t+1} + \mathbf{W}^{fb}\mathbf{y}_t) \tag{8}$$

$$\mathbf{y}_t = g(\mathbf{W}^{out}\mathbf{z}_t) \tag{9}$$

where $\mathbf{x}_t$ is the reservoir state at time t, $\mathbf{u}_t$ is the input vector, $\mathbf{y}_t$ is the output of the network, $\mathbf{z}_t = [\mathbf{x}_t; \mathbf{u}_t]$ is the system state, and $\mathbf{W}$, $\mathbf{W}^{in}$, $\mathbf{W}^{fb}$, and $\mathbf{W}^{out}$ are the reservoir weight matrix, the input weight matrix, the optional feedback matrix, and the output matrix, respectively.

One interesting feature of ESNs is that they are straightforward to train, unlike other recurrent neural networks. In ESNs, the recurrent layer is fixed, and this greatly simplifies the training process. Some hyperparameters need to be determined. The number of inputs (regressors) was determined by a trial-and-error approach. For the size of the reservoir, Lukoševičius [58] suggests that a ''big'' reservoir is usually better, given the sparsity of the connections between its units. Clearly, it is not possible to implement an arbitrarily big reservoir, since memory consumption needs to be taken into account. Keeping this in mind, the first choice was to use 500 units. Starting from that upper limit, reservoirs with 50, 100, and 200 units were also tested, to verify the assumption that ''bigger is better''. The reservoir density was chosen again by trial-and-error, and 0.1 was selected for its value. Another important parameter is the spectral radius. As already discussed, a value lower than 1 should guarantee the echo state property. It is usually a good choice to choose a value close to 1, as suggested in [58], so 0.9 was selected.

C. MULTI-STEP PREDICTIONS

Generally, the inference of neural networks to obtain the prediction of time-series can be done with three different modes [59]:

- iterative prediction;
- multi-output neural network;
- dedicated network for each forecast horizon.

The iterative method is often used, particularly for short-term horizons (i.e. few next time-steps). It is based on a network model trained for single-step predictions (i.e. with only one output), where for each step after the first one the prediction for the previous step is used as input. This has the disadvantage that the prediction errors tend to accumulate. Another option is to use a single network with n outputs, where n is the number of steps to predict. Unlike the previous method, it is necessary to determine the number of steps in the future that need to be predicted when the model is built. However, it is expected that, for longer forecast horizons, this method will give better performance than the iterative method. Finally, it is possible to use a different network for each forecast horizon. However, the study in [59] shows that this methodology has better results for short-term horizon (e.g. next 2 or 3 time-steps), but is outperformed by the multi-output method for longer horizons.

In this work, we evaluate the first two methods proving that the multi-output approach outperforms the iterative one (see Section IV-B), and it was therefore chosen for all the following experiments. This does not apply to the NAR network, however, since the realized model already contains a function to perform multi-step predictions, which was therefore used.

IV. RESULTS

In this Section, we present our experimental results. First, we briefly describe the statistical indicators used to analyse and compare the predictions. Then, we prove that by using a multi-output artificial neural network for predictions with many steps ahead, we can obtain better results than the iterative method, justifying the choice of the former approach. Finally, we describe the prediction performances obtained with raw GHI data, Tikhonov regularisation and clear-sky index.

The proposed ANNs are implemented in Python by using the Keras library with Tensorflow backend. To train and validate them, we run our simulations in a server equipped with a CPU 2× Intel Xeon E5-2680 v3 2.50 GHz and 128 Gb of RAM.

A. STATISTICAL INDICATORS

State of the art has a large number of statistical indicators to evaluate the performances of neural networks in terms of predictions [60]. In the study of time-series, the three main indexes adopted are:

- MAD - the *Mean Absolute Difference* between predicted and observed values;
- RMSD - the *Root Mean Square Deviation* defined as the standard deviation of the difference between the predicted and the observed values;
- R^2 - the *Coefficient of Determination*, defined as square of the correlation (R) between predicted and observed values.

Their mathematical expressions are shown in Equations 10, 11 and 12 respectively, where I_m is the measured value of irradiance, I_p is the predicted value, and the subscript *avg* indicates the average value.

$$MAD = \frac{100}{I_{m,avg}} \frac{\sum_{i=1}^{N} |I_{p,i} - I_{m,i}|}{N} \tag{10}$$

$$RMSD = \frac{100}{I_{m,avg}} \sqrt{\frac{\sum_{i=1}^{N} (I_{p,i} - I_{m,i})^2}{N}} \tag{11}$$

$$R^2 = 1 - \frac{(\mathbf{I}_m - \mathbf{I}_p) \cdot (\mathbf{I}_m - \mathbf{I}_p)}{(\mathbf{I}_m - I_{m,avg}) \cdot (\mathbf{I}_m - I_{m,avg})} \tag{12}$$

TABLE 2. MAD (%), R^2, and RMSD (%) of NAR predictions.

Model	Indicator	15 min	30 min	45 min	60 min	75 min	90 min	105 min	120 min
NAR-7 (7 regressors)	MAD	10.79	16.95	21.64	26.12	30.79	35.79	41.32	47.48
	R^2	0.96	0.93	0.91	0.89	0.86	0.83	0.80	0.76
	RMSD	30.50	41.29	47.78	53.30	58.79	64.60	70.78	77.26
NAR-12 (12 regressors)	MAD	10.85	16.74	21.03	24.90	28.86	32.84	36.96	41.39
	R^2	0.96	0.93	0.91	0.89	0.87	0.85	0.83	0.81
	RMSD	30.43	40.62	46.62	51.41	56.14	60.65	65.05	69.55

MAD and RMSD are expressed in percentage and not in absolute units (i.e. $W\,m^{-2}$ for irradiance). For MAD and RMSD, a lower value indicates a smaller error and therefore, better performance. R^2, on the other hand, shows the correlation between real and predicted values, where a value of 1 means complete correlation, while lower values indicate a lower correlation factor.

Therefore, to evaluate the performances of each single neural network model, first we need to compare the final GHI forecast with the corresponding real trend and express the error rate with statistical performance indicators described above. Then, to perform a correct fair comparison between the different neural network models, we compare their performance indicators. To do so, all the models must be trained and tested i) with the very same original data-set to which ii) we applied the very same pre-processing technique and iii) after their optimization phases succeeded.

B. ITERATIVE VS. MULTI-OUTPUT NETWORKS

As introduced in section III-C, we have applied two different methodologies for multi-step predictions, iterative and multi-output. The former uses a single-step model and iteratively generates multiple predictions; the latter gives the desired n predictions exploiting a single step of calculation (i.e. n steps in the future). Figure 7 shows the comparison for the FFNN exploiting raw GHI data with time horizon from 15 min to 2 h.

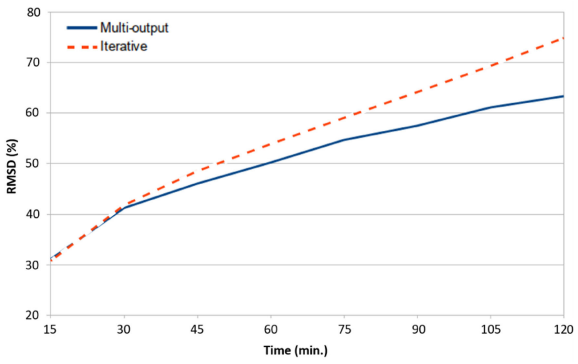


FIGURE 7. Comparison of RMSD for iterative and multi-output predictions using a feed-forward neural network (raw GHI).

The performance is similar for the first 30 minutes, but the multi-output network starts improving for mid-term time horizons. Increasing the time horizon will result in the accumulation of the error, further widening the gap between the two approaches. Figure 8 depicts the same experiment for the LSTM network still exploiting raw GHI. Again, the experimental results report similar behaviour with the LSTM network for the first 30 min. Instead, we do not apply our iterative approach to NAR and ESN because their models already embed a feature to perform multi-step predictions. These results justify our choice of using networks with multiple outputs to predict GHI many steps ahead, in accordance with [59]. Consequently, all the results illustrated in the following sections are based on the multi-output approach.

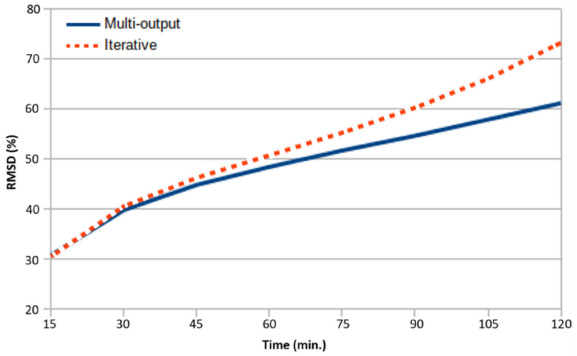


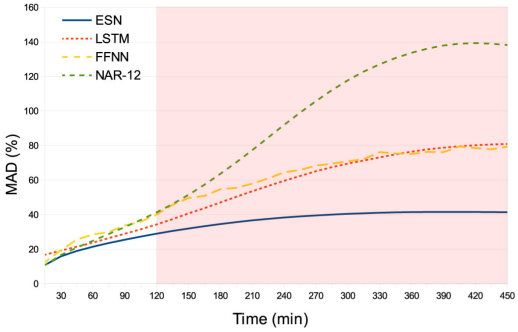
FIGURE 8. Comparison of RMSD for iterative and multi-output predictions using a long short-term memory (raw GHI).

C. GHI PREDICTION EXPLOITING RAW GHI DATA

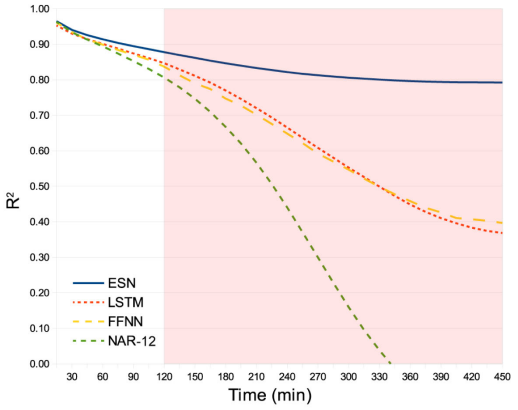
In this section, we present the performance of our neural networks optimised to use the raw GHI data-set for both training and test (see Section III-A).

Starting from NAR, we used as reference the architecture deeply described in [21]. Consequently, we designed the same network (with 7 regressors and 30 neurons, before pruning), exploiting the very same data-set (for both training and test) to guarantee a fair comparison. Then, as described in Section III-B1, different numbers of regressors were evaluated, trying to improve the performance and find the best network. Table 2 reports the comparison of raw GHI predictions between the reference model with 7 regressors [21] (hereinafter NAR-7) and our best model with 12 regressors (hereinafter NAR-12) in terms of MAD, R^2 and RMSD.

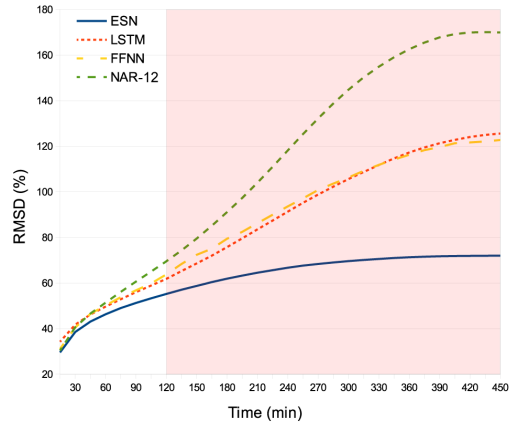
The results report that NAR-12 performs slightly better than NAR-7 for all three statistical indicators, particularly



(a) MAD

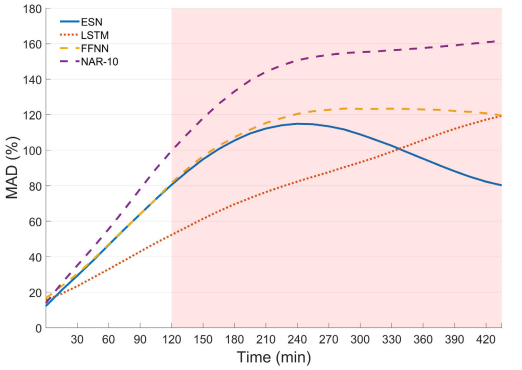


(b) R^2

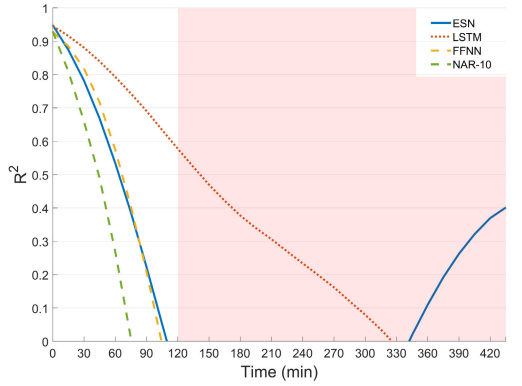


(c) RMSD

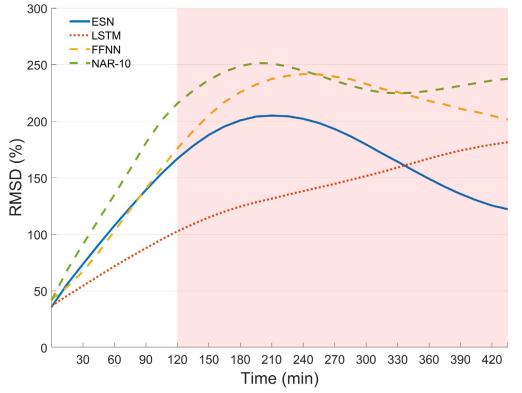
FIGURE 9. Trend evolution for predictions with raw GHI data.



(a) MAD



(b) R^2



(c) RMSD

FIGURE 10. Trend evolution for predictions with "hybrid" approach based on Tikhonov regularisation.

for longer time horizons. Up to 45 min, the performance of predictions is quite similar. The improvements can be noted from 60 min onward. At 120 min, NAR-12 performs better than NAR-7 with improvements of about 6%, 0.05 and 8% for MAD, R^2 and RMSD, respectively. Consequently, we chose this new model to be compared with the other networks.

For FFNN, LSTM and ESN, we have tested different configurations, as described in Section III, and we compared their RMSD values to decide which parameters give best results. In the following, we report the configuration of each neural network:

- **FFNN:** 19 regressors and 2 hidden layers; the first layer with 200 neurons and the second with 100 neurons.
- **LSTM:** 16 regressors with 2 hidden layers; the first layer with 10 neurons and the second with 5 neurons.
- **ESN:** 3 regressors with a reservoir of 500 neurons.

Looking at these parameters, it can be noted that both FFNN and LSTM work better with a greater number of regressors, while the ESN has the best performance with just 3 regressors. As for the size of the networks, the ESN works well with a big reservoir, as recognised in literature [58]. The FFNN also has better results using a big amount of units. Whilst, the LSTM works better with a smaller architecture.

TABLE 3. MAD (%), R^2, and RMSD (%) for predictions with raw GHI data.

Network	Indicator	15 min	30 min	45 min	60 min	75 min	90 min	105 min	120 min
NAR-12	MAD	10.85	16.74	21.03	24.90	28.86	32.84	36.96	41.39
	R^2	0.96	0.93	0.91	0.89	0.87	0.85	0.83	0.81
	RMSD	30.43	40.62	46.62	51.41	56.14	60.65	65.05	69.55
FFNN	MAD	12.62	19.53	25.66	28.55	30.00	33.76	35.94	40.10
	R^2	0.96	0.93	0.91	0.90	0.88	0.87	0.86	0.84
	RMSD	31.15	40.54	46.49	50.20	53.81	56.81	59.85	63.78
LSTM	MAD	16.87	19.29	21.51	23.86	26.47	28.98	31.54	34.36
	R^2	0.95	0.93	0.91	0.90	0.89	0.87	0.86	0.85
	RMSD	34.27	41.75	46.20	49.67	52.91	56.03	58.92	61.84
ESN	MAD	10.96	15.97	19.05	21.49	23.62	25.53	27.34	29.03
	R^2	0.96	0.94	0.93	0.91	0.90	0.89	0.89	0.88
	RMSD	29.60	38.55	43.10	46.31	49.03	51.28	53.31	55.22

TABLE 4. MAD (%), R^2, and RMSD (%) for predictions with the "hybrid" approach based on Tikhonov regularisation.

Network	Indicator	15 min	30 min	45 min	60 min	75 min	90 min	105 min	120 min
NAR-10	MAD	13.83	25.01	35.07	45.13	55.61	66.64	78.17	89.46
	R^2	0.93	0.81	0.66	0.49	0.27	0.01	−0.30	−0.61
	RMSD	41.65	68.45	91.20	113.03	135.10	157.53	180.69	200.80
FFNN	MAD	16.66	23.37	30.35	38.41	47.06	55.64	64.08	72.56
	R^2	0.93	0.89	0.82	0.72	0.58	0.41	0.21	−0.01
	RMSD	42.20	53.55	67.29	84.05	102.96	121.84	140.40	158.60
LSTM	MAD	15.65	19.16	23.41	28.23	33.00	37.93	42.90	47.80
	R^2	0.95	0.92	0.88	0.84	0.79	0.74	0.69	0.63
	RMSD	36.72	45.71	54.47	63.18	71.78	79.97	87.76	95.53
ESN	MAD	12.17	21.10	29.30	37.80	46.68	55.50	63.98	72.47
	R^2	0.95	0.87	0.78	0.67	0.54	0.39	0.22	0.05
	RMSD	35.73	56.01	73.69	90.94	107.73	123.66	139.19	153.80

Our first expectation was that the multi-output architecture used for multi-step prediction improves the results of the NAR, particularly for longer time horizons. We also expected that the LSTM would perform better than the FFNN. In fact, its recurrent structure should be more suitable to model the temporal behaviour of GHI time-series. Moreover, LSTM has shown very good performance in many tasks, including time-series forecasts [54].

Figures 9(a) to 9(c) show plots of MAD, R^2 and RMSD over the time for all the proposed neural network architectures. These plots show that the results are very similar for short-term predictions (i.e. up to 60 min). Then, the four trends start diverging each other for all three indicators. Performances of NAR-12 (see the green dotted line) rapidly decrease compared to the other networks. Contrary to our expectations, LSTM does not clearly outperform the FFNN (see red and yellow dotted-lines, respectively). In fact, both architectures have very similar trends with a few negligible differences. These plots strongly highlight the very good results of the ESN for mid-term forecasts, which clearly outperforms the other neural networks. Table 3 details the values of MAD, R^2 and RMSD for the four networks up to 120 min ahead predictions. In our view, this is a good time horizon in which the forecast error is still acceptable. As reported in Table 3, at 120 min ESN outperforms the other neural networks with $MAD = 29.03\%$, $R^2 = 0.88$ and $RMSD = 55.22\%$, clearly improving the performance of about 12%, 0.07 and 14% (for MAD, R^2, and RMSD, respectively) w.r.t. NAR-12.

D. GHI PREDICTION EXPLOITING TIKHONOV REGULARISATION

As a second analysis, we focused on predictions based on the Tikhonov regularisation applied to the raw GHI data-set, as described in Section III-A1. As discussed in [46], Tikhonov regularisation is not applicable to real-time data. Thus, we exploited the "hybrid" approach presented in [46] in which the Tikhonov regularisation is applied only on the training-set used for training our ANNs. Instead the test-set, used to assess the performance of our ANNs, consists of raw GHI data.

Since the data-set is different w.r.t. our previous NAR-7, we applied the very same methodology described in [21] to design the best NAR suitable for this training-set pre-processed with Tikhonov regularisation. The resulting architecture (hereinafter NAR-10) is characterised by 10 regressors and 21 neurons, before pruning. To design and train the best network architecture for FFNN, LSTM and

TABLE 5. MAD (%), R^2, and RMSD (%) for predictions with Tikhonov regularisation applied also to the test-set.

Network	Indicator	15 min	30 min	45 min	60 min	75 min	90 min	105 min	120 min
NAR-10	MAD	0.95	2.56	4.72	7.39	10.56	14.23	18.44	23.24
	R^2	1.00	1.00	1.00	0.99	0.99	0.98	0.97	0.95
	RMSD	3.28	5.49	8.37	11.92	16.14	21.02	26.64	32.90
FFNN	MAD	5.15	6.78	8.17	9.27	10.06	10.50	11.09	12.08
	R^2	1.00	1.00	1.00	0.99	0.99	0.99	0.98	0.98
	RMSD	6.21	8.47	10.72	12.88	14.87	16.85	19.33	22.42
LSTM	MAD	3.38	2.85	3.12	3.82	4.75	5.83	7.07	8.50
	R^2	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99
	RMSD	4.99	4.68	5.40	6.77	8.56	10.67	13.05	15.68
ESN	MAD	0.65	1.46	2.51	3.79	5.31	7.06	9.00	11.13
	R^2	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.98
	RMSD	2.76	4.35	6.26	8.47	10.99	13.81	16.90	20.21

TABLE 6. MAD (%), R^2, and RMSD (%) for K_c prediction.

Network	Indicator	15 min	30 min	45 min	60 min	75 min	90 min	105 min	120 min
NAR-10	MAD	9.75	14.43	17.31	19.71	21.72	23.44	24.93	26.15
	R^2	0.96	0.94	0.92	0.91	0.89	0.88	0.87	0.85
	RMSD	29.61	39.18	44.28	48.32	51.83	54.85	57.71	60.42
FFNN	MAD	10.38	14.20	16.59	18.71	20.80	22.28	23.66	25.14
	R^2	0.96	0.94	0.92	0.91	0.90	0.89	0.89	0.87
	RMSD	29.81	38.83	43.89	46.93	49.21	51.49	53.39	56.04
LSTM	MAD	10.30	14.02	16.46	18.24	19.92	21.54	23.03	24.38
	R^2	0.96	0.94	0.92	0.91	0.90	0.89	0.89	0.88
	RMSD	30.34	39.25	43.75	46.90	49.52	51.54	53.37	55.24
ESN	MAD	10.10	14.66	17.40	19.47	21.24	22.80	24.17	25.47
	R^2	0.97	0.94	0.92	0.91	0.90	0.90	0.89	0.88
	RMSD	29.56	38.71	43.42	46.51	49.05	51.14	53.05	55.02

ESN, we applied the same trial-and-error approach described in Section IV-C. In the following, we report the configuration of each neural network:

- **FFNN:** 10 regressors and 2 hidden layers; the first layer with 50 neurons and the second with 25 neurons.
- **LSTM:** 16 regressors with 2 hidden layers; the first layer with 10 neurons and the second with 5 neurons.
- **ESN:** 3 regressors with a reservoir of 100 neurons.

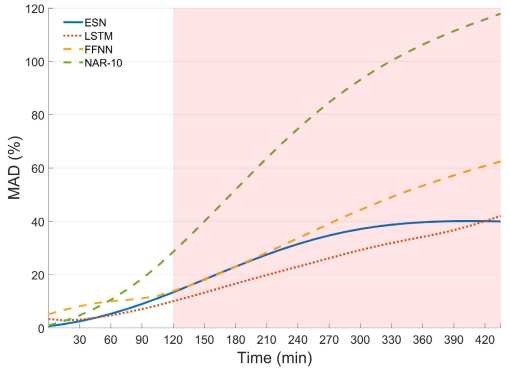
Table 4 reports the values of MAD, R^2 and RMSD for the four ANN up to 120 min ahead predictions. Comparing these results with those in Table 3 (i.e. prediction performance exploiting raw GHI data), we can notice that the performance of all the ANN gets worse when we exploit this "hybrid" approach. In general, the error on predictions is too high and it is not acceptable. This is also confirmed by the trends reported in Figures 10(a) to 10(c). Thus, this "hybrid" approach is not suitable for this application scenario.

If we apply the Tikhonov regularisation also on the test-set, performances significantly improve in average of about 10%, 1 and 25% for MAD, R^2 and RMSD, respectively (see Table 5 and Figures 11(a), 11(b) and 11(c)). However, the Tikhonov regularisation should be rethought to work in real-time.

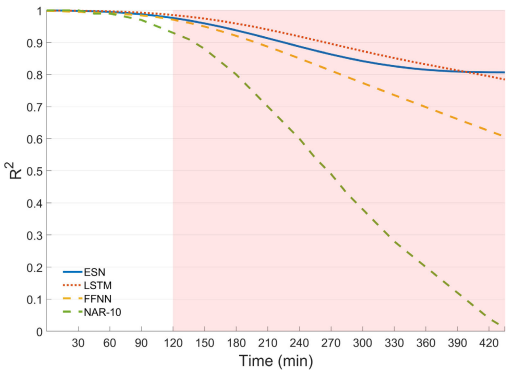
E. CLEAR-SKY INDEX PREDICTION

As a third analysis, we focus on predictions exploiting the clear-sky index (K_c). The same network configurations for the NAR, FFNN, LSTM and ESN used for the GHI prediction exploiting Tikhonov regularisation (see Section IV-D) were also applied to this new data-set.

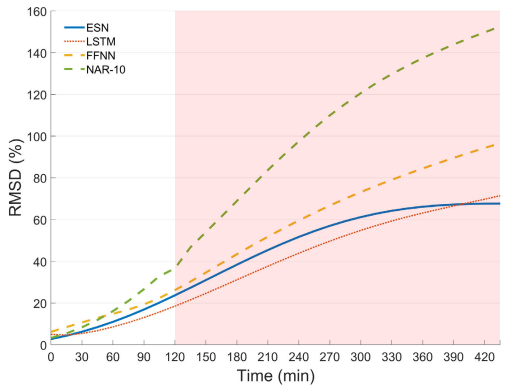
The transformation of GHI into K_c removes the seasonal trends due to the changing position of the sun during the year, and the clear-sky model already takes into account the atmosphere turbidity. So the networks only have to predict the stochastic component due to clouds. The expectation is that the results will be better than those described in the previous sections because these filtered data should be easier to model for the neural networks. The analysis of the obtained results confirms this assumption. As shown in Figures 12(a) to 12(c), the gap between ESN and the other three neural networks is reduced in terms of MAD, R^2, and RMSD. The ESN has only a slight improvement but still is more performing than the others. It appears that the ESN was already able to extract the seasonal trends from the raw GHI data better than the other networks. When only the stochastic component needs to be predicted, and the rest is handled by the clear-sky model, the differences among the networks greatly decreases. As reported in Table 6, at 120 min ESN is



(a) MAD

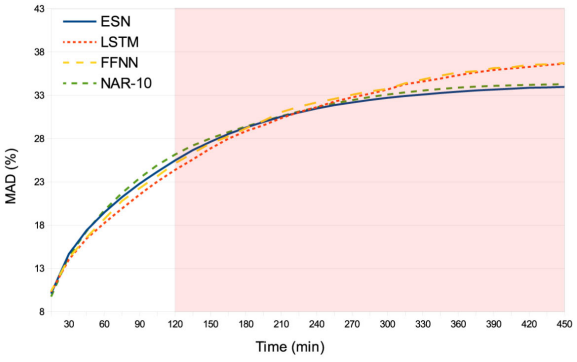


(b) R^2

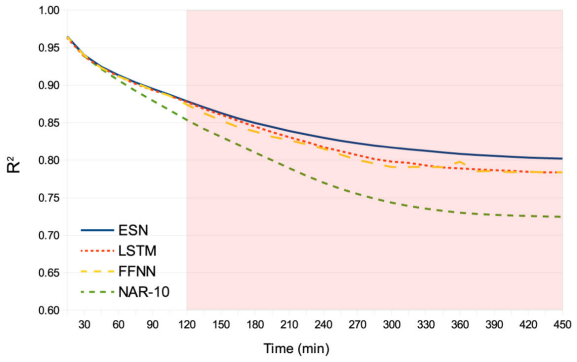


(c) RMSD

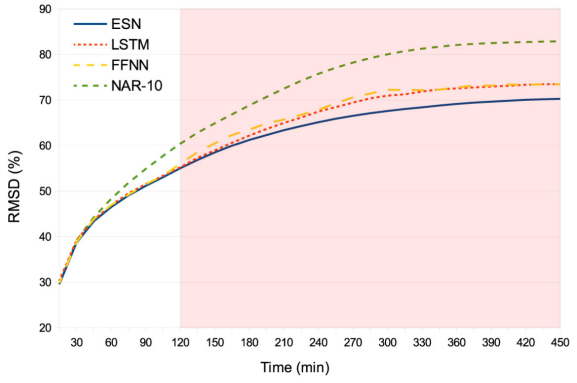
FIGURE 11. Trend evolution for predictions with Tikhonov regularisation applied also to the test-set.



(a) MAD



(b) R^2



(c) RMSD

FIGURE 12. Trend evolution for predictions with K_c.

still outperforming the other neural networks with $MAD = 25.47\%$, $R^2 = 0.88$ and $RMSD = 55.02\%$. Comparing the ESN with the NAR-10, which gives worst performances among the four neural networks, there is an improvement of about 0.6%, 0.03 and 5.4% (for MAD, R^2, and RMSD, respectively).

F. FINAL REMARKS ON PREDICTION RESULTS

Considering the results discussed in the previous sections and reported in Figure 13, the main findings can be summarised as follows:

- designing ANNs following the multi-output approach to forecasting GHI provides better performances than the iterative approach;
- the Echo State Network is the ANN architecture that better performs among those tested;
- using the clear-sky index the prediction accuracy significantly improves and allows to use of smaller networks with fewer regressors;
- best performances are achieved when Tikhonov regularisation is applied. However, to be suitable for GHI forecasts, it should be rethought to work in real-time;
- 120 min is the maximum time horizon reached by our ANNs in which the forecast error is still acceptable.

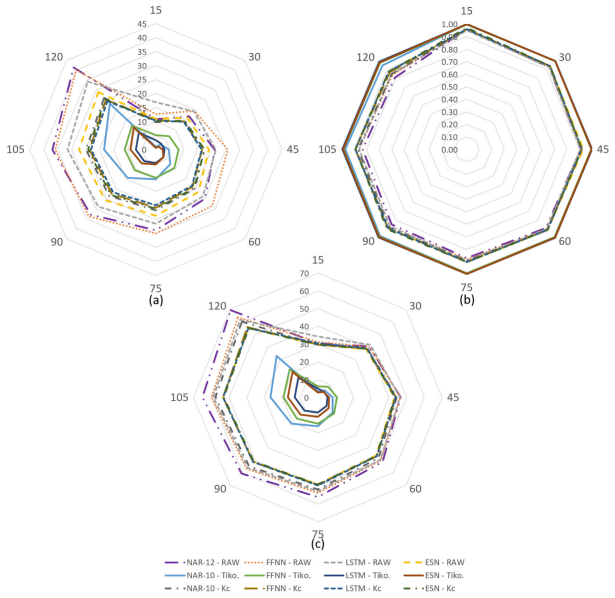


FIGURE 13. Performance comparison of all the presented ANN: (a) MAD, (b) R^2 and (c) RMSD.

V. CONCLUSION

The objective of this manuscript is to propose a methodology for the short- (i.e. from 15 up to 60 min.) and mid-term (i.e. from 60 to 120 min.) GHI forecast based on four different artificial neural networks due to the highly non-linear nature of the physical phenomenon of radiation:

- Nonlinear Autoregressive Neural Network;
- Feed-Forward Neural Network;
- Long Short-Term Memory Network;
- Echo State Network.

In detail, we specifically designed, optimized and compared these four ANNs architectures to understand which was the best to use in this research context.

Consequently, to fully optimize the proposed methodology, we investigated these neural architectures by applying three different approaches based mainly on the use of filtering techniques in the pre-processing phase of the dataset. Specifically, in the first scenario the raw GHI was directly used to train the networks and to make predictions. Secondly, the training set of the GHI series was filtered with Tikhonov regularisation before performing the training procedure and lastly, the GHI time-series was transformed into the clear-sky index series, which was then used to train the networks and make predictions. Lastly, we also compared which of the iterative and multi-output prediction architectural models were performing best.

The obtained experimental results suggest a few interesting considerations. First of all, using a multi-output approach significantly improves the accuracy of multi-step predictions when more than 60 min in the future is required. In our case, this means that for forecast horizons of 45 or 60 min and longer this approach is to be preferred over a single-output model used iteratively. Considering that a single-output model is simpler, it may be better to use the iterative approach when a long-term prediction is not needed for the application.

From a neural network point of view, the experimental results demonstrate that the ESN gives very good results compared to other models, especially when directly predicting GHI. Moreover, compared to the other models, it needs a smaller number of regressors to give very accurate results. The adoption of this model in the context of the GHI forecast has been insufficiently investigated. Given the lack of research in the literature, we are firmly convinced that these findings are genuinely new results worthy of further studies.

The proposed model that uses Tikhonov regularisation to filter the training data and uses the unfiltered GHI for the testing part, which is used successfully in other fields involving time-series forecasting, like blood glucose predictions, does not appear suitable for GHI forecasting. Therefore, filtered data was used in input for the testing part, too. This is not ideal since this method would require, when the system is used for actual predictions ''in the field'', to filter the data every time a new forecast is requested, which might limit the applicability of the method. Moreover, the Tikhonov filter was applied to the whole testing set, divided into long segments, but in a real application, using real-time data, this is not possible, because new data would have to be filtered when it becomes available (e.g. in our scenarios every 15 min). The algorithm would have to be modified accordingly, and the way it might affect the results needs to be studied more in detail. However, with this approach (i.e. exploiting filtered data in the entire process, from training to inference) the results were more interesting, showing that potentially, filtered data allows maintaining a better accuracy for short- and mid-term forecast.

Finally, from a filtering point of view, the results show that the clear-sky index, K_c, greatly improves prediction accuracy when predicting many steps ahead, particularly for NAR, FFNN and LSTM networks. As already stated, the improvement for ESN is small. However, using K_c allowed the ESN to give better results with a smaller reservoir, which is important in terms of memory usage.

REFERENCES

[1] L. N. Araujo, J. T. Belotti, T. A. Alves, Y. D. S. Tadano, and H. Siqueira, ''Ensemble method based on artificial neural networks to estimate air pollution health risks,'' *Environ. Model. Softw.*, vol. 123, Jan. 2020, Art. no. 104567.

[2] World Health Organization. (2019). *How Air Pollution is Destroying Our Health (2019)*. [Online]. Available: https://www.who.int/air-pollution/news-and-events/how-air-pollution-is-destroying-our-health

[3] V. Masson-Delmotte, P. Zhai, H. Pörtner, D. Roberts, J. Skea, P. Shukla, A. Pirani, W. Moufouma-Okia, C. Péan, and R. Pidcock, ''IPCC, 2018: Summary for policymakers,'' *Global Warming 1.5°, IPCC Special Report Impacts Global Warming, 1.5° Pre-Ind. Levels Rel. Global Greenhouse Gas Emission Pathways, Context Strengthening Global Response Threat Climate Change, Sustainable Develop., Efforts Eradicate Poverty*, vol. 1, 2018.

[4] S. Selvakkumaran and E. O. Ahlgren, ''Impacts of social innovation on local energy transitions: Diffusion of solar PV and alternative fuel vehicles in Sweden,'' *Global Transitions*, vol. 2, pp. 98–115, Jan. 2020.

[5] L. Wong and W. Chow, "Solar radiation model," *Appl. Energy*, vol. 69, no. 3, pp. 191–224, 2001.

[6] V. Shelia, J. Hansen, V. Sharda, C. Porter, P. Aggarwal, C. J. Wilkerson, and G. Hoogenboom, "A multi-scale and multi-model gridded framework for forecasting crop production, risk analysis, and climate change impact studies," *Environ. Model. Softw.*, vol. 115, pp. 144–154, May 2019.

[7] J. Nikoletatos and S. Tselepis, "Renewable energy integration in power grids," *Technol. Brief. IEA-ETSAP IRENA*, 2015.

[8] P. Siano, "Demand response and smart grids—A survey," *Renew. Sustain. Energy Rev.*, vol. 30, no. 2, pp. 461–478, 2014.

[9] C.-C. Lin, D.-J. Deng, W.-Y. Liu, and L. Chen, "Peak load shifting in the internet of energy with energy trading among end-users," *IEEE Access*, vol. 5, pp. 1967–1976, 2017.

[10] H. S. Ramadan, M. Becherif, and F. Claude, "Energy management improvement of hybrid electric vehicles via combined GPS/rule-based methodology," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 586–597, Apr. 2017.

[11] A. K. Yadav and S. S. Chandel, "Solar radiation prediction using artificial neural network techniques: A review," *Renew. Sustain. Energy Rev.*, vol. 33, pp. 772–781, May 2014.

[12] T. Mutavhatsindi, C. Sigauke, and R. Mbuvha, "Forecasting hourly global horizontal solar irradiance in South Africa using machine learning models," *IEEE Access*, vol. 8, pp. 198872–198885, 2020.

[13] H. Bouzgou and C. A. Gueymard, "Fast short-term global solar irradiance forecasting with wrapper mutual information," *Renew. Energy*, vol. 133, pp. 1055–1065, Apr. 2019.

[14] H. Zang, L. Liu, L. Sun, L. Cheng, Z. Wei, and G. Sun, "Short-term global horizontal irradiance forecasting based on a hybrid CNN-LSTM model with spatiotemporal correlations," *Renew. Energy*, vol. 160, pp. 26–41, Nov. 2020.

[15] L. Bottaccioli, A. Estebsari, E. Patti, E. Pons, and A. Acquaviva, "A novel integrated real-time simulation platform for assessing photovoltaic penetration impacts in smart grids," *Energy Procedia*, vol. 111, pp. 780–789, Mar. 2017.

[16] L. Bottaccioli, E. Patti, E. Macii, and A. Acquaviva, "GIS-based software infrastructure to model PV generation in fine-grained spatio-temporal domain," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2832–2841, Sep. 2017.

[17] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*. Springer, 2016.

[18] A. Madanchi, M. Absalan, G. Lohmann, M. Anvari, and M. R. R. Tabar, "Strong short-term non-linearity of solar irradiance fluctuations," *Sol. Energy*, vol. 144, pp. 1–9, Mar. 2017.

[19] C. Voyant, G. Notton, S. Kalogirou, M.-L. Nivet, C. Paoli, F. Motte, and A. Fouilloy, "Machine learning methods for solar radiation forecasting: A review," *Renew. Energy*, vol. 105, pp. 569–582, May 2017.

[20] D. S. Kumar, G. M. Yagli, M. Kashyap, and D. Srinivasan, "Solar irradiance resource and forecasting: A comprehensive review," *IET Renew. Power Gener.*, vol. 14, no. 10, pp. 1641–1656, Jul. 2020.

[21] A. Aliberti, L. Bottaccioli, G. Cirrincione, E. Macii, A. Acquaviva, and E. Patti, "Forecasting short-term solar radiation for photovoltaic energy predictions," in *Proc. 7th Int. Conf. Smart Cities Green ICT Syst.*, 2018, pp. 44–53.

[22] A. Aliberti, L. Bottaccioli, G. Cirrincione, E. Macii, A. Acquaviva, and E. Patti, "Non-linear autoregressive neural networks to forecast short-term solar radiation for photovoltaic energy predictions," in *Smart Cities, Green Technologies and Intelligent Transport Systems*, pp. 3–22, 2018.

[23] M. Diagne, M. David, P. Lauret, J. Boland, and N. Schmutz, "Review of solar irradiance forecasting methods and a proposition for small-scale insular grids," *Renew. Sustain. Energy Rev.*, vol. 27, pp. 65–76, Nov. 2013.

[24] E. Vonesh and V. M. Chinchilli, *Linear and Nonlinear Models for the Analysis of Repeated Measurements*. Boca Raton, FL, USA: CRC Press, 1996.

[25] M. Lydia, S. S. Kumar, A. I. Selvakumar, and G. E. P. Kumar, "Linear and non-linear autoregressive models for short-term wind speed forecasting," *Energy Convers. Manage.*, vol. 112, pp. 115–124, Mar. 2016.

[26] G. Reikard, "Predicting solar radiation at high resolutions: A comparison of time series forecasts," *Solar Energy*, vol. 83, no. 3, pp. 342–349, Mar. 2009.

[27] A. Sfetsos and A. H. Coonick, "Univariate and multivariate forecasting of hourly solar radiation with artificial intelligence techniques," *Sol. Energy*, vol. 68, no. 2, pp. 169–178, Feb. 2000.

[28] L. Martín, L. F. Zarzalejo, J. Polo, A. Navarro, R. Marchante, and M. Cony, "Prediction of global solar irradiance based on time series analysis: Application to solar thermal power plants energy production planning," *Solar Energy*, vol. 84, pp. 1772–1781, Oct. 2010.

[29] H. T. C. Pedro and C. F. M. Coimbra, "Assessment of forecasting techniques for solar power production with no exogenous inputs," *Sol. Energy*, vol. 86, no. 7, pp. 2017–2028, 2012.

[30] P. Lauret, C. Voyant, T. Soubdhan, M. David, and P. Poggi, "A benchmarking of machine learning techniques for solar radiation forecasting in an insular context," *Sol. Energy*, vol. 112, pp. 446–457, Feb. 2015.

[31] M. Rana, I. Koprinska, and V. G. Agelidis, "Univariate and multivariate methods for very short-term solar photovoltaic power forecasting," *Energy Convers. Manage.*, vol. 121, pp. 380–390, Aug. 2016.

[32] T. C. McCandless, S. E. Haupt, and G. S. Young, "A regime-dependent artificial neural network technique for short-range solar irradiance forecasting," *Renew. Energy*, vol. 89, pp. 351–359, Apr. 2016.

[33] S. Monjoly, M. André, R. Calif, and T. Soubdhan, "Hourly forecasting of global solar radiation based on multiscale decomposition methods: A hybrid approach," *Energy*, vol. 119, pp. 288–298, Jan. 2017.

[34] A. Alzahrani, P. Shamsi, C. Dagli, and M. Ferdowsi, "Solar irradiance forecasting using deep neural networks," *Procedia Comput. Sci.*, vol. 114, pp. 304–313, Jan. 2017.

[35] M. Abdel-Nasser and K. Mahmoud, "Accurate photovoltaic power forecasting models using deep LSTM-RNN," *Neural Comput. Appl.*, vol. 31, no. 7, pp. 1–14, 2017.

[36] S. Srivastava and S. Lessmann, "A comparative study of LSTM neural networks in forecasting day-ahead global horizontal irradiance with satellite data," *Solar Energy*, vol. 162, pp. 232–247, Mar. 2018.

[37] T. Kmet and M. Kmetova, "A 24h forecast of solar irradiance using echo state neural networks," in *Proc. 16th Int. Conf. Eng. Appl. Neural Netw. (INNS EANN)*, 2015, p. 6.

[38] S. M. Ruffing and G. K. Venayagamoorthy, "Short to medium range time series prediction of solar irradiance using an echo state network," in *Proc. 15th Int. Conf. Intell. Syst. Appl. Power Syst.*, Nov. 2009, pp. 1–6.

[39] A. Deihimi and H. Showkati, "Application of echo state networks in short-term electric load forecasting," *Energy*, vol. 39, no. 1, pp. 327–340, 2012.

[40] P. M. Norgaard, O. Ravn, N. K. Poulsen, and L. K. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems—A Practitioner's Handbook*. London, U.K.: Springer, 2000.

[41] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Elect. Eng.*, vol. 40, no. 1, pp. 16–28, Jan. 2014.

[42] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[43] R. C. de Andrade and C. Tiba, "Extreme global solar irradiance due to cloud enhancement in northeastern Brazil," *Renew. Energy*, vol. 86, pp. 1433–1441, Feb. 2016.

[44] C. J. Smith, J. M. Bright, and R. Crook, "Cloud cover effect of clear-sky index distributions and differences between human and automatic cloud observations," *Sol. Energy*, vol. 144, pp. 10–21, Mar. 2017.

[45] K. Ito and B. Jin, *Inverse Problems: Tikhonov Theory and Algorithms*. Singapore: World Scientific, 2015.

[46] A. Aliberti, I. Pupillo, S. Terna, E. Macii, S. Di Cataldo, E. Patti, and A. Acquaviva, "A multi-patient data-driven approach to blood glucose prediction," *IEEE Access*, vol. 7, pp. 69311–69325, 2019.

[47] M. Nø rgaard, O. Ravn, and N. K. Poulsen, "NNSYSID-toolbox for system identification with neural networks," *Math. Comput. Model. Dyn. Syst.*, vol. 8, no. 1, pp. 1–20, Mar. 2002.

[48] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quart. J. Appl. Math.*, vol. 2, no. 2, pp. 164–168, Jul. 1944.

[49] B. Hassibi and D. G. Stork, "Second order derivatives for network pruning: Optimal brain surgeon," in *Proc. Adv. neural Inf. Process. Syst.*, 1993, pp. 164–171.

[50] B. Karlik and A. Vehbi, "Performance analysis of various activation functions in generalized MLP architectures of neural networks," *Int. J. Artif. Intell. Expert Syst.*, vol. 1, no. 4, pp. 111–122, 2015.

[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[52] L. Prechelt, "Early stopping-but when?" in *Neural Networks: Tricks of the Trade*. Springer, 1998, pp. 55–69.

[53] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[54] K. Greff, R. K. Srivastava, J. Koutnìk, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.
[55] B. Schrauwen, D. Verstraeten, and J. Van Campenhout, "An overview of reservoir computing: Theory, applications and implementations," in *Proc. 15th Eur. Symp. Artif. Neural Netw.*, 2007, pp. 471–482.
[56] J. Herbert, "Echo state network," *Scholarpedia*, vol. 2, no. 9, p. 2330, 2007.
[57] I. B. Yildiz, H. Jaeger, and S. J. Kiebel, "Re-visiting the echo state property," *Neural Netw.*, vol. 35, pp. 1–9, Nov. 2012.
[58] M. Lukoševičius, "A practical guide to applying echo state networks," in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 659–686.
[59] D. M. Kline, "Methods for multi-step time series forecasting neural networks," in *Neural Networks in Business Forecasting*. Hershey, PA, USA: IGI Global, 2004, pp. 226–250.
[60] C. A. Gueymard, "A review of validation methodologies and statistical performance indicators for modeled solar radiation data: Towards a better bankability of solar projects," *Renew. Sustain. Energy Rev.*, vol. 39, pp. 1024–1034, Nov. 2014.

ALESSANDRO ALIBERTI (Member, IEEE) received the B.S. and M.S. degrees in cinema and media engineering with the Politecnico di Torino, and the Ph.D. degree in computer engineering from the Politecnico di Torino, in 2020. He is currently a Postdoctoral Researcher with the Politecnico di Torino. During his academic experience, he proposes innovative and optimized data stream processing and machine learning methodologies, ranging from energy and environmental data and moving to data from CPS systems. Moving toward smart and sustainable energy use, his research interest includes the design and the optimization of innovative machine learning methodologies by exploiting primarily neural networks for the forecasting of time-series in smart city context.

DANIELE FUCINI received the M.S. degree in computer engineering from the Politecnico di Torino, Turin, in 2019. During the academic career, his activities mainly focus on the design of innovative neural networks techniques to forecast global horizontal irradiance by investigating the implementation of such neural models on embedded systems.

LORENZO BOTTACCIOLI (Member, IEEE) received the Ph.D. degree *(cum laude)* in computer engineering from the Politecnico di Torino, Italy, in 2018. He is currently an Assistant Professor with the Energy Center Lab, Politecnico di Torino. His main research interests include smart energy, smart city, and smart communities, with focus on software solutions for planning, analysing, and optimizing smart energy systems, and for spatial representation of energy information.

ENRICO MACII (Fellow, IEEE) received the Laurea degree in electrical engineering from the Politecnico di Torino, in 1990, the Laurea degree in computer science from the Universita' degli Studi di Torino, in 1991, and the Ph.D. degree in computer engineering from the Politecnico di Torino, in 1995. From 1991 to 1997, he was also an Adjunct Faculty with the University of Colorado at Boulder. He was with the Politecnico di Torino, as an Assistant Professor, from 1993 to 1998, and an Associate Professor, from 1998 to 2001. He was with the Politecnico di Torino as the Vice Rector for European affairs, from 2007 to 2009, the Vice Rector for technology transfer, from 2009 to 2015, the Vice Rector for research, from 2009 to 2016, and a Rector's Delegate for international affairs, from 2012 to 2015. He is currently a Full Professor of computer engineering with the Politecnico di Torino. His research interests include the design of electronic digital circuits and systems, with particular emphasis on low power consumption aspects. In the last decade, he has extended his research activities to areas, such as bioinformatics, energy efficiency in buildings, districts and cities, sustainable urban mobility, and clean and intelligent manufacturing.

ANDREA ACQUAVIVA (Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Bologna, Italy, in 2003. In 2003, he became an Assistant Professor with the Department of Computer Science, University of Urbino, Italy. From 2005 to 2007, he was a Visiting Researcher with the Ecole Polytechnique Federale de Lausanne, Switzerland. In 2006, he joined the Department of Computer Science, University of Verona, Italy. He has been with the Interuniversity Department of Regional and Urban Studies and Planning, Politecnico di Torino. He is currently a Full Professor of computer engineering with the Università di Bologna. His research interests include parallel computing for distributed embedded systems, such as multicore and sensor networks, software solutions for smart cities, and simulation and analysis of biological systems using parallel architectures. In the fields above, he has authored over 200 scientific publications,between 2000 and 2018.

EDOARDO PATTI (Member, IEEE) received the M.Sc. and Ph.D. degrees in computer engineering from the Politecnico di Torino, in 2010 and 2014, respectively. From 2014 to 2015, he held an academic visiting position with The University of Manchester. He is currently an Assistant Professor with the Politecnico di Torino. His research interests include ubiquitous computing, the Internet of Things, smart systems and cities, software architectures with particular emphasis on infrastructure for ambient intelligence, software solutions for simulating and optimizing energy systems, and software solutions for energy data visualization to increase user awareness. In the fields above, he has authored over 70 scientific publications,between 2011 and 2018.