



HAL
open science

Short-Term Temperature Forecasting on a Several Hours Horizon

Louis Desportes, Pierre Andry, Inbar Fijalkow, Jérôme David

► **To cite this version:**

Louis Desportes, Pierre Andry, Inbar Fijalkow, Jérôme David. Short-Term Temperature Forecasting on a Several Hours Horizon. ICANN 2019, Sep 2019, Munich, Germany. 10.1007/978-3-030-30490-4_42 . hal-02280784

HAL Id: hal-02280784

<https://hal.science/hal-02280784>

Submitted on 6 Sep 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Short-Term Temperature Forecasting on a Several Hours Horizon^{*}

Louis Desportes^{1*}, Pierre Andry¹, Inbar Fijalkow¹[0000-0002-7322-6794], and Jérôme David²

¹ ETIS, Univ Paris Seine, Univ Cergy-Pontoise, ENSEA, CNRS, F-95000 France
surname.name@ensea.fr

² ZenT, F-95000 France jerome.david@zent-eco.com

Abstract. Outside temperature is an important quantity in building control. It enables improvement in inhabitant energy consumption forecast or heating requirement prediction. However most previous works on outside temperature forecasting require either a lot of computation or a lot of different sensors. In this paper we try to forecast outside temperature at a multiple hour horizon knowing only the last 24 hours of temperature and computed clear-sky irradiance up to the prediction horizon. We propose the use different neural networks to predict directly at each hour of the horizon instead of using forecast of one hour to predict the next. We show that the most precise one is using one dimensional convolutions, and that the error is distributed across the year. The biggest error factor we found being unknown cloudiness at the beginning of the day. Our findings suggest that the precision improvement seen is not due to trend accuracy improvement but only due to an improvement in precision.

Keywords: Forecast · Temperature · smart building · CNN

1 Introduction

1.1 Motivation: EcobioH2 building

Today's buildings energy consumption is decreasing due to progress in used materials and appliances energy management, so that in the near future we could envision positive energy buildings, i.e. buildings that produce more energy than they consume. This implies using a local source of energy, like solar panels on the rooftop or a windmill, rather than power from the grid. However, local energy sources are usually intermittent. Therefore, a storage of the energy is required for the low-to-no-production periods, like batteries, and a way to control storage/usage periods. This implies knowing in advance the energy production

^{*} The research reported in this publication is part of the EcobioH2 project supported by EcoBio and ADEME, the french agency for environnement and energy. This project is funded by the PIA, the french national investment plan for innovation.

and demand. Both being highly influenced by external climate, our study focuses on temperature forecast.

The EcobioH2 project [3] [4] intends to be the first low footprint building in France using hydrogen fuel cells for energy storage and a neural network for its control. The 6-storey building of approximately 10 000 squared meters will host retail, cultural, lodging, offices and digital activities. It will have solar panels on its rooftop to produce energy, hybrid hydrogen energy storage to store it. EcobioH2 project requires temperature forecast to design an energy control and monitoring system balancing local energy needs and energy production.

1.2 Temperature forecast

Temperature forecast is required to refine electricity load forecast [9]. The influence of heat is important on appliances consumption and need for cooling/heating. Moreover the weather fluctuations cause behavior shifts of inhabitants.

Knowing what the outside temperature will be in the next 6 to 24 hours, we can predict, and take into account, how much energy will be needed for heating or cooling the building. The aim of this paper is to investigate the temperature forecast on a several hours horizon with a limited amount of sensors. In particular neither wind speed nor wet-bulb thermometers will be available. The exploitation of our predictor should also not require a too large amount of data.

1.3 Related works

Different methods have been proposed for short-term temperature forecasting. [6] uses Abductive networks, a method that links multiple Volterra series together in order to ease network interpretation. However this method uses a network for each prediction hour which induces a huge complexity. Better methods have since been found. Those methods are outlined below.

Based upon a physical model of temperature, [15] uses Volterra series to propose probabilistic forecasts. The authors propose to use hidden Markov models and the Viterbi algorithm to account for the cloudiness variation. They predict at short terms of 15 minutes and 30 minutes and still have a lot of parameters to learn (2 Volterra series, 2 HMM, 1 Autoregressive filter).

[9] is using Echo-State networks hidden state to account for cloudiness. This implies a long and complex convergence for the network. Furthermore the authors only predict at an hour horizon, when we want to have a several hours horizon, or a day horizon using one completely different network depending on the hour the forecast is made.

Simpler Artificial Neural Networks forecasting methods have also been proposed, such as [11]. The authors use many different sensors (wet temperature, wind speed, humidity, pressure, ...) that we don't have on site. [7] needs the last 10 years of values of a given day as an input of their network and only predicts the values of the next day while we need a forecast for several hours of the current day.

The closest work to ours is [16] detailed in [17]. After training it predicts temperature using only the last 24 hours of temperature measures and computed irradiance data. This prediction happens only at the next hour. The authors then use the prediction in 1 hour of the network to predict the temperature in 2 hours yielding to a propagation error that increases with the forecast horizon. Moreover this prediction method preprocess the data before feeding it to the network. This preprocessing might limit the network learning capacity.

In the sequel, we will investigate different neural networks architectures to improve the several hours horizon forecast

2 Model and problem setting

We want to forecast the outside temperature T_{out} for each hour up to an horizon of $H = 6$ hours or $H = 24$ hours. At the trained network input, we use only the $N = 24$ last hours of temperature values $\mathbf{T}_{out}(t - N + 1 : t)$ and the computed irradiance I_{th} between $t - N + 1$ and $t + H$, where t is the current instant. In other words, we want to find the function f such that the H hours temperature forecast $\widehat{\mathbf{T}}_{out}(t + 1 : t + H)$ is given by: $\widehat{\mathbf{T}}_{out}(t + 1 : t + H) = f(\mathbf{T}_{out}(t - N + 1 : t); \mathbf{I}_{th}(t - N + 1 : t + H))$, where MSE: $\sum_{t=1}^T \sum_{h=1}^H (T_{out}(t + h) - \widehat{T}_{out}(t + h))^2$ is minimal. This problem is shown in Fig. 1.

We denote $\mathbf{X}(t + 1 : t + H) = (X(t + 1), X(t + 2), \dots, X(t + H))$ the vector containing the H values of X between hours $t + 1$ and $t + H$ with a time step of 1 hour. Identically for $X(t - N + 1 : t)$. The computed irradiance I_{th} is the clear-sky irradiance. It is the power received if the sky does not have any cloud. I_{th} can be computed using the equations found in [12]. T_{out} is given in K , and I_{th} in W/m^2 .

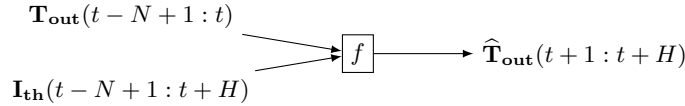


Fig. 1. Temperature forecast scheme

3 Temperature forecasting using neural networks

In the sequel, we will use neural networks to learn the function f as defined in Fig. 1, from the sole T_{out} and I_{th} .

Our work is based upon [17]. This method preprocesses the input data as

$$\mathbf{y}_1 = [I_{th}(t+1); \overline{T_{out}}(t); \max(\mathbf{T}_{out}(t-N+1 : t)); \min(\mathbf{T}_{out}(t-N+1 : t)); T_{out}(t); T_{out}(t-1)]$$

where $\overline{T_{out}}(t)$ is the mean of $\mathbf{T}_{out}(t - N + 1 : t)$. This preprocessed input y_1 is then fed to an hidden neural network layer with bias \mathbf{b}_2 and a \tanh activation function such that $\mathbf{y}_2 = \tanh(\mathbf{y}_1 \times W_2 + \mathbf{b}_2)$. The hidden layer output is then fed to the output layer $Dense(1)$ with bias \mathbf{b}_2 and no activation function giving the one value output $\widehat{T}_{out}(t + 1) = \mathbf{y}_2 \times \mathbf{w}_3 + b_3$. This method is displayed in Fig. 2. We suspect that using $\widehat{T}_{out}(t + 1)$ as an input of the neural network to predict $T_{out}(t + 2)$ may induce error propagation.

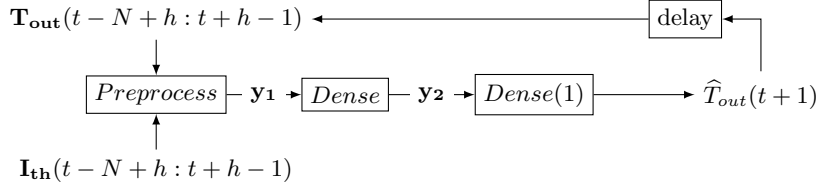


Fig. 2. [17] network structure

3.1 Multi-horizon

In order to avoid error propagation, we propose to adapt the network proposed in [17] to directly forecast up to $t + H$ horizons with each $t + n$, $n \in [1, H]$ as an output of the network.

We preprocess the data in the same way as [17]. However to ensure our network has the same input information as their when run over H horizons we add $I_{th}(t + 2 : t + H)$ to the output of the preprocessing: $\mathbf{y}_1 = [I_{th}(t + 1 : t + H); \overline{T_{out}}(t); \max(\mathbf{T}_{out}(t - N + 1 : t)); \min(\mathbf{T}_{out}(t - N + 1 : t)); T_{out}(t); T_{out}(t - 1)]$. In the contrary of [9], we don't want to train as many networks as the number of outputs. This means that our network's $Dense$ layer is common to all prediction horizons. The formula for this layer is the same, only the dimension changes; the output is made of H values, one for each of the horizon time-step. This yields to: $\widehat{\mathbf{T}}_{out}(t + 1 : t + H) = \mathbf{y}_2 \times W_3 + \mathbf{b}_3$. This network named preprocess_multih is shown in Fig. 3.

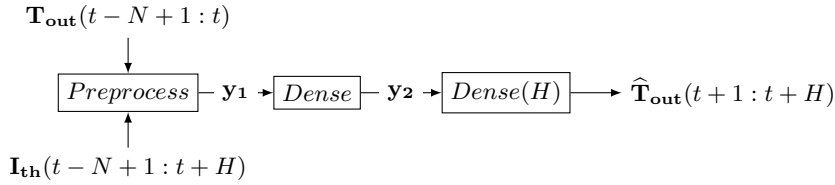


Fig. 3. preprocess_multih network structure

3.2 Raw input

In order to understand if the preprocessing proposed by [17] limits the method performance, we propose to remove the inputs preprocessing and to feed the network with the raw inputs: in $\mathbf{y}_r = (\mathbf{I}_{th}(t - N + 1 : t + H); \mathbf{T}_{out}(t - N + 1 : t))$. These raw inputs are sent to *Dense* and *Dense(H)* layers using the same formulas as the previous network preprocess_multih with different dimension. Fig. 4 show this network named raw_multih.

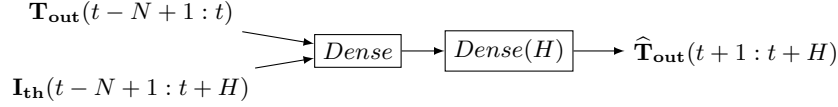


Fig. 4. raw_multih structure

3.3 Convolutions

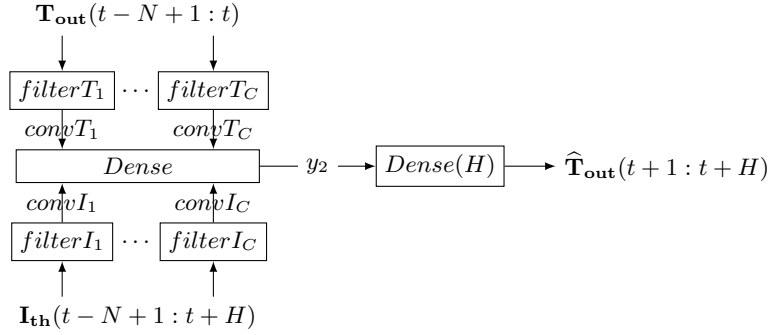


Fig. 5. conv_multih structure

Next, we investigate the usage of a convolutional layer enabling the network to do a better analysis of the inputs [14] because convolutions enable the network to factor for local temporal convolutions. Doing so, we adopt a similar approach as in audio processing [13]. In our case a 1D convolution should be sufficient since our signal seems has a slow frequency evolution. To our knowledge, it is the first attempt to apply such solution to temperature forecasting.

Each raw input is fed to a separate convolutional layer with bias, no activation and no padding. For each convolution c in C , the number of convolutions, the formula is $convI_c = \mathbf{I}_{th}(t - N + 1 : t + H) * filterI_c + bI_c$ and $convT_c =$

$\mathbf{T}_{\text{out}}(t-N+1:t) * filters T_c + b T_c$. With $*$ the convolution operator. The output of those two convolutional layer is flattened and concatenated in a 1 dimension vector y_1 to be sent to the hidden layer. The hidden layer *Dense* and the output layer *Dense(H)* use the same formulas as *raw_multih* and *preprocess_multih*. This method depicted in Fig. 5 is named *conv_multih*.

3.4 Linear predictor

For the sake of comparison, we want to measure the benefit of the neural networks in regards to linear forecasts. We will call the linear method *linear_raw_multih*. In this case, the same raw inputs, $\mathbf{I}_{\text{th}}(t-N+1:t+H)$ and $\mathbf{T}_{\text{out}}(t-N+1:t)$, are fed to the output layer with bias \mathbf{b} and no activation function:

$$\widehat{\mathbf{T}}_{\text{out}}(t+1:t+H) = \mathbf{y}_r \times W + \mathbf{b}$$

4 Available datasets

There are many datasets available that take their data from weather stations around the world. The World Meteorological Organization (WMO) has its own set of weather stations. Composed of an aggregation of weather stations from country specific meteorological organizations. Information about the current weather status is broadcasted using synoptic code, also known as "code synop" [1]. Each station has its own diffusion schedule from each hour to every 6 hours.

The aeronautic industry also has its own weather records called Metar (Meteorological Aerodrome Report)[2]. Each airport makes its own report and broadcasts it every half hour.

The US National Radiation Research Laboratory built some weather stations to study solar radiations. Their data [8] is freely available on each project website and include the local temperature.

Other datasets take their sources from satellite observations. They only use weather stations to calibrate the interpretation of their imagery. Satellite imagery has the benefit of having data for more locations instead of a few discrete measure points.

In this work, we use the NASA Merra-2 [10] data. Those data are composed of an aggregation of different, worldwide, observations with a 1 hour frequency. The dataset is packed with clear-sky irradiance and available freely for specific locations and the years 2005-2006

In the following experiments we used the Merra-2 dataset on the town of Avignon (43.95, 4.817) in France, the location of the EcobioH2 building, obtained through SoDa HelioClim-3 Archives for free [5]. Using the data of 2005 as the train set and of 2006 as the validation and test set. Using the same data for test and validation can be made since we did not reach overfitting in any of our training. And therefore, could not optimize the number of epochs for the validation dataset.

5 Used metrics

We pose $T = T_{out}(t + 1 : t + H)$ to improve readability and N the sample size. Next, we recall the equations of different metrics.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=0}^N (T_i - \hat{T}_i)^2} \quad (1)$$

$RMSE$ is our primary metric. It lends itself easily to interpretation as an error interval since it is expressed in the same metric as the output variable.

$$R^2 = 1 - \frac{\sum_{i=0}^N (T_i - \hat{T}_i)^2}{\sum_{i=0}^N (T_i - \bar{T}_i)^2} \quad (2)$$

R^2 allows us to know how much of the signal is predicted. Giving us an idea of our room for improvement.

$$MAE = \frac{1}{N} \sum_{i=0}^N |T_i - \hat{T}_i| \quad (3)$$

MAE is used in many works. It can be interpreted as an error interval but does not penalize far-off predictions.

6 Results

Using the training data, we perform a stochastic gradient descent algorithm in order to find the different parameters W_i and b_i for the different networks and hyperparameters combination. Then we evaluate each hyperparameter combination on the train set and select the best one. The result is displayed in Table 1.

Algorithms	epochs	learning rate	batch size	number of neurons in hidden layer	conv sizes	number of conv
[17]	50k	0.001	8	5		
preprocess_multih	50k	0.001	8	50		
linear_raw_multih	150k	0.001	32			
raw_multih	150k	0.001	32	14		
conv_multih	50k	0.001	8	60	3	24

Table 1. Best hyper-parameters found for each network

Then we predict on the test set to obtain RMSE, MAE and R^2 . While the error values of RMSE and MAE are in Kelvin, they are equal to the error values in degree Celsius.

In Table 2 we see that [17] error increases way more the further the horizon. It suggests that indeed using $\widehat{T}_{out}(t+1)$ as an input of the neural network to predict $T_{out}(t+2)$ induce error propagation. Therefore our approach to multiple horizon forecast is the right one.

Algorithms	t+1	t+2	t+3	t+4	t+5	t+6
[17]	0.475	0.996	1.433	1.753	1.962	2.084
preprocess_multih	0.412	0.715	0.962	1.171	1.354	1.515
linear_raw_multih	0.409	0.756	1.030	1.256	1.413	1.540
raw_multih	0.375	0.644	0.882	1.098	1.289	1.428
conv_multih	0.340	0.602	0.846	1.053	1.236	1.380

Table 2. RMSE (K) for each horizon and network

The same Table exhibits that, according to the RMSE metric, the best precision for all given metrics is achieved with the conv_multih network. We explain this result by the ability of convolution to characterize the sky cloudiness. Table 3 is available to enable comparison with other works who uses this metric.

Algorithms	t+1	t+2	t+3	t+4	t+5	t+6
[17]	0.342	0.761	1.125	1.396	1.568	1.665
preprocess_multih	0.293	0.531	0.729	0.895	1.038	1.169
linear_raw_multih	0.294	0.565	0.785	0.964	1.089	1.182
raw_multih	0.277	0.481	0.660	0.832	0.982	1.091
conv_multih	0.2413	0.440	0.629	0.792	0.936	1.048

Table 3. MAE (K) for each horizon and network

Table 4 indicates that we predict most of the signal. The fact that even the most basic predictor, linear_raw_multih, gives excellent results validate the way we stated the problem of temperature forecast. However even small forecasting improvement can be useful since they can be leveraged by other predictors.

Algorithms	t+1	t+2	t+3	t+4	t+5	t+6
[17]	99.72	98.77	97.44	96.17	95.20	94.59
preprocess_multih	99.79	99.36	98.85	98.29	97.72	97.14
linear_raw_multih	99.79	99.29	98.68	98.03	97.51	97.05
raw_multih	99.82	99.48	99.03	98.50	97.93	97.46
conv_multih	99.86	99.55	99.11	98.62	98.10	97.63

Table 4. R^2 in percentage for each horizon and network

7 Analysis

We analyze in more details the results of the proposed `conv_multih`, because it is the most precise one, to understand its weaknesses. In Fig. 6, we plot the RMSE against the prediction hour. We see that there is a spike in error at the beginning of the day, from 5am to 8am. Since cloudiness is defined as $c = 1 - \frac{I_{real}}{I_{expected}} = 1 - \frac{I_{local}}{I_{clear\ sky}}$, and since the clear sky irradiance is zero before sunrise, we can't have a cloudiness information before the sunrise. Hence, this spike is due to the insufficient information regarding the upcoming cloudiness of the day.

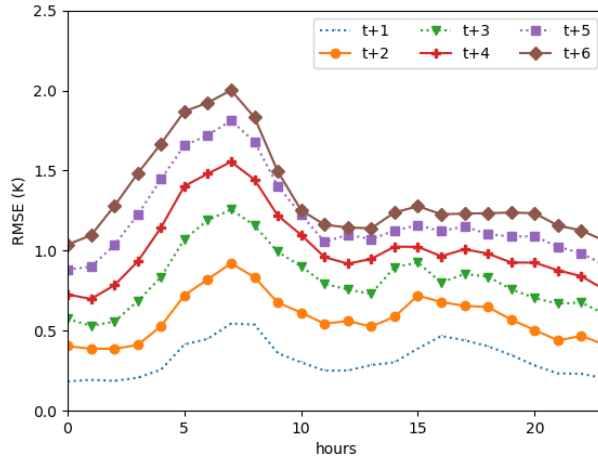


Fig. 6. RMSE in Kelvin, depending on the forecast hour for the best network

We did the same analysis regarding the month of the instant t (Fig. 8) and the evolution of error (RMSE) during the year (Fig. 7). No other spike can be seen, the error is evenly shared across the year.

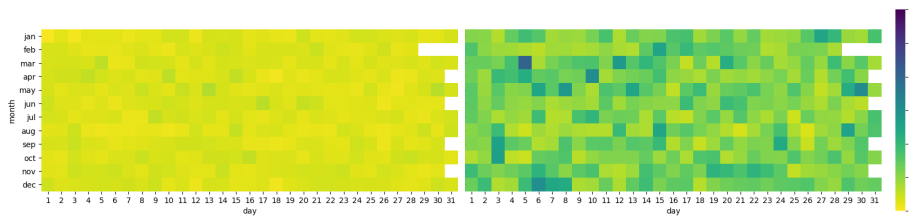


Fig. 7. RMSE in Kelvin, versus the day of the year for t+1 and t+6

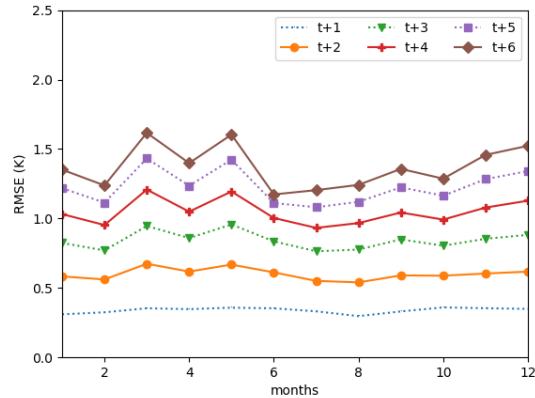


Fig. 8. RMSE in Kelvin, versus the month for the best network

The location of prediction has a great influence on the prediction error. We see in Table 5 that Nice, a city by the sea in the south of Avignon, has better results than Avignon. This is due to the climatic conditions of the city, Nice having way less clouds than Avignon. We used Nice as a comparison point since it is the location [17] used.

Location	t+1	t+2	t+3	t+4	t+5	t+6
Nice	0.2171	0.4006	0.5635	0.7004	0.8049	0.8916
Avignon	0.3397	0.6021	0.8460	1.0527	1.2355	1.3800

Table 5. RMSE (K) of conv_multih for the cities of Nice and Avignon

The goal of forecasting is to be precise (RMSE). It should be noted that this precision can be slightly improved by letting the training continue even when the gradient is small. This can be seen when using a logarithmic scale as in Fig. 9. Still this improvement cost a lot more computations per error unit.

We also want to know if the prediction is reliable. For this reason, we introduce the trend accuracy:

$$\begin{aligned}
 trend_k(x) &= \begin{cases} 0 & \text{if } |x'| \leq k \\ sign(x') & \text{otherwise} \end{cases} \\
 trend_accuracy_k &= \frac{count(trend_k(\hat{T}_i) = trend(T_i))}{N}
 \end{aligned} \tag{4}$$

That is the accuracy of the network to forecast if the temperature will rise, fall or stay constant. We choose $k = 0.3$ as the interval for the constant class. From our results, Table 2, this value seems to be lowest standard deviation we could have. Thus values in this interval could be seen as a stable trend.

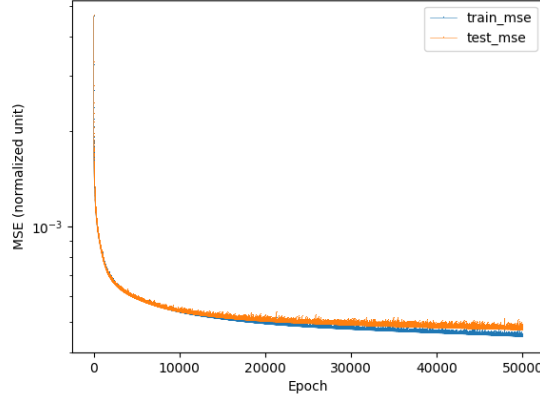


Fig. 9. Normalized MSE of the best network throughout learning

In Table 6 we see that if the conv_multih is always very close from the best accuracy. It is rarely the best one. When preprocessing is removed accuracy values are in a very small interval. This show the improvement seen in RMSE is not due to an improvement in accuracy but solely in an improvement in precision. Keep in mind that the categorization of trends been a bit arbitrary, variation is to be expected, so the difference seen may not be significant.

Algorithms	t+1	t+2	t+3	t+4	t+5	t+6
[17]	0.834	0.710	0.659	0.665	0.683	0.695
preprocess_multih	0.835	0.735	0.690	0.694	0.683	0.689
linear_raw_multih	0.809	0.724	0.704	0.704	0.699	0.705
raw_multih	0.811	0.729	0.714	0.697	0.699	0.697
conv_multih	0.848	0.756	0.713	0.698	0.696	0.695

Table 6. Trend accuracy (.3) for each horizon and network

8 Conclusion

In this paper, we proposed several neural networks for temperature forecast based on the sole previous 24 hours temperature and computed irradiance. We showed that convolutional neural networks are a good tool for temperature forecasting. The proposed networks display precision improvement over linear predictors and non-linear ones. However progress should be made to account for cloudiness at sunrise and improve prediction accuracy. Our solution has the main advantage of not propagating forecasting error through time, and to have the best precision of forecast.

References

1. International Codes, Volume I.1, Annex II to the WMO Technical Regulations: part A- Alphanumeric Codes (2011-2018), https://library.wmo.int/doc_num.php?explnum_id=5708
2. Meteorological Service for International Air Navigation (Annex 3) (2013), https://www.icao.int/Meetings/METDIV14/Documents/an03_cons_secured.pdf
3. ECOBIO H2 – ADEME (Mar 2019), <https://www.ademe.fr/ecobio-h2>
4. Ecobioh2 - etis (Feb 2019), <https://ecobioh2.ensea.fr>
5. HelioClim-3 Archives for Free - www.soda-pro.com (Mar 2019), <http://www.soda-pro.com/web-services/radiation/helioclim-3-archives-for-free>, [Online; accessed 11. Mar. 2019]
6. Abdel-Aal, R.: Hourly temperature forecasting using abductive networks. *Engineering Applications of Artificial Intelligence* **17**(5), 543 – 556 (2004). <https://doi.org/10.1016/j.engappai.2004.04.002>
7. Abhishek, K., Singh, M., Ghosh, S., Anand, A.: Weather forecasting model using artificial neural network. *Procedia Technology* **4**, 311 – 318 (2012). <https://doi.org/10.1016/j.protcy.2012.05.047>, 2012 C3IT
8. Andreas, A.M.: NREL: Measurement and Instrumentation Data Center (MIDC) Home Page (Mar 2019), <https://midcdmz.nrel.gov>, [Online; accessed 2. Apr. 2019]
9. Deihimi, A., Orang, O., Showkati, H.: Short-term electric load and temperature forecasting using wavelet echo state networks with neural reconstruction. *Energy* **57**, 382 – 401 (2013). <https://doi.org/10.1016/j.energy.2013.06.007>
10. Gelaro, R., McCarty, W., Suárez, M.J., Todling, R., al.: The Modern-Era Retrospective Analysis for Research and Applications, Version 2 (MERRA-2). *Journal of Climate* (Jun 2017). <https://doi.org/10.1175/JCLI-D-16-0758.1>
11. Hayati, M., Mohebi, Z.: Application of artificial neural networks for temperature forecasting. *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering* **1**(4), 662 – 666 (2007). <https://doi.org/10.5281/zenodo.1070987>
12. Ineichen, P.: Quatre années de mesures d’ensoleillement à Genève. Ph.D. thesis (07/19 1983). <https://doi.org/10.13097/archive-ouverte/unige:17467>
13. Korzeniowski, F., Widmer, G.: A fully convolutional deep auditory model for musical chord recognition. In: 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP). pp. 1–6 (Sep 2016). <https://doi.org/10.1109/MLSP.2016.7738895>
14. LeCun, Y., Bengio, Y., et al.: Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* **3361**(10) (1995)
15. Ramakrishna, R., Bernstein, A., Dall’Anese, E., Scaglione, A.: Joint probabilistic forecasts of temperature and solar irradiance. In: IEEE ICASSP 2018. <https://doi.org/10.1109/ICASSP.2018.8462496>
16. Salque, T., Marchio, D., Riederer, P.: Neural predictive control for single-speed ground source heat pumps connected to a floor heating system for typical french dwelling. *Building Services Engineering Research and Technology* **35**(2), 182–197 (2014). <https://doi.org/10.1177/0143624413480370>
17. Salque, T.: Méthode d’évaluation des performances annuelles d’un régulateur prédictif de PAC géothermiques sur banc d’essai semi-virtuel. Ph.D. thesis (2013), <http://www.theses.fr/2013ENMP0095>, eNMP 2013