Total Solar Irradiance Forecasting with Keras Recurrent Neural Networks

Amita Muralikrishna[1,2(✉)], Luis E. A. Vieira[1], Rafael D. C. dos Santos[1], and Adriano P. Almeida[1]

[1] National Institute of Space Research,
Av. dos Astronautas, 1758, São José dos Campos, Brazil
{luis.vieira,rafael.santos,adriano.almeida}@inpe.br
[2] Federal Institute of Education, Science and Technology of São Paulo,
Route Presidente Dutra km 145, São José dos Campos, Brazil
amita@ifsp.edu.br
https://www.inpe.br, https://sjc.ifsp.edu.br

Abstract. The prediction of solar irradiance at the top of the atmosphere is useful for research that analyzes the behavior and response of the different layers of the Earth's atmosphere to variations in solar activity. It would also be useful for the reconstruction of the measurement history (time series) of different instruments that suffered from time failures and discrepancies in scales due to the calibration of equipment. In this work we compare three Keras recurrent neural network architectures to perform forecast of the total solar irradiance. The experiments are part of a larger proposal for modularization of the prediction workflow, which uses digital images of the Sun as input, and aims to make the process modular, accessible and reproducible.

Keywords: Recurrent neural network · Total solar irradiance · Keras

1 Introduction

The energy the Earth receives from the Sun is what drives life and many of the processes on the planet. Many of the studies of this relationship and its consequences requires constant monitoring of the Sun, in order to investigate its activity and to try to predict certain types of events on Earth. The main way to study this energetic influence is through the radiation received by the Sun in the form of temperature and irradiation over time.

Recurrent neural networks (RNNs) have shown to be a powerful tool for dealing with time series or data of a sequential nature. New architectures have emerged to further improve the prediction capabilities of the network. This work proposes to perform the prediction of one of the measures of data from the Sun, the total solar irradiance (TSI), using as the input for the network 40 parameters extracted from areas that measure sunspots and active regions. Two types of images of the Sun are used for that.

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12253, pp. 255–269, 2020.
https://doi.org/10.1007/978-3-030-58814-4_18

A workflow describes the whole procedure, based on [21], organizing the steps in modules. The purpose is to make it available for the validation of the results, the execution of more experiments with the same data, or even a new one, and the modification of modules by other researchers with different techniques for tasks like identification, classification and the prediction itself. The intention is, therefore, to make the workflow modular, accessible and reproducible.

As options for experiments with RNNs, the simplest architecture - SimpleRNN (Simple Recurrent Neural Network) - is used, as well as those created to solve the problem of vanishing gradient, LSTM (Long Short Term Memory) and GRU (Gated Recurrent Unit). This work presents their performance for different net configurations.

2 Sun Activity, Sunspots and Active Regions

The Sun owes much of its activity to its rotation and consequent complexity generated in its magnetic field. The events resulting from this activity can be observed under different types of records, in different layers of the star. The presence and the evolution of active regions and sunspots are two important indicators of disturbances in the solar magnetic field. And when studying the effects of this disturbances from the Earth, the TSI is also an important indicator [11].

The sunspots appear visually distinct on the surface of the Sun because they are darker and colder regions, emitting less energy than the rest of the surface [14]. They are formed by two regions: the umbra, a central and darker region, and the penumbra, which thrives around the umbra in only half of the spots, and has a lighter, gray coloration [5]. And the active regions are directly related to the presence of sunspots and can be observed in moments of time that precede and accompany the birth of the spots and can still be visible after their disappearance.

Figure 1 shows a morning record from 2011 November 9th of the SDO (Solar Dynamics Observatory) HMI (Helioseismic and Magnetic Imager) instrument[1] through two types of images, collected in different wavelengths. Active regions are visible in the image on the left, where the opposite colored pixels represent opposite polarities of the solar magnetic field lines, and with more extensive areas than those of the corresponding sunspots, present in the image on the right, at the same instant of time. The presence of sunspots occurs in areas where the magnetic field is most concentrated and intense, but the active regions extend over the entire area where the field disturbance is visible.

Data extracted from both types of images in Fig. 1 will be combined and used at the beginning of the procedure developed to forecast solar irradiance in this work.

[1] SDO webpage: https://sdo.gsfc.nasa.gov/mission/.

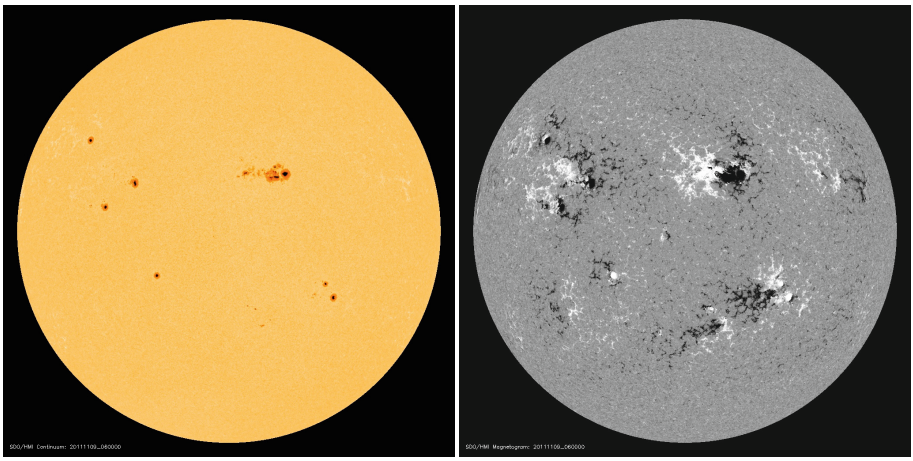


Fig. 1. On the left, an image with sunspots, and on the right, another image with active regions, both registered in the same time on 2011, November 9th.

3 TSI Forecasting

The solar irradiance can be defined as a certain amount of solar radiation per unit area.

When measured on the ground, the solar irradiance is different from when measured at the top (outside) of the Earth atmosphere, since its emission at different wavelengths varies according to the physical and chemical properties of the Earth's atmosphere layers [24].

The solar irradiance measured at the top of the Earth's atmosphere is considered an influential parameter in studies of the atmosphere layers properties' and the consequences of the disturbances they suffer from the influence of solar activity [3]. Other relevant studies about weather and climate on Earth recognized the great influence of solar irradiation for the creation of climate models, as, for example, [4,6,22].

For the second form of irradiance there are two measures: the spectral solar irradiance (SSI), defined by a range of wavelengths covered by solar radiation and received by Earth at a distance of 1 AU[2]; and the TSI: radiation emitted by Sun in all spectral regions, equivalent to the integral of all SSI spectral bands centered per second in 1 m^2 at 1 AU.

The TSI, focus of this work, has a history of measurements performed by instruments on board of different satellites. However, this collection of measurements has gaps in the time series discrepancies (Fig. 2), due to problems such as calibration and degradation of the equipment, which prevents studies that require a long and continuous period of data. Since 2003, TSI measures have

[2] 1 AU (Astronomical Unit) is the average distance between Sun and Earth, outside the Earth's atmosphere.

been collected by the instrument TIM (Total Irradiance Monitor) on board the SORCE (Solar Radiation and Climate Experiment) mission[3]. This data set is used in this work.

In order to solve this problem, many works attempted to estimate the irradiance curve with different methods [2,8,17,21,23]. Most of them use physical models that allow the reconstruction of time fractions of TSI or SSI missing data and not specifically the task of series prediction. One of them [21], whereas, can predict continuous TSI values using a type of Recurrent Neural Network present in a proprietary software. Although it presents results with good accuracy, the software restrictions such as the need for a paid license and the incompatibility of toolboxes between different versions does not allow the procedure to be easily distributed and reproduced.

The aim of this work is in the direction of offering a procedure that performs the TSI estimation, which can allow its reconstruction, but also predict its future variation six hours in advance, using neural networks implemented in free programming languages and libraries.

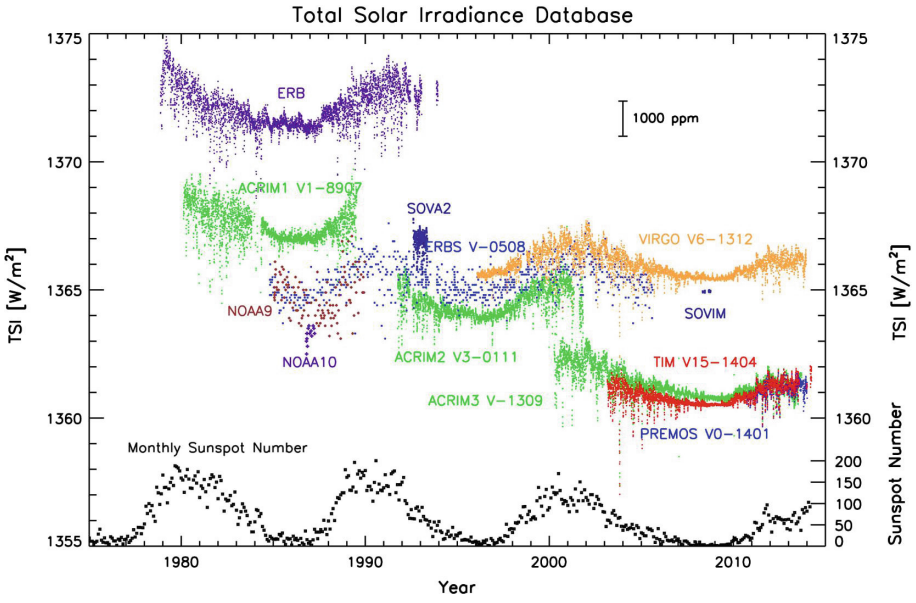


Fig. 2. History of TSI measurements by different instruments on board satellites, as their calibration discrepancies and faults. Measurements performed by TIM were initiated in 2003. The monthly averages of the number of sunspots are also displayed and confirms the influence of the solar activity on the TSI variation [16].

[3] SORCE webpage: http://lasp.colorado.edu/home/sorce/.

3.1 Forecasting Workflow

The complete workflow for solar irradiance prediction (in this work, specifically TSI) begins with the processing of two types of solar images (as shown in Fig. 1, collected by NASA's SDO mission), both of which are records of the solar photosphere (essentially the solar "surface"):

- The continuum images: corresponding to those visible to the human eye, which record the sunspots;
- The magnetograms: which allows the visualization of the active regions, through the polarity of the solar magnetic field.

A set of both images are downloaded considering the desired time period, and then a merged procedure of identification, classification and calculation of the disturbed areas over both is performed. In the sequence, a data matrix is prepared with the calculated and classified areas which are used as input to the neural network that will perform a supervised training, presenting as desired output the TSI values observed for an instant of 6 h ahead. The workflow tasks, separated into modules, where each black outlined block represents a module, can be seen in Fig. 3. This figure also suggests, through the dash outlined boxes, replacement options for some of the tasks.

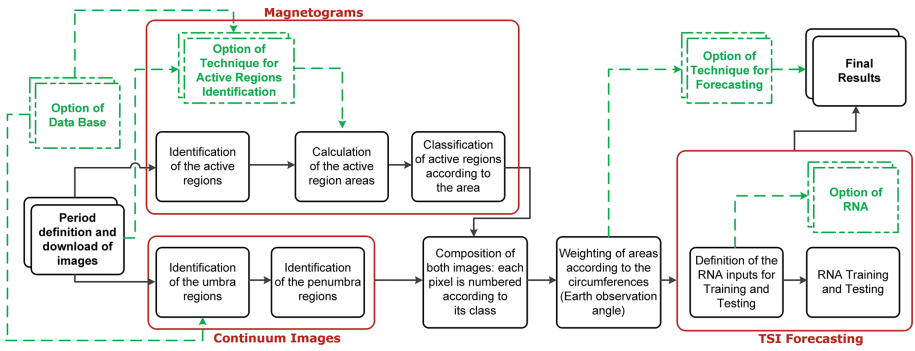


Fig. 3. Complete workflow for processing the solar images and predicting TSI.

Parallel to the specific objective of comparing the performances of different Recurrent Neural Networks architectures in TSI forecasting, this work is part of a larger proposal that aims to share the complete procedure, with the partial and final results, in a modular way, in order to allow modifications on steps of the procedure and extensions that includes other preprocessing and classification approaches.

3.2 Modularity and Reproducibility

Research in areas such as Space Geophysics uses computational tools and techniques to process and analyze data, focused on generating quality final results. However, the detailed record of the entire analytical workflow that was adopted to arrive at such results, and its availability for reuse, is also a valuable tool, and could bring a series of benefits for the researchers.

This concept is already used in other scientific areas. Bioinformatics researchers, for example, can share robust and reliable workflows in the cloud, joined by a large volume of data, that can be reused by a research community who need the same data and processing [7,18,19].

This work is the beginning of a proposal for the modularization of the solar images analysis process, with the final objective of making available a solar irradiance prediction workflow that is, mainly:

Accessible: available on a free platform, language and libraries in the cloud, so that it can be used by an interested researcher, offering with it all the necessary processing devices like operating system, software and libraries, with the originally used versions;

Reproducible: with notebook commented codes that can be: re-executed to validate the developed procedures and the partial and final results obtained; or edited to, for example, allow the reuse of the workflow to process and analyze other input data;

Modular: in order to allow parts (modules) of the workflow to be used separately, or edited to have part of the process replaced, purposing, for example, the creation of new tests with new techniques or tools that perform the same task.

4 Recurrent Neural Networks

This work focuses on the tests performed for TSI forecasting using three types of recurrent neural networks (RNNs).

RNNs are a class of artificial neural networks (ANN) with dynamic behavior, which consider the existing dependency between the instances of entry. This is what differentiates them from the traditional Feed Forward Networks, which treat the inputs as independent from each other.

In order to generate internal memory devices, RNNs also have connections between neurons of the same layer, different from the traditional Feed Forward, in which the connections exist only between the different layers. What happens basically in this new architecture is that the output of a neuron can be fed back into the neuron itself [9].

This flexibility of RNNs allows them to be used in language processing problems such as speech recognition, language modeling, machine translation [10], price monitoring [20] and time series in general, cases in which considering the historical context of the inputs can be an essential differential in the forecast quality.

Essentially, the RNNs follow the basic functioning of other ANNs, regarding the internal processing of neurons, which results in the weighted sum of the input over which an activation function is applied. However, besides the traditional matrix of weights, the simple types of RNNs have another matrix of weights that is used exclusively in the processing of the feedback coming from the previous processing moments.

In Fig. 4, it is possible to visualize the basic process which occurs in a recurrent neuron to calculate its state h in the time instant t, given the input at the same time ($x^{(t)}$) and its previous state ($h^{(t-1)}$), generating its output ($o^{(t)}$). On the left of the picture a briefer representation is made and on the right the unfolding of the process considering instants of time from $t-1$ to $t+1$. In addition to this information, the figure also distinguishes the three weight matrices, with U the input layer weight matrix, V the output layer weight matrix and W the weight matrix applied to the feedback signals [9].

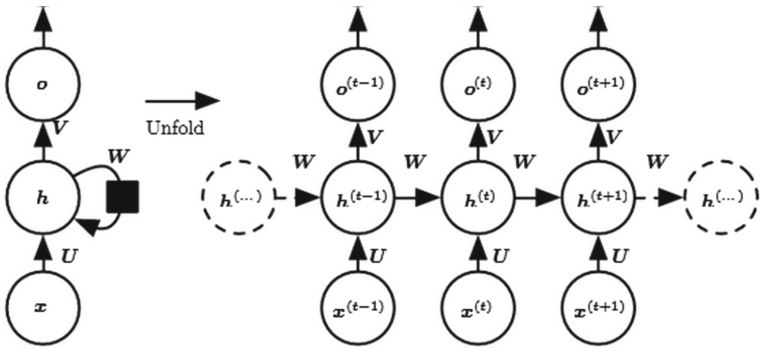


Fig. 4. Basic working of a simple class of RNNs neuron [9].

Equations 1a–1c formalize the operations indicated in Fig. 4, where we observe that the cell in the state $h^{(t-1)}$ that receives and processes the input $x^{(t-1)}$, in addition to transmitting the result as output ($o^{(t-1)}$) to the next layer, also uses it as input to update its own state to $h^{(t)}$. Therefore, when that same cell receives the next entry $x^{(t)}$, it has the memory of the previous entry $x^{(t-1)}$. And so the process continues, successively, until it processes all the input instances.

The activation function commonly used in recurrent layers is the hyperbolic tangent ($tanh$), to deal with nonlinearity of data. Equations 1b and 1c show, respectively, the weighted sum on which the activation function $tanh$ with the respective weight matrices is applied, and then the calculation of the output of each recurring neuron.

$$h^{(t)} = f\left(h^{(t-1)}, x^{(t)}\right) \tag{1a}$$

$$h^{(t)} = tanh\left(Wh^{(t-1)} + Ux^{(t)}\right) \tag{1b}$$

$$o^{(t)} = Vh^{(t)} \tag{1c}$$

The training algorithm normally used in RNNs is Backpropagation Through Time, adapted from the traditional backpropagation used in multilayer perceptrons, but, in this case, using the time instants as the basis for the backpropagation. It makes the calculations of the chain rule more complex, since the backpropagation of the error must reach not only the neurons of the previous layer in question, but all those who influenced them with the feedback through time. Thus, the large number of steps through which the error is retropropagated generates a drastic reduction in the value of the gradient, approaching zero, resulting in the so-called vanishing gradient problem, in which the weights begin not to suffer significant updates, not contributing then to the learning process [13].

The operation of the simple recurrent nets architectures, like Keras SimpleRNN, tested in this work, is based on the previously described functioning and does not deal with the vanishing gradient problem, which limits the memory offered by this type of nets to short term. This problem motivated improvements in the basic RNN algorithm, creating variations, among which the main and most used today are LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit), which will be the ones also used in this work.

4.1 LSTM

LSTM is a special type of RNN, which is able to learn long-term dependencies, solving the problem of vanishing gradient by adding more interactions in the processing of the recurrent neuron [12].

Instead of the single layer of simple RNNs, where the *tanh* function is applied on the weighted sum between the input and the previous state, LSTMs have more layers interacting in a very specific way, each one with its own matrix of weights [10]:

Forget Gate *f*: Defines how much of the previous state $h^{(t-1)}$ should be allowed to pass on. Activates the signal with sigmoid function (Eq. 2).

$$f = \sigma\left(W^{(f)}h^{(t-1)} + U^{(f)}x^{(t)}\right) \tag{2}$$

Input Gate *i*: Defines how much of the newly computed state with the x^t entry will be kept for the next instants of time. Activates the signal with sigmoid function (Eq. 3).

$$i = \sigma\left(W^{(i)}h^{(t-1)} + U^{(i)}x^{(t)}\right) \tag{3}$$

Output Gate *o*: Defines how much of the internal state $h^{(t-1)}$ is desired to be exposed for the next layer. Activates the signal with sigmoid function (Eq. 4).

$$o = \sigma\left(W^{o}h^{(t-1)} + U^{o}x^{(t)}\right) \tag{4}$$

Internal Hidden State *g*: It is computed based on the current entry $x^{(t)}$ and the previous state $h^{(t-1)}$ of the neuron, similar to what happened in the elements of Simple RNNs, using the function *tanh* (Eq. 5).

$$g = tanh\left(W^{(g)}h^{(t-1)} + U^{(g)}x^{(t)}\right) \tag{5}$$

After obtaining the results of *f*, *i*, *o* and *g*, the so-called Cell-State is calculated ($c^{(t)}$), given by Eq. 6, in which a combination is made between the long-term memories and the most recent ones, through the Forgot gate and the Input gate. This way, their values are weighted in order to ignore the desired memories (with value 0) or making them relevant (with value 1).

$$c^{(t)} = \left(c^{(t-1)} \otimes f\right) \oplus (g \otimes i) \tag{6}$$

The output h of the neuron is finally calculated as a function of the Cell-State value, after applying the *tanh* function to it, and then determining, through the value of the output gate, how much this output is relevant to this instant of time (Eq. 7).

$$h^{(t)} = tanh\left(c^{(t-1)} \otimes o\right) \tag{7}$$

4.2 GRU

It can be considered that the GRU is a variation of LSTM and very similar to it, with the advantage of having a considerably simpler gates structure than LSTM. GRUs are composed of only two gates: **Update** and **Reset**, which are able to be trained, respectively, to define how much of the older information should be kept, and to merge the new information entries with the previous memories. Equations 8a–8d summarize the computations that take place in GRU cells [10].

$$z = \sigma\left(W^{(z)}h^{(t-1)} + U^{(z)}x^{(t)}\right) \tag{8a}$$

$$r = \sigma\left(W^{(r)}h^{(t-1)} + U^{(r)}x^{(t)}\right) \tag{8b}$$

$$c = tanh\left(W^{(c)}\left(h^{(t-1)} \otimes r\right) + U^{(c)}x^{(t)}\right) \tag{8c}$$

$$h^{(t)} = (z \otimes c) \oplus \left((1-z) \otimes h^{(t-1)}\right) \tag{8d}$$

According to some works, LSTM and GRU are networks that have shown themselves to be very similar in terms of performance [15]. In this work, both of the architectures, as well as the Simple RNN, will be tested to see which will bring better results for the TSI forecast.

5 Results

Experiments were performed with Keras library for Python RNN architectures layers: SimpleRNN, LSTM and GRU.

As training and validation data set, it was used the period from 2011 November 05 to 2012 March 30, which presented records of quiet and disturbed sun activity. They had 6-h temporal resolution, making a total of 588 time instants, from which 80% were used as training set and 20% as validation set.

The networks fixed configuration involved: 40 input parameters, a recurrent layer and a linear output layer (Keras Dense layer) with a neuron, to predict a single TSI value. The nets parameters that were varied in the experiments were: number of recurrent hidden neurons, batch size, time steps, the use of Dropout (an algorithm to prevent the over-fitting problem in neural networks training process [1].) and the net architecture.

5.1 Varying Hidden Units

For all the architectures, experiments were made varying the recurrent layer hidden numbers, considering two experiment groups:

Few units without a dropout layer: For the first batch of experiments, few units were used, from 1 to 8, taking as basis [21]. For each different number of hidden units, five complete training processes were done, being adopted as fixed parameters: batch size of 5 and 1 time step. The Fig. 5 shows the least Validation Root Mean Square Error (RMSE) variation obtained in five training processes for each hidden unit number.

Many units with a dropout layer: For the second batch of hidden units experiments, 10 to 80 units were used, with 5 full training processes for each configuration. This time, a Dropout layer was added to the net, trying to prevent over-fitting, that was observed on the first group of hidden units tests. An initial dropout rate of 0.2 was assumed. The Fig. 5 shows the results for this experiment.

The first class of tests showed that low quantities of hidden units, specifically one unit, brought better accuracy results. However, the experiment with more units and the use of Dropout layer brought even better accuracy, besides of faster convergence with 20% of the epochs that were needed in the first group of tests.

The second group also presents less variability in RMSE for the same batch of tests.

In most of cases, in both groups of tests, specially in the second one, the GRU and LSTM nets performance looked very similar, bringing minor error than the SimpleRNN.

5.2 Varying Batch Number and Dropout Rates

For droptout rates varying tests, rates from 0% to 50% were used, considering that 50% is the higher recommended rate in Keras documents. The results with

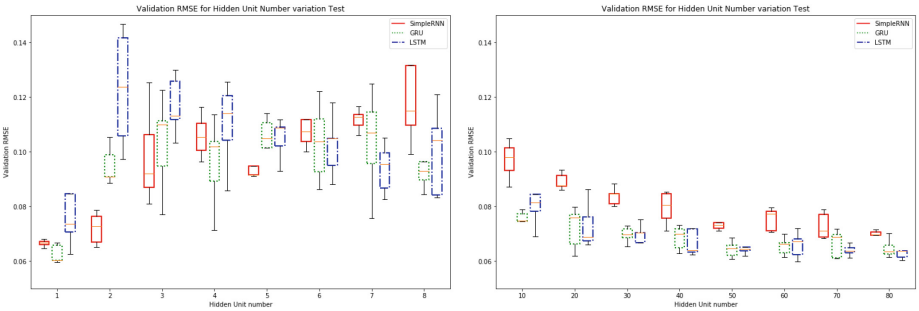


Fig. 5. Validation RMSE for hidden units number experiment. On the left, units from 1 to 8, and on the right, units from 10 to 80.

GRU and LSTM, the architectures which presented minor RMSE, did not show a considerable variance for the different dropout rates, but the higher rate showed a significantly less RMSE. Therefore, once a big number of hidden units was used, a rate of 50% was chosen for the subsequent experiments.

The batch number variation was defined calculating different percentages of the training data length. It started on 0.5% and was increased until 50%.

Tests were made for few as for many hidden units and both showed not a considerable difference in performance throughout, but a little worst performance over the batch size increasing in both test groups. Therefore, even though bigger size batches result in faster training processes, smaller batches were opted for the subsequent tests.

5.3 Varying Time Step

This set of tests was made varying the number of time steps in the input data composition. All previous tests used only one time step, what means that only the input at time t was being used for predicting the output at time $t + 1$ (6 h ahead).

When using more time steps, more previous time instants are used. For example, when using 3 time steps, for predicting the output at time $t + 1$, the input at times t, $t - 1$ and $t - 2$ are used.

Figure 6 shows the results obtained for the three net architectures (on the left) when varying the time step from 1 to 5. The same figure (right side) shows the performance comparison between the two architectures that showed best performance.

The tests for 2 and 3 time steps gave a minor value for RMSE, maybe suggesting time dependence between the Sun activity and TSI. More experiments should be done to confirm that relationship.

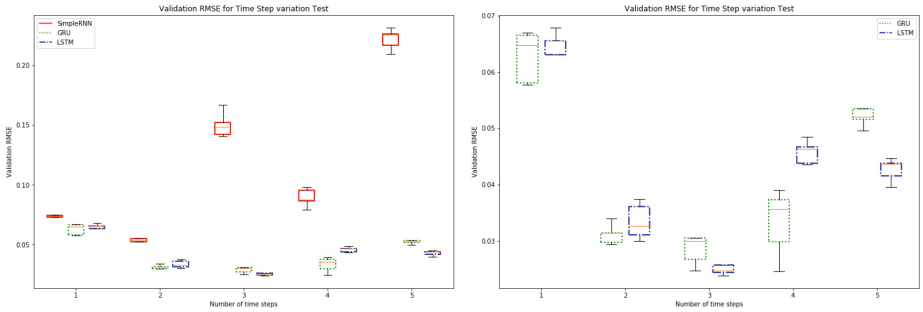


Fig. 6. Validation RMSE for time step varying experiment. On the left, the comparison between the three different architectures, and on the right, the comparison between only the GRU and LSTM.

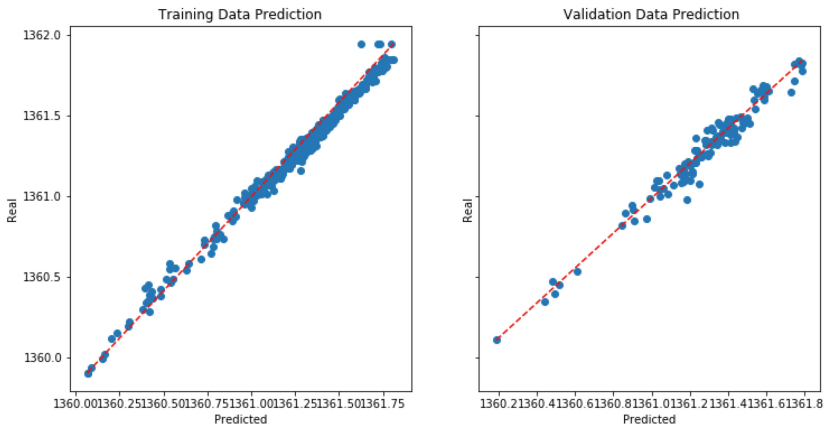


Fig. 7. TSI Training and Validation data prediction comparing the desired data to the obtained from GRU net.

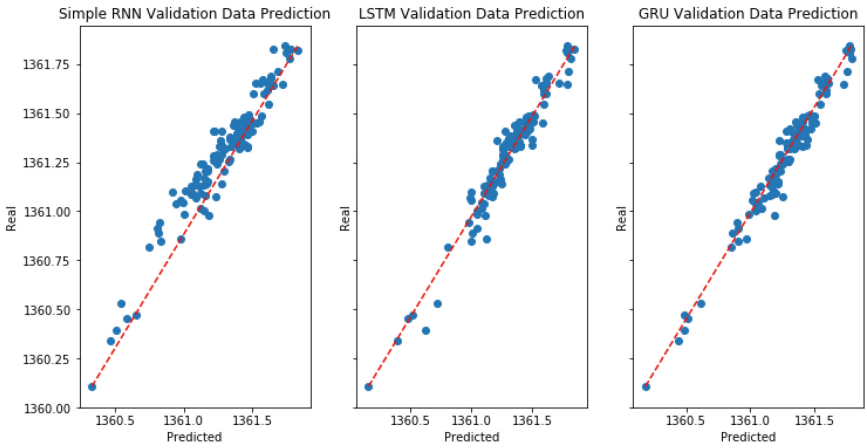


Fig. 8. TSI Validation data prediction comparing the performances between LSTM, GRU and SimpleRNN.

5.4 Forecast Accuracy

In an attempt to combine the best results obtained in the experiments described above, the last training steps were conducted using the considered best parameters.

Figure 7 shows, for one of these trainings, the relationship between the desired and obtained TSI values for the training and the validation data. Figure 8 shows the validation performance for each architecture for the same training parameters.

6 Conclusions and Future Works

In some experiments, the GRU net showed a slightly better performance, but in the most of them LSTM and GRU showed almost the same performance.

The SimpleRNN, for the first experiments, showed good results too, but in subsequent tests, it presented considerably worse performance than the two other architectures.

The time steps varying results suggest a better investigation over the time dependency period between the sun activity and its consequences at the top of the Earth's atmosphere.

Further tests are considered necessary to arrive at a final configuration for the TSI forecast. For that, it is intended to train a larger set of data after finding a satisfactory way of linking long periods of data, considering the existing gaps between them. The use of new practices to better analyze the network performance for different configurations is considered too for the continuity of this work.

There is the intention of continuing this work towards the mentioned proposal, having as a future step the testing of available tools in the cloud for turning available the workflow and its modular structure.

Parallel to this objective, the exchange of workflow modules, such as input images to images in other frequency bands is expected, in order to predict some of the SSI spectra instead of TSI. Afterwards, a change in the method of classification of disturbed regions is intended. The intention is to use Machine Learning techniques in all module exchanges.

Acknowledgements. This work was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

References

1. Baldi, P., Sadowski, P.J.: Understanding dropout. In: Advances in Neural Information Processing Systems, pp. 2814–2822 (2013)
2. Ball, W.T., Krivova, N.A., Unruh, Y.C., Haigh, J.D., Solanki, S.K.: A new SATIRE-S spectral solar irradiance reconstruction for solar cycles 21–23 and its implications for stratospheric ozone. J. Atmos. Sci. **71**(11), 4086–4101 (2014)
3. Burrell, A.G., Yeoman, T.K., Stephen, M., Lester, M.: Influence of solar irradiance on polar ionospheric convection. In: AGU Fall Meeting Abstracts (2016)
4. Coddington, O., Lean, J.L., Pilewskie, P., Snow, M., Lindholm, D.: A solar irradiance climate data record. Bull. Am. Meteorol. Soc. **97**(7), 1265–1282 (2016)
5. Echer, E., Rigozo, N.R., Nordemann, D.J.R., Vieira, L.E.A., Prestes, A., de Faria, H.H.: O Numero de Manchas Solares, Índice da Atividade do Sol. Rev. Bras. Ensino Física **24**(2), 157–163 (2003)
6. Ermolli, I., et al.: Recent variability of the solar spectral irradiance and its impact on climate modelling. arXiv preprint arXiv:1303.5577 (2013)
7. Fjukstad, B., Bongo, L.A.: A review of scalable bioinformatics pipelines. Data Sci. Eng. **2**(3), 245–251 (2017)
8. Gómez, J.R., Vieira, L.A., Dal Lago, A., Palacios, J., Balmaceda, L.A., Stekel, T.: Modelling short-term Solar Spectral Irradiance (SSI) using coronal electron density and temperature profiles based on solar magnetic field observations. Proc. Int. Astron. Union **12**(S327), 82–85 (2016)
9. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016). http://www.deeplearningbook.org
10. Gulli, A., Pal, S.: Deep Learning with Keras. Packt Publishing Ltd., Birmingham (2017)
11. Hathaway, D.H.: The solar cycle. Living Rev. Solar Phys. **7** (2010). https://doi.org/10.12942/lrsp-2010-1
12. Hochreiter, S., Schmidhuber, J.: LSTM can solve hard long time lag problems. In: Advances in neural information processing systems, pp. 473–479 (1997)
13. Hochreiter, S.: Recurrent neural net learning and vanishing gradient. Fuzziness Knowl.-Based Syst. Int. J. Uncert. **6**(2), 107–116 (1998)
14. Hoyt, D.V., Schatten, K.: The Role of the Sun in Climate Change. Oxford University Press, Oxford (1997)
15. Jozefowicz, R., Zaremba, W., Sutskever, I.: An empirical exploration of recurrent network architectures. In: International Conference on Machine Learning, pp. 2342–2350 (2015)

16. Kopp, G.: An assessment of the solar irradiance record for climate studies. J. Space Weather Space Clim. **4**, A14 (2014)
17. Krivova, N.A., Vieira, L.E.A., Solanki, S.K.: Reconstruction of solar spectral irradiance since the Maunder minimum. J. Geophys. Res.: Space Phys. **115**(A12), 1–11 (2010)
18. Leipzig, J.: A review of bioinformatic pipeline frameworks. Brief. Bioinf. **18**(3), 530–536 (2017)
19. O'Connor, B.D., et al.: The Dockstore: enabling modular, community-focused sharing of Docker-based genomics tools and workflows. F1000Research **6**, 52 (2017). https://doi.org/10.12688/f1000research.10137.1
20. Pinheiro, C.A.O., Senna, V.D.: Multivariate analysis and neural networks application to price forecasting in the Brazilian agricultural market. Ciência Rural **47**(1), 1–7 (2017)
21. Vieira, L.E.A., de Wit, T.D., Kretzschmar, M.: Short-term forecast of the total and spectral solar irradiance. arXiv preprint arXiv:1111.5308 (2011)
22. Williamson, C.E., et al.: Solar ultraviolet radiation in a changing climate. Nat. Clim. Change **4**(6), 434–441 (2014)
23. Yeo, K.L., Ball, W.T., Krivova, N.A., Solanki, S.K., Unruh, Y.C., Morrill, J.: UV solar irradiance in observations and the NRLSSI and SATIRE-S models. J. Geophys. Res. Space Phys. **120**(8), 6055–6070 (2015)
24. Yiğit, E., Knížová, P.K., Georgieva, K., Ward, W.: A review of vertical coupling in the atmosphere-ionosphere system: effects of waves, sudden stratospheric warmings, space weather, and of solar activity. J. Atmos. Solar-Terrestrial Phys. **141**, 1–12 (2016)