MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÕES

sid.inpe.br/mtc-m21d/2021/10.25.11.22-TDI

SOLAR IRRADIANCE PREDICTION: REPLICATING A WORKFLOW AND MAKING IT REPRODUCIBLE

Amita Muralikrishna

Doctorate Thesis of the Graduate Course in Applied Computing, guided by Drs. Rafael Duarte Coelho dos Santos, and Luis Eduardo Antunes Vieira, approved in October 21, 2021.

URL of the original document:
<http://urlib.net/8JMKD3MGP3W34T/45LRUHH>

INPE
São José dos Campos
2021

PUBLISHED BY:

Instituto Nacional de Pesquisas Espaciais - INPE
Coordenação de Ensino, Pesquisa e Extensão (COEPE)
Divisão de Biblioteca (DIBIB)
CEP 12.227-010
São José dos Campos - SP - Brasil
Tel.:(012) 3208-6923/7348
E-mail: pubtc@inpe.br

MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÕES

sid.inpe.br/mtc-m21d/2021/10.25.11.22-TDI

SOLAR IRRADIANCE PREDICTION: REPLICATING A WORKFLOW AND MAKING IT REPRODUCIBLE

Amita Muralikrishna

Doctorate Thesis of the Graduate Course in Applied Computing, guided by Drs. Rafael Duarte Coelho dos Santos, and Luis Eduardo Antunes Vieira, approved in October 21, 2021.

URL of the original document:
<http://urlib.net/8JMKD3MGP3W34T/45LRUHH>

INPE
São José dos Campos
2021

Cataloging in Publication Data

Muralikrishna, Amita.

M931s Solar irradiance prediction: replicating a workflow and making it reproducible / Amita Muralikrishna. – São José dos Campos : INPE, 2021.

xxx + 183 p. ; (sid.inpe.br/mtc-m21d/2021/10.25.11.22-TDI)

Thesis (Doctorate in Applied Computing) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2021.

Guiding : Drs. Rafael Duarte Coelho dos Santos, and Luis Eduardo Antunes Vieira.

1. Solar irradiance. 2. Recurrent neural network. 3. LSTM. 4. GRU. 5. Reproducibility. I.Title.

CDU 551.521.31:004.032.26

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS
Serviço de Pós-Graduação

DEFESA FINAL DE TESE DE AMITA MURALIKRISHNA
BANCA Nº 267/2021, REG63169/2017

No dia 21 de outubro de 2021, as 09h, por teleconferência, o(a) aluno(a) mencionado(a) acima defendeu seu trabalho final (apresentação oral seguida de arguição) perante uma Banca Examinadora, cujos membros estão listados abaixo. O(A) aluno(a) foi APROVADO(A) pela Banca Examinadora, por unanimidade, em cumprimento ao requisito exigido para obtenção do Título de Doutor em Computação Aplicada. As recomendações sugeridas pelos membros da Banca deverão ser incorporadas na versão final do manuscrito e será a responsabilidade dos Orientadores.

Título: “Solar Irradiance Prediction: Replicating a Workflow and Making it Reproducible ”.

Membros da banca:

Dr. Nandamudi Lankalapalli Vijaykumar - Presidente - INPE
Dr. Rafael Duarte Coelho dos Santos - Orientador - INPE
Dr. Luis Eduardo Antunes Vieira - Orientador - UNIFESP
Dr. Alisson Dal Lago - Membro Interno – INPE
Dra. Franciele Carlesso - Membro Interno – INPE
Dra. Ana Carolina Lorena - Membro Externo – ITA
Dra. Jenny Marcela Rodriguez Gomez - Membro Externo - SkolTech

---

Documento assinado eletronicamente por **Rafael Duarte Coelho dos Santos**, **Tecnologista**, em 22/10/2021, às 05:54 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Franciele carlesso (E)**, **Usuário Externo**, em 22/10/2021, às 08:30 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Ana Carolina Lorena (E)**, **Usuário Externo**, em 22/10/2021, às 10:39 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **nandamudi lankalapalli vijaykumar (E)**, **Usuário Externo**, em 25/10/2021, às 11:49 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Luis Eduardo Antunes Vieira**, **Pesquisador**, em 26/10/2021, às 08:30 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de

2020.

Documento assinado eletronicamente por **Alisson Dal Lago**, **Chefe da Divisão de Heliofísica, Ciências Planetárias e Aeronomia**, em 26/10/2021, às 10:46 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site http://sei.mctic.gov.br/verifica.html, informando o código verificador **8294234** e o código CRC **D15FB495**.

Referência: Processo nº 01340.007059/2021-73 SEI nº 8294234

"A ship in harbor is safe, but that is not what ships are built for."

JOHN A. SHEDD
"Salt from My Attic", 1928

Aos meus pequenos sóis: Neha, Nathan e Noah.

ACKNOWLEDGEMENTS

Esta tese não teria se concretizado se não fosse o apoio de algumas pessoas, que foram essenciais durante todo o caminho.

Agradeço à minha família, que sempre me apoiou de várias formas. Ao Binho por ter abraçado esse desafio junto comigo, aos meus pais por sempre incentivarem os meus estudos e reconhecerem o meu esforço, e por também terem ficado tantas vezes com as crianças para que eu pudesse ter momentos produtivos. Às minhas irmãs Aasita e Anoopa que, mesmo longe às vezes, sempre estiveram presentes. Aos meus sogros, que nunca deixaram faltar amor e cuidados aos pequenos. Aos meus filhos, meus pequenos sóis, que apesar de dias de atividade intensa, causando tempestades, foram luz, brilho e fontes essenciais de energia pra mim.

Agradeço ao meu orintador Rafael, pela confiança desde o início, pela disponibilidade de sempre, por deixar bem claro, desde o início, que a tese era minha e que o tema deveria ser algo que me agradasse e não imposta por ele. Obrigada, Rafael, pelo respeito e por abraçar as minhas opções. Ao Luis por me transmitir seu conhecimento, sempre com toda a paciência e serenidade, por estar também sempre disponível, e pelas boas risadas durantes as reuniões. Obrigada, Luis!

Aos meus colegas de curso, desde a época das disciplinas, agradeço pelas conversas e pelo companheirismo. Agradeço especialmente ao Adriano Almeida, ao Felipe Carvalho e ao Felipe Menino pelo grande suporte que me deram e por se fazerem "presentes", mesmo em tempos de pandemia.

Aos meus amigos que, cada um ao seu jeito, amenizaram e alegraram os dias mais difíceis. Em especial a Angeli, Kalyani, Salma e Aline, que foram ouvidos e abraços (mesmo que virtuais) quando mais eu precisei.

Agradeço ao Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, que me concedeu o afastamento por quatro anos, para me dedicar ao doutorado.

Apesar da pandemia de 2020 ter trazido muita dor a alguns, inasanidade a outros, e no mínimo um teste de resiliência para todos nós, trouxe também reflexões e aprendizados; ressaltou os significados dos valores mais simples e essenciais do nosso dia a dia.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

In english: This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

ABSTRACT

In times when computational resources - such as data, code, software tools, libraries, etc. - play a fundamental role in the development of scientific works, it has become evident that transparency regarding all the computational arsenal involved in such type of work is essential for its validation. This concern is the basis of the culture of reproducibility, which aims to add to a work the possibility of it being reproduced by an unknown person or by the author herself/himself in the future. Reproducibility can bring other benefits such as enabling the reuse and continuity of a work, which is associated with other terms such as replicability. This thesis is based on a workflow developed for solar irradiance prediction, and focuses on replicating it and adopting mechanisms to make the new workflow reproducible, as well as better exploiting recurrent neural networks for the prediction task. The prediction of the total solar irradiance at the top of the atmosphere would contribute, for example, in studies of solar variability, or could bring improvements to atmospheric and climate models on Earth; however, it is a service still not much explored by the scientific community in the area of space weather. The new version of the workflow was developed attempting to use free computational resources, such as the Python language and Linux operating system, and performs the prediction task using different recurrent neural network architectures from the Keras library. The work confirms the effectiveness of recurrent networks in predicting total solar irradiance and for one of the emission lines tested: *lyman-*α; and suggests that the prediction of other lines of the spectrum need additional parameters to obtain better accuracy. This document reports the replication process, presents the irradiance prediction results, and lists the computational resources employed to try to make the new workflow reproducible.

Keywords: Solar irradiance. Recurrent neural network. LSTM. GRU. Reproducibility. Replicability.

PREVISÃO DA IRRADIÂNCIA SOLAR: REPLICANDO UM FLUXO DE TRABALHO E TORNANDO-O REPRODUTÍVEL

RESUMO

Em tempos em que diversos recursos computacionais - como dados, códigos, ferramentas de *software*, bibliotecas, etc. - são utilizados para o desenvolvimento de trabalhos científicos, tornou-se evidente que a transparência sobre todo o arsenal computacional envolvido em um trabalho é essencial para a sua validação. Essa preocupação é a base da cultura da reprodutibilidade, a qual tem como objetivo, agregar a um trabalho a possibilidade de ele ser reproduzido por um desconhecido ou pelo próprio autor futuramente. A reprodutibilidade pode trazer outros benefícios como possibilitar o reaproveitamento e a continuidade de um trabalho, o que é associado a termos como a replicabilidade. Esta tese usa como base um workflow desenvolvido para a previsão da irradiância solar, e se concentra em replicá-lo e adotar mecanismos para tornar o novo workflow reprodutível, além de explorar melhor as redes neurais recorrentes para a tarefa de previsão. A previsão da irradiância solar total no topo da atmosfera contribuiria, por exemplo, em estudos da variabilidade solar, ou poderia trazer melhorias para modelos atmosféricos e climáticos na Terra; no entanto, é um serviço ainda pouco explorado pela comunidade científica da área de clima espacial. A nova versão do workflow foi desenvolvida buscando utilizar recursos computacionais gratuitos, como a linguagem Python e sistema operacional Linux, e realiza a tarefa de previsão utilizando diferentes arquiteturas de redes neurais recorrentes da biblioteca Keras. O trabalho confirma a eficácia das redes recorrentes na previsão da irradiância solar total e para uma das linhas de emissão testadas, a *lyman-*α; e sugere que a previsão de outras linhas do espectro necessitam de parâmetros adicionais para obter melhor acurácia. Este documento relata o processo de replicação, apresenta os resultados da previsão da irradiância e relaciona os recursos computacionais empregados para tentar tornar o novo workflow reprodutível.

Palavras-chave: Irradiância solar. Redes neurais recorrentes. LSTM. GRU. Reprodutibilidade. Replicabilidade.

LIST OF FIGURES

Page

LIST OF TABLES

Page

LIST OF ABBREVIATIONS

ANN	–	Artificial Neural Network
EMBRACE	–	Brazilian Studies and Monitoring of Space Weather (*Estudo e Monitoramento Brasileiro do Clima Espacial*)
EVE	–	Extreme Ultraviolet Variability Experiment
GRU	–	Gated Recurrent Unit
HMI	–	Helioseismic and Magnetic Imager
IDE	–	Integrated Development Environment
INPE	–	National Institute for Space Research (*Instituto Nacional de Pesquisas Espaciais*)
LASP	–	Laboratory for Atmospheric and Space Physics
LISIRD	–	LASP Interactive Solar Irradiance Datacenter
LSTM	–	Long Short-Term Memory
MAPE	–	Mean Absolute Percentage Error
MLP	–	Multi-layer Perceptrons
MSE	–	Mean Squared Error
R	–	Linear Correlation Coefficient
R^2	–	Coefficient of Determination
RNN	–	Recurrent Neural Network
SDO	–	Solar Dynamics Observatory
SOLSTICE	–	Solar Stellar Irradiance Comparison Experiment
SORCE	–	Solar Radiation and Climate Experiment
SSI	–	Solar Spectral Irradiance
TIM	–	Total Irradiance Monitor
TSI	–	Total Solar Irradiance
VE	–	Virtual Environment
VEM	–	Virtual Environment Manager

CONTENTS

Page

1 INTRODUCTION

1.1 Context

Solar activity is characterized by a variety of events that originate in its innermost layers and propagate to the outermost ones, often expelling matter and radiation towards the interplanetary medium. Therefore, it constantly influences its surroundings, mainly through matter, radiation and temperature that Sun emits as the result of its activity.

When it comes to Earth, this influence manifests itself in many ways outside and inside the planet's atmosphere, which motivates the so-called Space Weather and Space Climate area to continuously monitor the star and the many parameters involved in the Sun-Earth relationship. One such parameter is solar irradiance measured at the top (outside) of the Earth's atmosphere, and its prediction would improve the quality of its monitoring.

The underlying questions addressed are:

a) Is it possible to forecast the total and spectral solar irradiance?

b) How can the total and spectral solar irradiance be predicted based on solar surface magnetic field observations?

c) What is the time window that we can predict the Total Solar Irradiance (TSI) and the Solar spectral Irradiance (SSI)?

d) How to check if a workflow developed for irradiance prediction is valid?

e) How could this workflow be reused and improved?

In this work, we show that recurrent neural networks can be employed to forecast/predict the evolution of the total and spectral solar irradiance in a time window from six hours to three days.

1.2 Motivation

1.2.1 Solar irradiance prediction

Solar irradiance is an important parameter for studies of both solar variability and the layers of the Earth's atmosphere (HAIGH, 1996). For these reasons, for decades

its measurement has been collected by equipment aboard space missions, and also empirical and semi-empirical models have been created to reconstruct long periods of irradiance values (KRIVOVA et al., 2003; VIEIRA et al., 2011a; BADESCU, 2014; YEO et al., 2015; Rodríguez Gómez et al., 2016; GÓMEZ, 2017; Rodríguez Gómez et al., 2018).

Some of these models use solar parameters at the same instant of time as that at which irradiance is to be estimated, and would then allow the tracking of irradiance variation in real-time. Irradiance prediction could mainly become an option to feed atmospheric models, like the one proposed by Nogueira et al. (2015), and predict ionosphere conditions hours or days ahead. The prediction could also contribute to the previous monitoring of solar variability, and could help filling in, or reconstructing, periods of missing data. Although irradiance prediction can make such contributions, it is a service not yet offered to the Brazilian scientific community and is still not much explored worldwide.

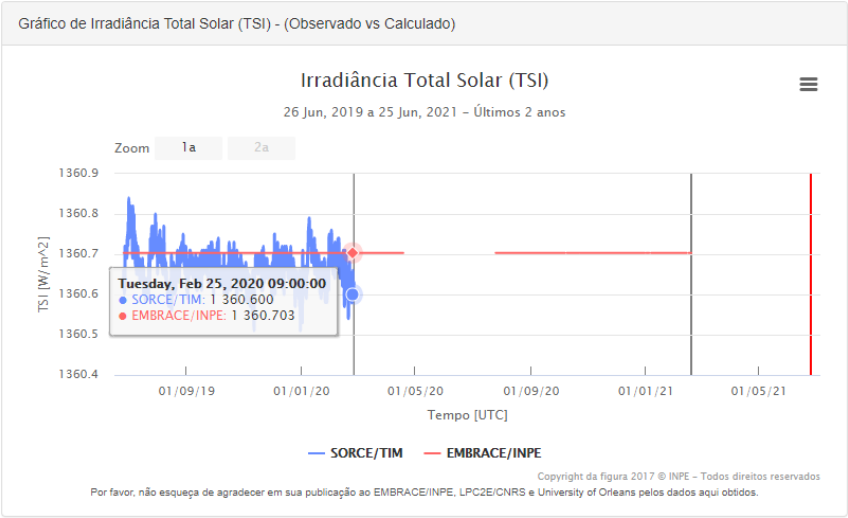
There was a proposal of offering this service on a web platform, through the web page of the Brazilian Studies and Monitoring of Space Weather (EMBRACE[1]), a program of the National Institute for Space Research (INPE). A procedure devised and developed by Vieira et al. (2011b), which accurately predicts the total solar irradiance, was used to implement and offer the service.

However, the procedure was not made continuously available as a service. Figure 1.1 shows the discontinuation of the service on EMBRACE's web page, in which the goal was to provide total solar irradiance (TSI) values predicted by recurrent neural network hours ahead and allow their comparison to the values collected by the TIM/SORCE instrument in near real-time.

This work replicates the procedure proposed by Vieira et al. (2011b), using a different language from the original version, to validate and improve the techniques used there. Also, a new workflow was generated and it would be helpful to reproduce it in different programming languages.

[1]EMBRACE's web page: `http://www2.inpe.br/climaespacial/portal`, accessed on September 2021.

Figure 1.1 - Discontinuation of the short-term total solar irradiance forecasting service.



SOURCE: INPE - National Intitute for Space Research (2021)

1.2.2 Reproducibility

The original procedure has been developed in a proprietary platform, which brings limitations to its reproduction and reuse. This limitation, which will be better discussed in the following chapters, was the second motivation for this thesis, as it raised questions about the reproducibility of scientific works. We believe that a reproducible procedure is more likely to be implemented and continued.

The workflow developed for solar irradiance prediction can be used as an example in which the lack of reproducibility can interrupt a scientific work or bring barriers to its continuity. The reasons behind these difficulties can be diverse, among which some can be cited, such as:

- The work is to be re-executed by people different from those who participated in its development, who may not find the readability and structure understanding to fully comprehend the procedure;
- The work must be continued or re-executed with computing resources (such as hardware, software, operating systems) different from those that were used initially to develop the work, which may cause problems such as lack of compatibility or non-availability of some software in the required version;
- The software dependencies (such as packages and libraries) used during

the workflow development, in their appropriate versions, are not known or are no longer available, not allowing the procedure to be run properly, or

- The data used may no longer be available, making it difficult to validate the results generated in the original work.

Scientific works that make use of computational elements, such as data, code, and other resources, often fail to add value to science when it is not at least verifiable. If there is no concern with reproduction while conducting a particular work, it may happen that not even the authors themselves will be able to reproduce it sometime later. Graphs and analyses, for example, may have no validity if the data and codes that led to such results are not made available as well.

Nowadays, many works declare themselves to be reproducible, when in practice they are not. Attempts to reproduce works often result in failure, for a variety of reasons. This thesis reviewed definitions for reproducibility and other related terms, such as replicability, accessibility, and others, and discussed the efforts and benefits involved in making a work reproducible and in reproducing works.

1.3 Objectives

1.3.1 General objectives

The main aim is to develop and validate a total and spectral solar irradiance forecast model based on artificial recurrent neural networks (RNN).

1.3.2 Specific objectives

This work has as specific objectives:

a) Perform the translation of the workflow developed by Vieira et al. (2011b) from *Matlab* to *Python*, so it can be executed on free software platforms.

b) Exploit RNNs architectures to the solar irradiance prediction task.

c) Make available the new version of the workflow.

d) Use concepts and tools to expand the workflow, and thus make it flexible to integrate modern machine learning techniques for TSI and SSI prediction, and assess its performance and prediction quality.

Besides the previous objectives, it is expected that the discussion about reproducibility will contribute to the readers' gradual awareness of the importance of reproducibility in scientific works, especially nowadays, when so many computational resources are used and, at the same time, hidden in the publication of research results. The inclusion of the concern of reproducibility in the development of scientific works, even if gradually, would enable those works to be better made available, validated, and reused.

1.4 Document organization

This document was designed to organize the content studied and the results generated during the thesis in a logical structure. The chapters were organized as follows.

Chapter 2 opens the document by defining and contextualizing solar irradiance as an important parameter that takes part in physical interactions involving the Sun and the Earth. This relation becomes the basis for understanding the importance of monitoring the solar irradiance at the top of the atmosphere, and the benefit that its prediction would bring to related studies. In the same chapter, the data used in this work and the equipment that collects it are described.

Chapter 3 introduces concepts, terms, and scenarios related to the reproducibility of scientific works. It focuses on computational reproducibility and on the factors involved in the task of making a computational work available so that it can be reproduced.

In the sequence, the Chapter 4 first describes the original workflow and its specifications. It then reports the steps followed in the replication of the original workflow, focusing on the first part of the processing, i.e. the step prior to the prediction. The chapter also describes the obstacles encountered during the migration, and the efforts made to make the new workflow reproducible. The features added to the new version of the workflow are listed at the end of the chapter.

Before describing the experiments related to solar irradiance prediction (which is done in Chapter 6), Chapter 5 gives a brief introduction to machine learning and describes recurrent neural networks and their architectures. After listing the network architectures chosen for this work, the chapter presents the corresponding models for them available in the Keras library for use with the *Python* language. The parameters that were explored to tune the neural networks and the metrics that were used to evaluate their performance as well as the quality of the results were also listed in

this chapter.

Chapter 6 describes the main prediction experiments performed with recurrent networks, comparing different architectures, and predicting total irradiance and some spectral emission lines as well. A comparison between the RNN prediction and physical models reconstructions is also performed.

Closing the document, general considerations, and suggestions for future works are presented in Chapter 7.

2 SOLAR IRRADIANCE

This chapter will give a brief presentation of the Sun-Earth relationship, which will be the basis for the introduction of one of the parameters involved in this relationship, solar irradiance. The Sun and its activity will be the first subject to be described, in order to understand the great influence that it exerts on the Earth. Based on this relationship, the studies of solar irradiance and the importance of its monitoring for space weather and Earth atmosphere will be defined.

2.1 The Sun

The Sun is a star whose composition is of approximately 90% Hydrogen (H) and 10% Helium (He). There is still another 0.1% Carbon (C), Nitrogen (N), and Oxygen (O). The temperature on the different layers of the Sun can vary from thousands to millions of degrees Celsius. In its core, the gases that compose the star are almost completely ionized, forming a plasma, due to the very high temperature and pressure conditions. This condition triggers thermonuclear fusion reactions that result in the high energy power emitted by the Sun. The energy produced in the solar core is transported through internal layers: the so-called radiative and convective zones. The convective zone is the most external layer of the solar interior, through which the energy generated in the core is transported by convection[1] currents to the "surface", where it is released in the form of radiation (CECATTO, 2003).

It can be said that the great protagonist of the various phenomena that occur on the Sun is the Sun's magnetic field, built by the solar dynamo, which moves from the bottom of the convection zone (SOLANKI et al., 2006). The solar activity is characterized by a set of phenomena, such as sunspots, flares[2], coronal mass ejections[3] (or CMEs). These phenomena, among others, are triggered mainly due to the complexity of the solar magnetic field, combined with the Sun's differential rotation, which varies according to the star's latitude and depth (SCHRÖTER, 1985). As a consequence, solar activity determines the characteristics of, for example, the solar wind, the solar cycle, and the variability of solar irradiance, as well as other diverse effects felt on Earth. Some of these terms will be briefly described in this

[1] The convection process consists of the transfer of heat by the circulation or movement of the heated parts of a liquid or gas.

[2] A solar flare is an intense burst of radiation that occurs in the solar atmosphere

[3] The coronal mass ejections are large eruptions of high-temperature ionized gas observed at the Sun's corona. The expelled gas constitutes part of the solar wind, and when it hits the Earth's magnetic field, it can cause geomagnetic storms, damaging, for example, communications and power stations.

chapter.

The structures resulting from solar activity influence all the space that compose the heliosphere[4], and can expel plasma and part of the magnetic field lines towards the interplanetary medium (Figure 2.1). With the interaction of this matter with the planets' magnetic fields which it encounters along the way, especially the closest ones, like the Earth, the magnetosphere if formed formed.

A constant flow, called solar wind (Figure 2.1), brings along the structures resulting from solar activity and influences life on Earth in different ways, which include the atmosphere's behavior, climate, temperature, and tide (PARR et al., 2005). One of the greatest examples of this influence is the formation of the magnetosphere. It is an almost static structure resulting from the interaction of the solar wind with the geomagnetic field which has a characteristic shape, and acts as a stage for physical processes as a precise result of this interaction. Figure 2.1 illustrates the magnetosphere, which acts as a protective shield formed around the Earth, deflecting much of the matter brought by the solar wind.

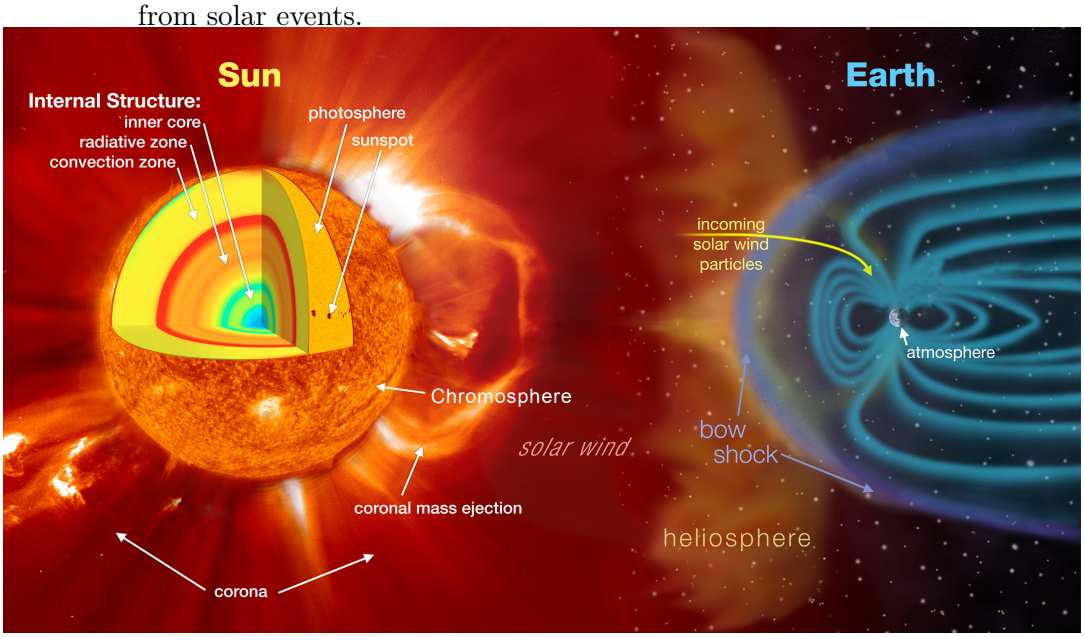
Despite this protection provided by the magnetosphere, intense events that occur on the Sun, such as the CMEs, can eventually interfere - directly or indirectly - in the conditions of the planet activities, depending on their characteristics. When part of the matter that travels with the solar wind penetrates the magnetosphere, there may be impacts, which include electronic failures or even loss of satellites in the atmosphere, communication and navigation problems in airplanes, radiation risks to astronauts, power interruption in homes and commercial buildings, to name a few. On the other hand, events such as the phenomenon of aurora borealis can occur as a non-harmful consequence of this interaction.

The study of the Sun-Earth relationship and its causes and consequences is the focus of the research area called Space Weather, which works towards a constant monitoring of the star. The aim is to deeply understand the phenomena that occur on the Sun, and thus investigate their relationship with terrestrial events, as well as the consequences of solar activity that can negatively interfere in the atmosphere and life on Earth.

Figure 2.2 presents a composition of images of two elements studied by Space Weather: the CME and the aurora borealis. The CME is represented by a com-

[4]Heliosphere is the region of space, which covers the solar system, in which the solar wind has a significant influence.

Figure 2.1 - On the left, a graphic representation of the Sun and some of its inner and outer layers. Still on the left, a solar wind, sunspots, and CME representation. On the right, a representation of the magnetosphere and one of the points where solar wind's charged particles can eventually enter carrying matter resulted from solar events.



SOURCE: Adapted from NASA (2021a).

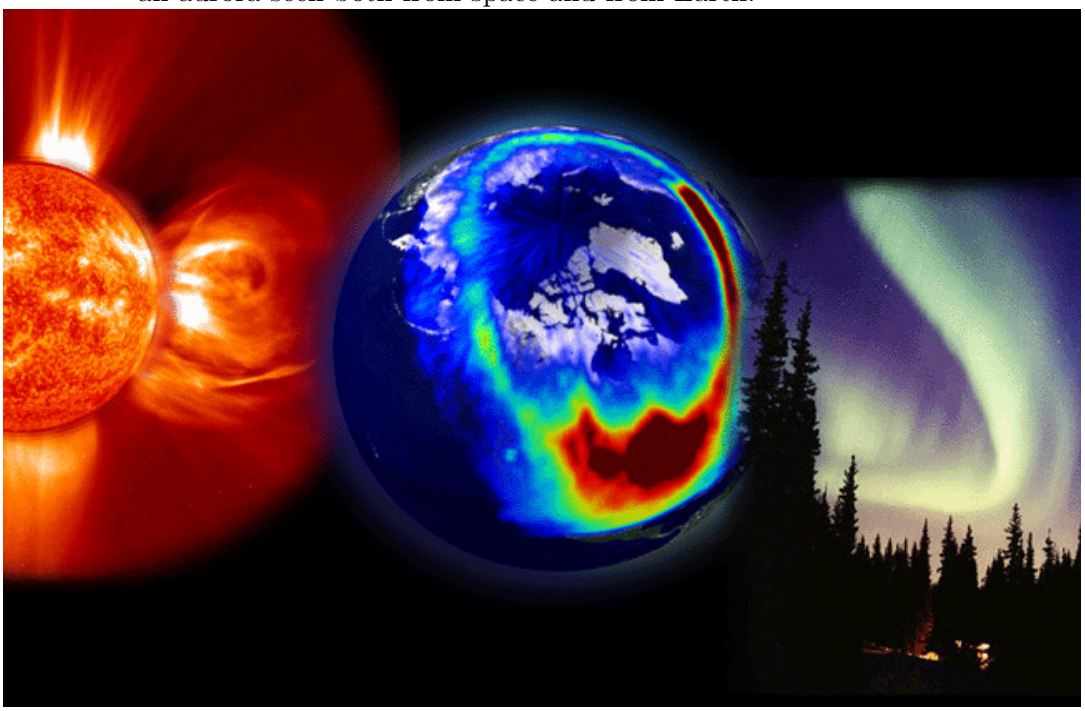
position of SOHO[5](Solar and Heliospheric Observatory) images (DOMINGO et al., 1995), and the aurora borealis is represented by registries made both from space and from Earth. The image represents an intense solar storm that occurred on July 14^{th}, 2000, in distinct phases: the explosion on the Sun, followed by the spread of charged particles over the United States, and the resulting aurora in Alaska.

2.2 Solar cycle

Solar missions have monitored the Sun by means of instruments capable of collecting either images or numerical data. Images can reveal active regions, sunspots, and prominence related to solar events, among other structures and features that can be observed inside or around the solar disk. Numerical data present characteristics of the interplanetary environment that are closer to (but still outside) the Earth.

[5]SOHO's webpage: https://sohowww.nascom.nasa.gov

Figure 2.2 - Records of two of the main events monitored by Space Weather: a CME, and an aurora seen both from space and from Earth.



SOURCE: Max Planck Institute for Solar System Research (2021).

But long before technology enabled the shipment of solar monitoring missions into space, the Sun was already being monitored in other ways. For example, temperature and radiation were measured and recorded in the 18^{th} century by an instrument called *heliothermometer* (VINCENZI; FASANO, 2020). Records of the presence of spots on the solar disk were made in the early 19^{th} century (NICHOLSON, 1958). Although, reports indicate that Galileo made a naked-eye observation of a sunspot in the 17^{th} century (VAQUERO, 2004; GALILEI et al., 1982), there are naked-eye sunspots catalogs available in the literature even previous to that (WITTMANN; XU, 1987; YAU; STEPHENSON, 1988).

Naked-eye observation of the sunspots has occurred at least since 1000 B.C. in ancient China (EDDY, 1976), and through telescopes in Europe since 1611 (EDDY, 1976; SCHOVE, 1983). Nowadays, besides the naked-eye observations, the sunspots are observed through Sun monitoring instruments such as the SDO (Solar Dynamics Observatory) (PESNELL et al., 2011).

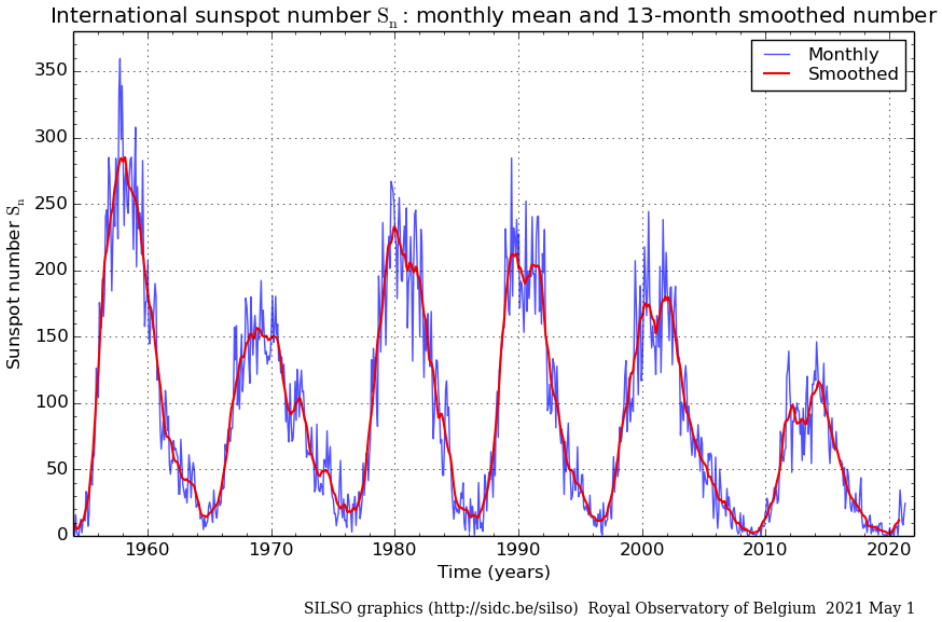
Through the sunspot records, a periodicity in solar activity was observed and called the solar cycle. It lasts approximately 11 (eleven) years and relates mainly to the

growth and reduction in the number of sunspots over time (NICHOLSON, 1958; STUIVER; QUAY, 1980).

Figure 2.3 shows a graph of the occurrence of sunspots over the years, proving the periodicity of solar cycles. Each solar cycle starts and ends in a calm phase of activity - called solar minimum - characterized by a decrease in the number of sunspots; and reach, in between, the intense phase of solar activity - called solar maximum - characterized by an increase in the occurrence of sunspots (HATHAWAY, 2015).

Solar cycles are identified with sequential numbering and their duration is estimated according to the observation of their maximum and minimum periods. The previous cycle, number 24, had its maximum between the years 2014 and 2015, and its second minimum was reached in December 2019. Since the beginning of 2020, the Sun has been in cycle 25, still in a period of minimum.

Figure 2.3 - Number of sunspots observed over the years (the monthly averages in blue, and its smoothed curve in red), suggesting a cycle of approximately 11 years.



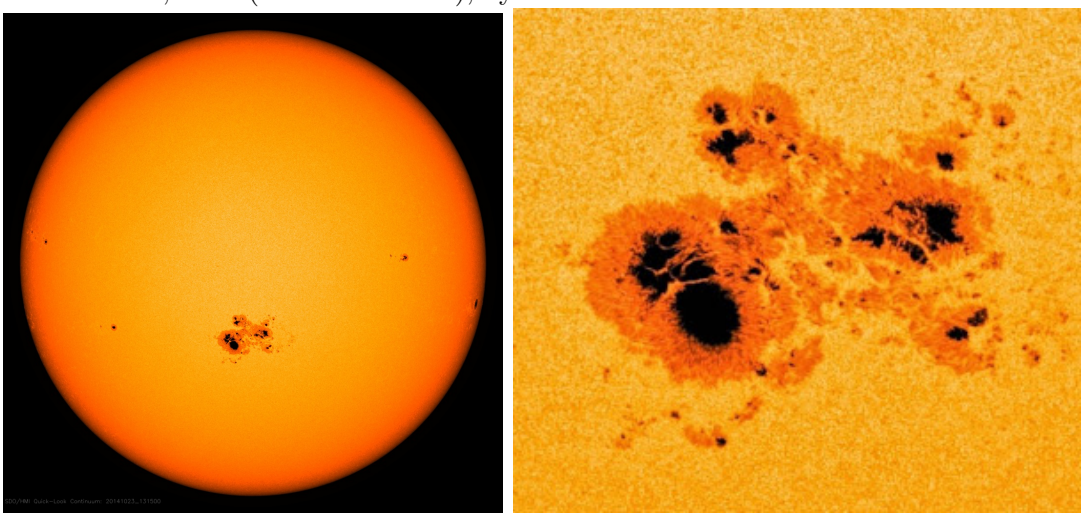
SOURCE: Solar Influences Data Analysis Center (2021).

2.3 Sunspots and active regions

Sunspots are distinguished on the solar surface because they are darker and cooler regions, emitting less energy than the rest of the solar surface. The sizes vary: a large sunspot can cover an area 700 times the size of the Earth's surface area (HOYT; SCHATTEN, 1997).

They are formed by two regions (as shown in Figure 2.4): the umbra, which is the central and darkest region, with a temperature around 4100K; and the penumbra, which can be found in only half of the spots, and is present around the umbra, having a lighter and grayish color, with a diameter that generally reaches 2.5 times that of the umbra (ECHER et al., 2003). They have intense magnetic fields (about 1000 times more than the normal solar surface), which are brought to the surface during the solar cycle, when they get the appearance of spots.

Figure 2.4 - Sunspot AR2192, the largest one in solar cycle 24, was recorded on October 21, 2014 (solar maximum), by HMI instrument aboard SDO mission.



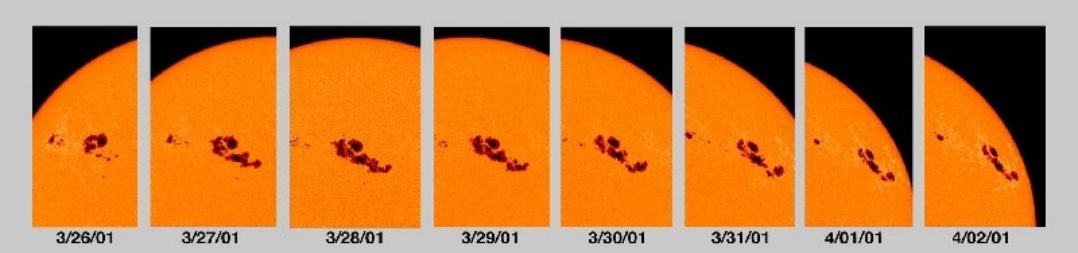
On the left, it can be found in the image original size, and on the right, zoomed in. The zoomed image allows the areas of umbra and penumbra to be better distinguished.

SOURCE: Schou et al. (2011).

Each spot (or a set of them) can be identified by the active region to which it belongs so that its evolution can be tracked along the Sun's rotation, whose duration is shorter in the solar equator region (27 days) and longer in the pole region (around 36 days). The record of one of the largest sunspot groups, in the Active Region 2192 (or AR2192), can be seen in Figure 2.4. An example of the evolution of another

sunspot group (in AR9393) over eight consecutive days can be seen in Figure 2.5.

Figure 2.5 - Evolution of a sunspot group (the largest in its solar cycle) labeled AR9393, which caused several flares and CMEs between March 26th and April 2nd, 2001.

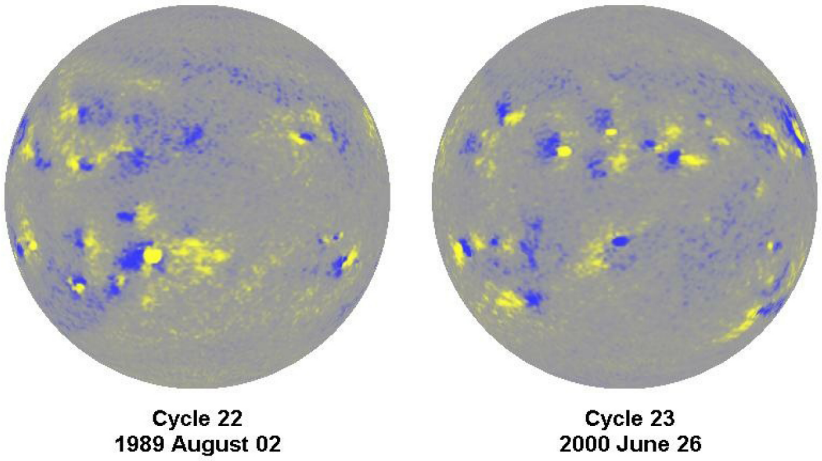


SOURCE: Solar and Heliospheric Observatory (2021).

The passage from one cycle to another is characterized, with few exceptions, by the appearance of sunspots that present the opposite polarity to their predecessors. The sunspot's polarity can be visualized through the active region images, in which two distinct colors are used to identify opposite poles. It is then possible to observe that, when the sunspots appear in binary groups, they appear in one solar hemisphere with inverted polarity to their corresponding sunspots in the other hemisphere (HALE et al., 1919), as shown in the magnetograms of Figure 2.6. On the left, a magnetogram represents sunspots of cycle 22 (August 2, 1989) with positive pole in yellow and negative pole in blue; and, on the right, a magnetogram with the same marking represents sunspots of cycle 23 (June 26, 2000). Note that the major spots in one hemisphere have opposite magnetic polarity to those in the other hemisphere, and the signs are inverted from one cycle to the next (HATHAWAY, 2010).

Sunspots appear in regions where there is an evident perturbation of the solar magnetic field. Less perturbation may not generate spots but can be observed through magnetograms that show the active regions.

Figure 2.6 - The main spots in one hemisphere have opposite magnetic polarity to those in the other hemisphere, and the signs are reversed from one cycle to the next.



SOURCE: Hathaway (2010).

The definition of the so-called active regions (or ARs), which can be identified before the appearance of sunspots, has suffered changes with the deepening of the study and the knowledge obtained about them. D'Azambuja (1953), for example, initially defined the ARs as the union of all the visible phenomenon that come accompanied by the birth of sunspots. Later, Kiepenheuer (1968) changed D'Azambuja's term "visible" to "observable" and extended its connection with sunspots, relating ARs to observations that precede, accompany, and follow the birth of the spots.

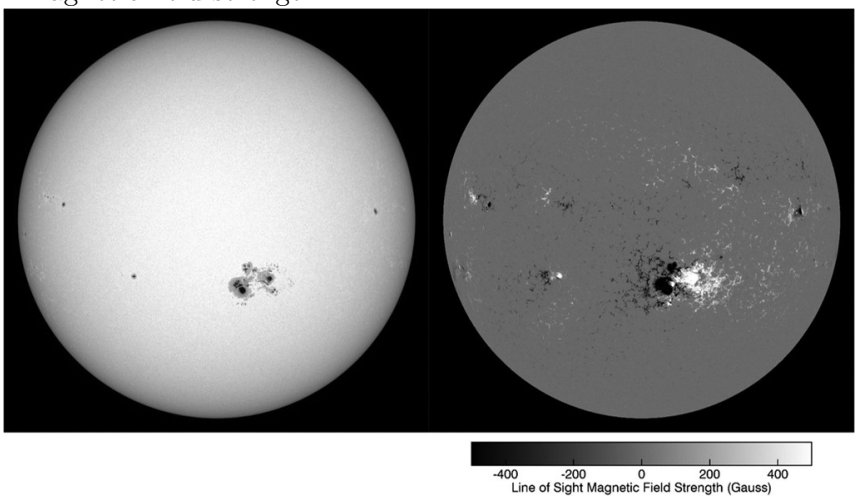
Nowadays, the sunspots' evolution can be followed by observing the Sun in a range of wavelengths. Some of the bands allow the association of the ARs presence not only to the presence of sunspots but mainly to the perturbation of the solar magnetic field (DRIEL-GESZTELYI; GREEN, 2015). This condition allows a more recent definition of ARs given by Driel-Gesztelyi and Green (2015) as:

> "...the totality of observable phenomena (...) represented by the extension of the magnetic field from the photosphere to the corona, revealed by emissions over a wide range of wavelengths (...) accompanying and following the emergence of strong twisted magnetic flux (...) through the photosphere into the chromosphere and corona."

Figure 2.7 shows the October 23rd, 2014 solar record from the SDO satellite's HMI

instrument through two images, a white light image and a magnetogram, where it is visualized that the active regions exhibiting magnetic fields with opposite signs have a larger area than the corresponding sunspots. The presence of the sunspots occurs in the areas where the magnetic field is most concentrated and intense, but the AR extends over the entire area where the field disturbance is visible (DRIEL-GESZTELYI; GREEN, 2015).

Figure 2.7 - Record of Sun on October 23rd, 2014, from the HMI/SDO instrument: on the left, a group of sunspots in a white light image, and on the right a magnetogram showing the corresponding active regions in a range of the solar magnetic field strength.



SOURCE: Strong et al. (2017).

2.4 Solar activity monitoring

Monitoring the solar cycle phases allows us to monitor the influence of the solar activity on the Earth over long periods, such as months or years. The study of this influence over shorter periods, such as days or hours, is supported by different types of measurements and recordings, such as images of the Sun in its visible part or at different wavelengths, measurements in the interplanetary field (before structures reach the Earth), in the Earth's upper atmosphere, in all the layers below, or on the ground. Each of these environments provides its own conditions that must be considered for the type of information to be obtained.

For example, an image of the Sun, if collected outside the Earth's atmosphere, will present different characteristics compared to one collected on the ground, because

this image would suffer interference from all the layers of the Earth's atmosphere, which would serve as filters, due to their physical characteristics. At the same time, data collected onboard orbiting satellites can also suffer interference, as caused, for example, by instabilities, loss of calibration, wear or damage to the collection instruments, which are sometimes in the way of high-energy structures that affect their operation.

Space missions have aimed to collect data for short and long-term follow-ups, such as images and measurements that allow studies of processes that occur on Earth in shorter periods of time - such as geomagnetic storms - and in longer periods, such as the impact of solar activity on climate change on the planet. Other studies consider the Sun as a model star whose characteristics will make it possible to understand the behavior of other stars more distant in the universe.

The studies of the evolution in the number of sunspots, for example, are long term studies, since they allow associating past solar cycles with the current one and the ones to come. The evolution of a specific spot, or of a certain group of spots, along with the evolution of the corresponding active regions, can be studied in the short term, considering the possibility of structures resulting from these regions being associated with disturbances in the geomagnetic field or changes in climatic conditions such as temperature or radiation received by the Earth.

Studies that focus on the radiation coming from the Sun consider that the most intense portion of the energy is emitted by the visible layer of the solar atmosphere - the photosphere - considered the "surface" of the Sun, on which sunspots are observable with the naked eye. The photosphere and the main layers of the Sun are illustrated in Figure 2.1.

The solar activity is characterized by all the phenomena already mentioned in this chapter, which occur in the different layers of the Sun, in addition to several others not mentioned. The evolution of these phenomena, which modify the Sun's status in short term, is called "solar variability" (GRAY et al., 2010). The occurrence and evolution of sunspots and active regions are two indicators of solar activity that are recorded in the photosphere and represent part of this variability.

Parallel to this scenario, the energy that the Earth receives from the Sun is what drives life and various processes on the planet. The Sun-Earth relationship and its consequences, therefore, require constant monitoring of solar variability in an attempt to track solar activity in the short term and thus predict its possible effects

on Earth. One way to study the influence of solar radiation on the Earth is through the radiation received at the top of the Earth's atmosphere.

2.5 Solar irradiance

In general, the radiation emitted by any electromagnetic radiation source can be described quantitatively by the following terms:

- Radiant energy: is the total energy emitted in a radiative process. It is measured in Joules (J).
- Radiant power (or flux): is defined as the total power of radiation emitted by a source, transmitted through a surface, or incident on a surface. It is measured in Joules per second (J/s) or Watts (W).
- Irradiance: is considered to be the amount of radiant power incident on a surface per unit area. Given in Watts per meter squared (W/m^2).

Since the main process of energy transfer in the Earth's atmosphere is the electromagnetic radiation coming from the Sun, solar irradiance can be defined as the amount of solar radiant energy that is received by the Earth's atmosphere, and considered an important indicator of the effects of solar variability on the Earth (HATHAWAY, 2015).

When measured at the top of the Earth's atmosphere, studies consider it an influential parameter on the properties of the layers of the Earth's atmosphere and the effects suffered as a consequence of perturbation of these layers. As an example, it can be cited the disturbance in the communication signals from ground to satellites, due to the perturbation of the Ionosphere (VIEIRA et al., 2011b). After the definition of solar irradiance, other motivations for monitoring this parameter will be exposed.

Solar irradiance can be measured and used in two different formats: Solar Spectral Irradiance (SSI) and Total Solar Irradiance (TSI), which are described below.

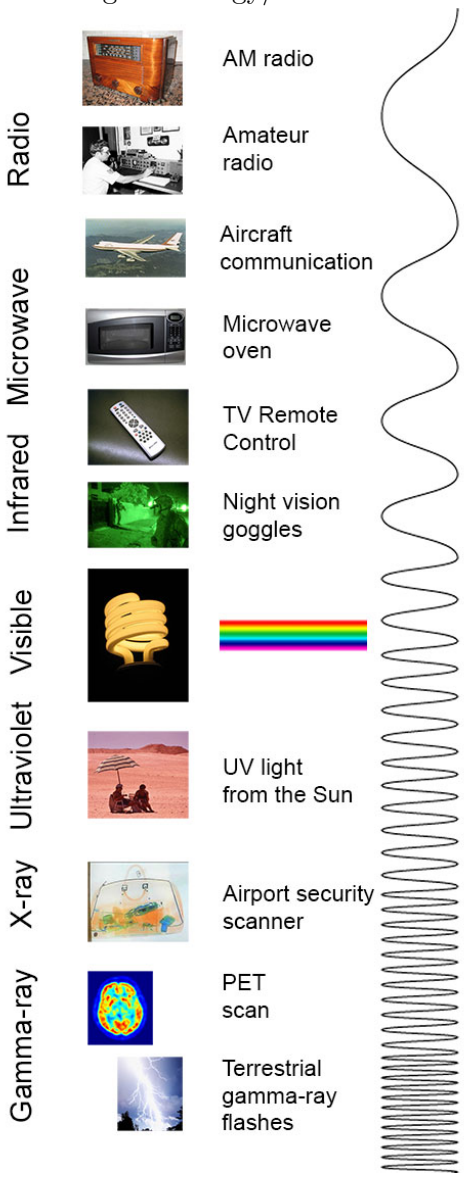
2.5.1 SSI

Before understanding spectral solar irradiance, it is necessary to understand what an electromagnetic spectrum is: an electromagnetic spectrum[6] (Figure 2.8) is the

[6]Spectrum is the intensity of light transmitted at different wavelengths.

range of all types of electromagnetic radiation, visible[7] and non-visible, according to the frequency and the characteristic wavelength of each radiation: x-rays, ultraviolet light, infrared radiation, microwaves, radio (LIOU, 2002).

Figure 2.8 - The electromagnetic spectrum from the lowest energy/the longest wavelength (at the top) to the highest energy/the shortest wavelength (at the bottom).



SOURCE: NASA (2021b).

[7]Visible light: a region of the electromagnetic spectrum that is in a frequency range sensitive to the human eye, which does not respond to larger electromagnetic wave frequencies, such as those of ultraviolet light, or smaller ones, such as that of infrared radiation.

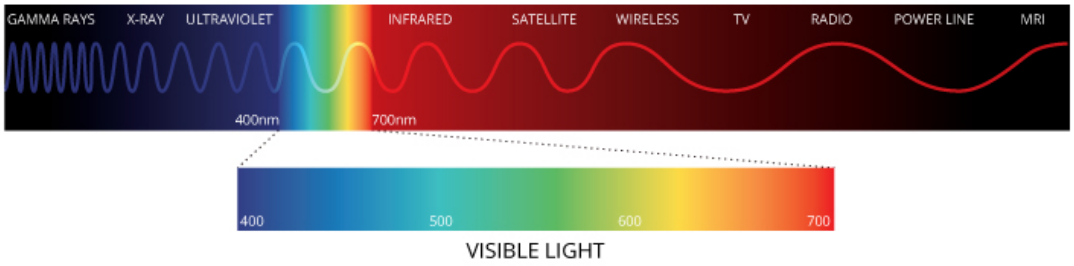
Solar radiation is transmitted through electromagnetic waves and can then be analyzed as an electromagnetic spectrum, formed by a continuous arrangement of the wavelengths mentioned in the previous paragraph (ECHER et al., 2001). It can be considered that in the visible and infrared wavelengths, the radiation emitted by the photosphere is practically constant, while in shorter and longer wavelengths than visible light, as is the case of the radiation emitted by the Sun's outer layers, it is variable (BRASSEUR; SOLOMON, 2006). Figure 2.8 separates the full range of the spectrum into the main types of emission waves, associated with their most popular occurrences.

SSI is defined by that range of wavelengths covered by solar radiation received by the Earth at a distance of 1 AU (Astronomical Unit), considered to be the average distance between the Sun and Earth, outside the Earth's atmosphere.

Some spectral ranges of solar irradiance are considered relevant for climate studies on Earth, as their variability has been found to impact climate modeling. Other bands are interesting for studying patterns in solar variability (ERMOLLI et al., 2013).

For this work, we used different spectral emission lines, which will be introduced in the next chapters. Figure 2.9 covers the range of the spectrum received from the Sun in *nm* (nanometer), highlighting the visible range.

Figure 2.9 - The electromagnetic radiation spectrum collected from Sun is covered in this figure, in *nm*: from extreme ultraviolet, passing through the visible range and reaching near-infrared wavelengths.



SOURCE: FONDRIEST (2021).

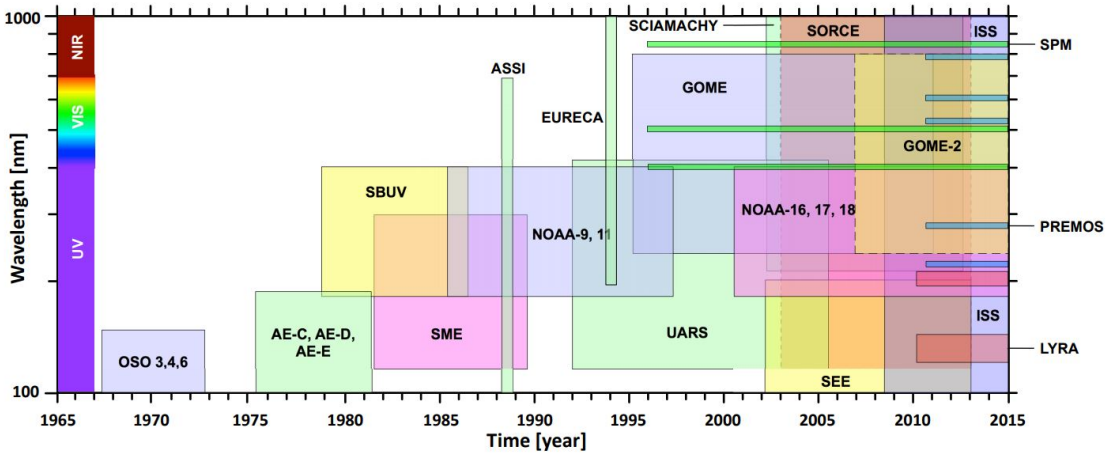
Satellite measurements of SSI have existed since the 1960s. However, there are many gaps in the observations, as the instruments that measure it have their detectors degraded over time and therefore a limited lifetime. Existing records include measurements at EUV (extreme ultraviolet), UV (ultraviolet), VIS (visible range), and

NIR (near infrared) wavelengths.

SSI's measurement history is also limited due to the need to have an instrument to observe every wavelength of the spectrum. Figure 2.10 shows that each mission focuses on specific bands, which makes the coverage of the entire spectrum small. It also shows a survey of the history of the major missions, as well as the wavelength range of the SSI data collected (above $100nm$) and the duration of each mission.

The variation of SSI, at its different wavelengths, influences the chemical reactions and dynamics of the Earth's atmosphere at different altitudes. Its measurement history and continuous observation are very important for understanding this influence on the layers of the atmosphere and consequently for monitoring climate change on Earth. However, the large gaps in the sequence of measurements make this kind of study difficult.

Figure 2.10 - Survey of the main satellite missions that have collected SSI at wavelengths longer than $100nm$.



SOURCE: Ermolli et al. (2013).

2.5.2 TSI

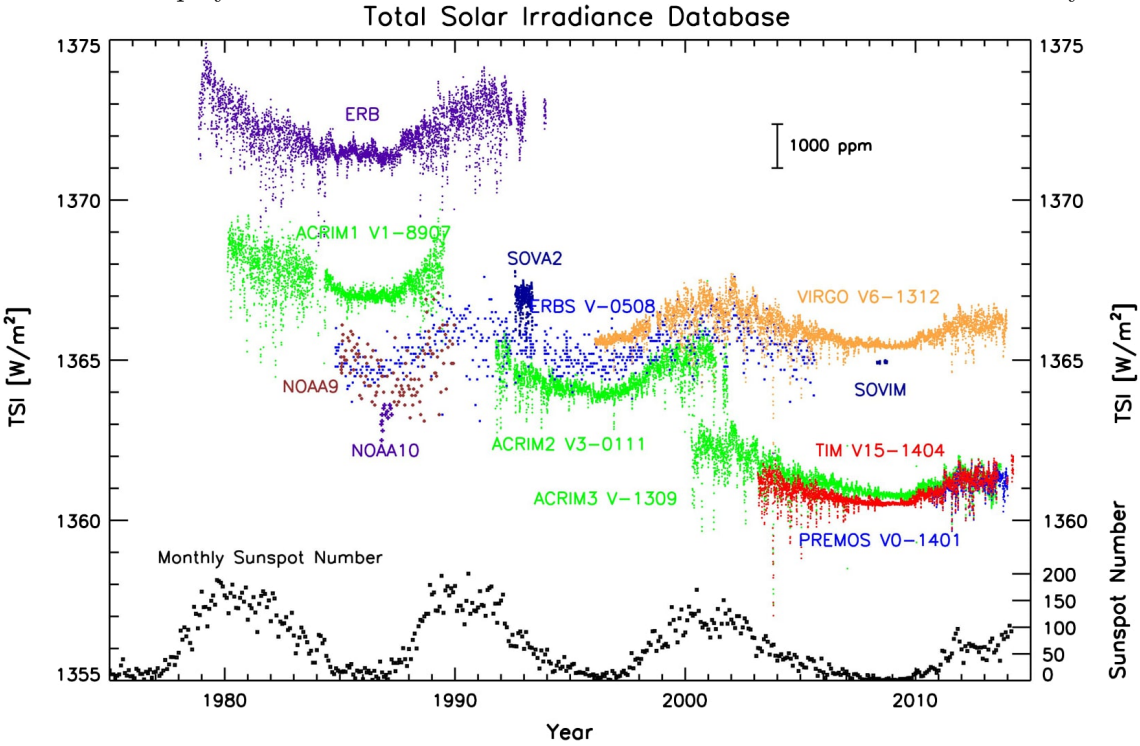
The total solar irradiance can be defined as the radiation emitted by the Sun in all spectral regions, analogous to the integral of all spectral bands of the SSI, incident per second in $1m^2$ at 1 AU.

TSI measurements have historically been performed by ground instruments, and

aboard aircraft, balloons, ships, and also satellites. Analysis of records of TSI measurements made by different instruments has shown that higher accuracy is obtained when measurements are made above the Earth's atmosphere, by instruments aboard satellites (PARR et al., 2005).

In the case of TSI, there is also the problem of degradation of the measurement equipment, which in some cases even requires mission interruption for equipment maintenance. In addition, the calibration of each piece of equipment interferes with the standardization of measurements between different pieces of equipment, which affects the existence of a continuous history of measurements in the same standard. Figure 2.11 shows the discrepancy between the calibrations of different types of equipment, which demands the need for TSI reconstructions for long-term studies.

Figure 2.11 - TSI measurement history that shows the calibration difference between one instrument and another. The monthly average sunspot number is also displayed as a basis to show that overall the TSI varies with solar activity.



SOURCE: Kopp (2014).

2.5.3 Solar irradiance reconstruction and prediction

The irradiance, when measured on the ground, is different from when it is measured at the top (outside) of the Earth's atmosphere, since its emission at different wavelengths varies after interacting with the physical and chemical properties of the layers, mainly the ionized ones, of the Earth's atmosphere (YIĞIT et al., 2016).

For some areas of research, such as oceanography, the amount and level of irradiance diffusion reaching the sea is of interest to marine biology evolution studies. In botany, on the other hand, an example of a study is the influence of irradiance, at different times of the day, and on days of high and low rates, on the production of starch in certain types of plants (PILKINGTON et al., 2015). In a different context, in engineering, there are studies that analyze the influence of spectral , in different bands and in different seasons, on the performance of photovoltaic technologies and energy (DIRNBERGER et al., 2015; WANG et al., 2015). Another type of long-term study reveals a direct association of the solar activity, more specifically of sunspot number and solar wind properties, on the production rate of a type of carbon in the Earth's upper atmosphere (STUIVER; QUAY, 1980).

Studies on terrestrial weather and climate, another research topic at INPE, also recognize the great influence of solar irradiance on climate (HAIGH et al., 2010), inclusively on the creation of climate and atmospheric models (CODDINGTON et al., 2016; ERMOLLI et al., 2013; WILLIAMSON et al., 2014).

In this scenario, some works have tried to model or perform the prediction of solar irradiance reaching the ground, based on parameters such as ambient temperature, cloud cover, among others (WANG et al., 2012; DIAGNE et al., 2013; YANG et al., 2014; VOYANT et al., 2017). However, for studies of the earth's atmosphere and climate, it would be important to predict the irradiance before it reaches the planet's atmosphere, an effort that is not yet evident in the literature.

This work will therefore focus only on solar irradiance measured at the top of the Earth's atmosphere, relevant to studies of the atmosphere itself, which suffers its constant influence (PUDOVKIN, 2004; LIU et al., 2011; BURRELL et al., 2016; NINA et al., 2011), a topic that is part of the study context of research groups at the National Institute for Space Research (INPE), to which this thesis is linked.

Besides being important for the study of atmospheric and climatic phenomena and characteristics on Earth, solar irradiance is also an allied parameter in the study

of the activity of the Sun itself (FRÖHLICH, 2013), which consequently raises the possibilities of what the activity of other more distant and not easily monitorable stars would be like.

However, the continuous collection of irradiance data has some shortcomings that cause the data to be missing in certain periods. Problems such as the temporary failure of the instruments or the difference in calibration between measurements of different instruments have motivated the creation of physical models that are able to reconstruct long periods of continuous missing solar data both in total emission and in several emission lines of the spectrum (KRIVOVA et al., 2010; BALL et al., 2014; YEO et al., 2015; Rodríguez Gómez et al., 2016; YEO et al., 2014; YEO et al., 2017).

Historical measurements of solar irradiance can contribute to short and long-term studies that allow, for example, the analysis of solar activity during some of its cycles, over a time period of years or decades; or some short-term survey that requires real-time monitoring related, for example, to some specific solar event or phenomenon (FRÖHLICH, 2013; PIEDEHIERRO et al., 2014; ANDRADE; TIBA, 2016; WOODS et al., 2006).

The class of semi-empirical models estimates the solar irradiance also with the purpose of filling the measurement gaps and is based on the observation of magnetograms and continuum images (FLIGGE et al., 1998; KRIVOVA et al., 2010; BALL et al., 2011; FONTENLA et al., 2011). Other models reconstruct solar irradiance at past time periods as functions of solar parameters measured at that same instant of time (KRIVOVA et al., 2003; VIEIRA et al., 2011a; BADESCU, 2014; YEO et al., 2015; Rodríguez Gómez et al., 2016).

Irradiance prediction could benefit some cited studies, which need irradiance values for decision-making or as part of the input to their models. The great advantage of the prediction task over the reconstruction task is the fact that prediction allows estimating future data, while reconstruction can be applied only to past periods or, at most, to the present. For example, solar irradiance prediction could be a differentiating factor in models, for example, ionospheric or mid-atmosphere models (SUKHODOLOV et al., 2017; NOGUEIRA et al., 2015), which could be improved by using irradiance data estimated hours or days ahead.

Considering the gains that irradiance prediction could bring, there are efforts to search for a prediction model with good performance, which offers an accuracy close to those offered by reconstruction models. Vieira et al. (2011b) developed a process

for predicting TSI and SSI based on solar photosphere images, work which was used as a basis for this thesis.

Even if forecasting is about future values, there is nothing initially that prevents forecasting systems from being used for reconstruction as well. Considering this, in this work we also compare some irradiance prediction results to the physical models SATIRE (YEO et al., 2014) and EMPIRE (YEO et al., 2017) reconstructed values.

2.6 Instruments and data

For decades, the Sun-Earth relationship has been the target of missions that aim to observe and record this interaction by collecting images at different frequencies and measuring parameters at different points in the space between them. Some satellites have been launched with the focus on observing the Earth from space, in order to study the behavior of its atmosphere, oceans, and ecosystem; and others have focused on the Sun, to study its activity, its variability, and its influence on the processes involved with it on Earth. This thesis uses data from two such missions: from SDO, and from SORCE (Solar Radiation and Climate Experiment). Both missions will be presented in the following subsections.

2.6.1 SDO

Launched in February 2010, the Solar Dynamic Observatory[8] is a mission linked to NASA's Living With a Star program, directed at understanding the causes of solar variability and its impacts on Earth, by studying the solar atmosphere on short spatial and temporal scales and at various wavelengths. Its main objective is to study the structure of the solar magnetic field and how it evolves into the other layers of the solar atmosphere and into the interplanetary medium in the form of the solar wind, energetic particles, and variations in solar irradiance (PESNELL et al., 2011).

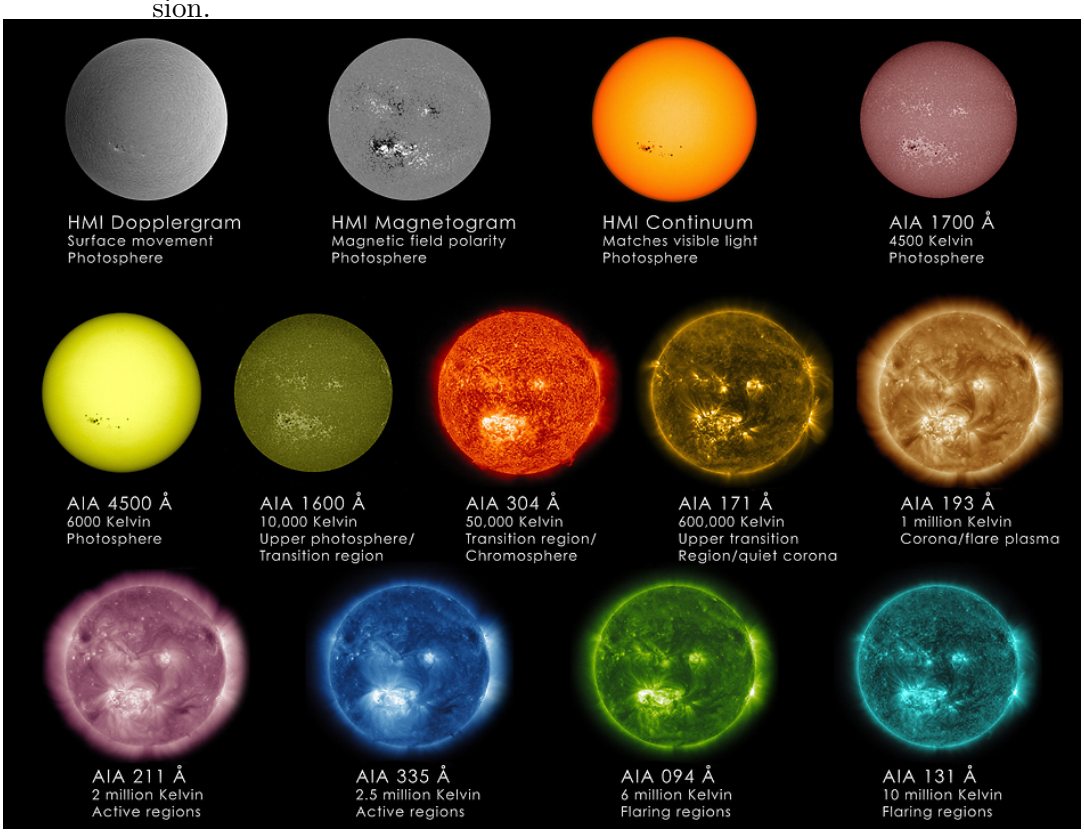
The satellite carries with it three instruments: the HMI (Helioseismic and Magnetic Imager) (SCHOU et al., 2011), the AIA (Atmospheric Imaging Assembly) (LEMEN et al., 2011) and the EVE (Extreme Ultraviolet Variability Experiment) (WOODS et al., 2010).This work uses data collected by HMI, which was developed in order to allow the study of magnetic field oscillations on the solar surface, providing a continuous image coverage of the entire solar disk, revealing the presence of sunspots and active regions, at a high spatial resolution (1024x1024 and 4096x4096, the first was the one

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

used in this work).

Some examples of images provided by the SDO HMI instruments can be seen in Figure 2.12, along with images from the AIA. Images in different wavelengths evidence different solar layers and events at the same time instant. It is also possible to observe the formation of sunspots in the same locations where active regions are formed. Active regions generally begin to appear before the spots, as the spots represent a "deeper" condition of the magnetic field perturbation in a given region.

Figure 2.12 - Sample images collected by the HMI and AIA instruments of the SDO mission.



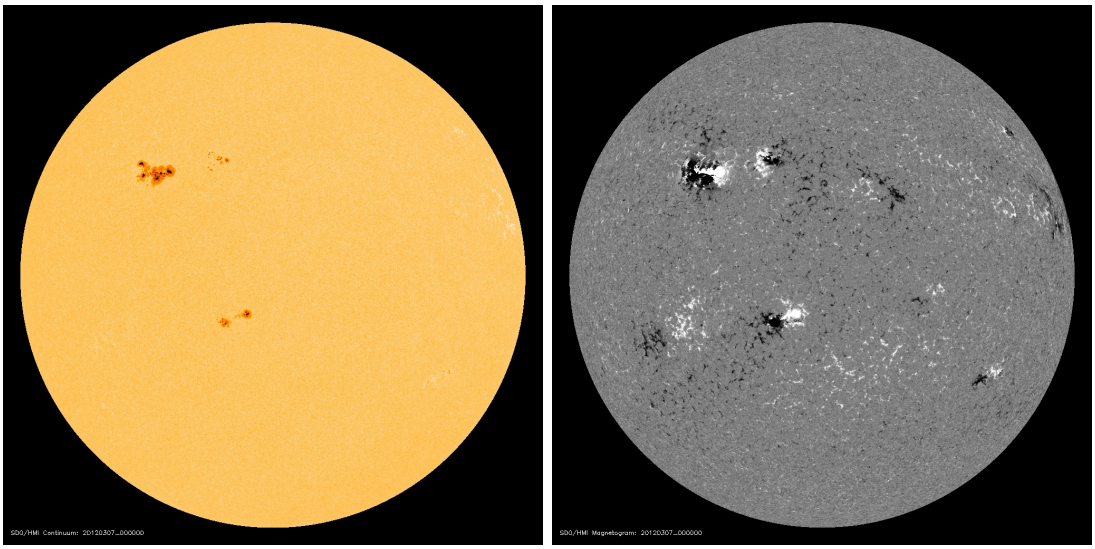
SOURCE: NASA (2021c).

For this thesis a data set formed by the following pair of HMI images was used:

- **Continuum image**: displays the visible range of the solar photosphere and in which it is possible to visualize the formation of sunspots separated into umbra and penumbra regions [Figure 2.14(a)].

- **Magnetogram**: displays the map of the Sun's magnetic field also in the photosphere in grayscale, which represents the magnetic field intensity over the entire solar disk and also the polarity of the active regions' magnetic field lines [Figure 2.14(b)].

Figure 2.13 - Sample of the HMI/SDO instrument observation of 00h00 on March 7th, 2012. On the left a continuous image and on the right a magnetogram.



(a) Continuum Image (b) Magnetogram

SOURCE: Joint Science Operations Center (JSOC) / Science Data Processing (SDP) (2021).

Both types of imagery are found as products of the SDO satellite available in the Joint Science Operations Center (JSOC) - Science Data Processing (SDP)[9] webpage. The version chosen is the *Quick Look*, available in 15-minute time intervals and in 1024 x 1024 pixel resolution, over which a preprocessing has already been applied to correct any distortions suffered at the edges. Other versions available with other size resolutions and collection intervals may be options for future testing.

Each pair of images for a given instant of time will be used to compose a map of the active regions and spots present in the inner area of the solar disk circumference at that instant, which will serve as input for the prediction technique that will determine the value of solar irradiance hours or days ahead, which is desired as the final result.

[9]JSOC webpage: `http://jsoc.stanford.edu/`

The SDO also makes available, through the EVE instrument, extreme ultraviolet (EUV) solar irradiance measurements with the spectral resolution, which will be used to compose the SSI dataset in this work, along with SORCE onboard instrument data. The main objective of the EVE is to contribute to the advancement of the understanding of EUV solar irradiance variations based on the activity of solar magnetic features. The time series used from EVE were downloaded from the LISIRD Data Center[10] (LASP Interactive Solar Irradiance Data Center), from the LASP[11] (Laboratory for Atmospheric and Space Physics) laboratory, which is responsible for the instrument.

2.6.2 SORCE

Launched in January 2003, the SORCE[12] was a NASA-funded mission that provided the state of the art in incoming measurements of X-ray, ultraviolet, visible light, near infrared (at wavelengths totaling 95% of the contribution to TSI), and TSI, aiming at studies such as long-term climate variations and climate prediction (ROTTMAN et al., 2014).

The satellite carried with it four instruments: the SIM (Spectral Irradiance Monitor) (HARDER et al., 2005), the SOLSTICE (Solar Stellar Irradiance Comparison Experiment) (MCCLINTOCK et al., 2005), the TIM (or Total Irradiance Monitor) (KOPP; LAWRENCE, 2005) and the XPS (XUV Photometer System) (WOODS et al., 2005).

For this thesis, the following data collected by SORCE were used:

- TSI (in W/m^2): of the TIM equipment, in temporal resolutions of 6 hours and 1 day;
- SSI (in $W/m^2/nm$): of XPS, SIM and SOLSTICE equipment to monitor the irradiance evolution in different spectral bands, with daily temporal resolution.

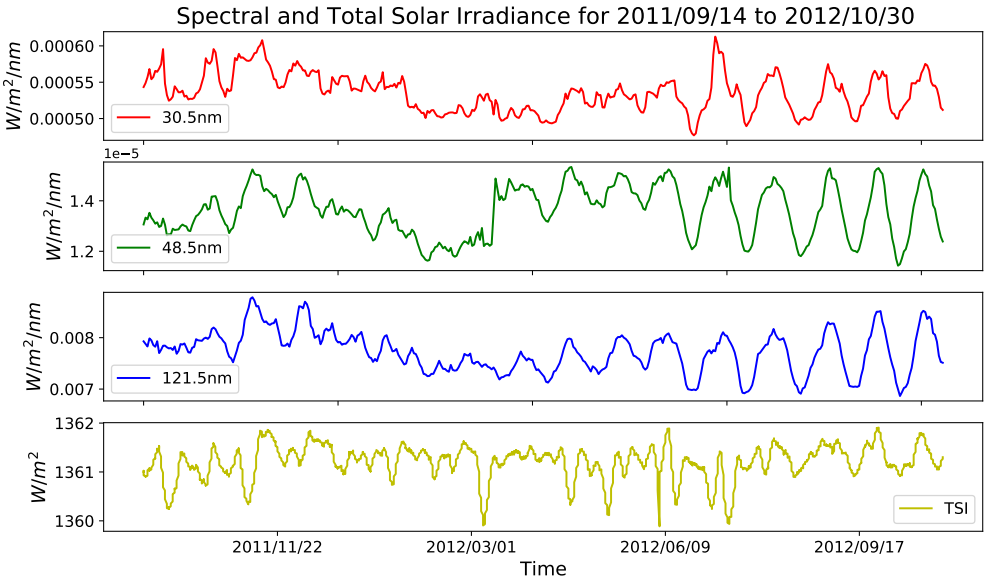
SORCE's TSI and SSI data are also available on LISIRD Data Center. The mission, after 17 years of TSI and SSI data collection, was completed in February 2020, discontinuing the solar irradiance data provision.

As an example of the irradiance data used in this work, Figure 2.14 shows the SSI

[10]LISIRD webpage: https://lasp.colorado.edu/lisird/
[11]LASP webpage: https://lasp.colorado.edu/home/eve/science/
[12]SORCE webpage: http://lasp.colorado.edu/home/sorce/

three emission lines and the TSI data for a time period of 413 days (from 2011/09/14 to 2012/10/30). TSI data was collected by TIM/SORCE, the line 30.5nm data was collected by XPS/SORCE, the 48.5nm was collected by EVE/SDO, and the line 121.5nm was collected by SOLSTICE/SORCE.

Figure 2.14 - A sample of irradiance data used in this work, collected from instruments aboard different missions.



3 REPRODUCIBILITY

Low reproducibility has been an issue in focus in scientific communities in different areas. In areas that make extensive use of computational methods and artifacts, it has been realized that reproducible work is not only limited to making the computational codes available since the final results are a function of a number of intermediate details that are not captured by traditional publishing approaches. This chapter will explore the main concepts and main tools surrounding reproducibility that founded the creation of the new version of the workflow for total and spectral solar irradiance prediction.

3.1 Reproducibility in scientific works

Before any discussion of reproducibility in science, it is important to agree on some meanings of terms that will be used frequently in this chapter and that may have divergent meanings in the literature (NATURE, 2016).

Even the term *reproducibility*, besides having multiple definitions, is still an ambiguous concept, and sometimes confused with terms such as replicability, repeatability, among others (PLESSER, 2018; GOODMAN et al., 2016). It can be said that some of the concepts that will be listed below are complementary to reproducibility, others can be considered to go parallel to it, and still, others may be a consequence of it. More terms will be introduced in other sections of this chapter so that they are aligned with the proper context; for now, the definitions of *reproducibility* and *replicability* have been chosen, which are initial concepts that should be well understood for the rest of the content involved in the topic.

Reproducibility: In general, in the scientific context, it is the characteristic that scientific work has that allows it to be reproduced (or remade) by the researchers who produced it, as well as by others who are interested in remaking it, by using the same elements - data, methods, techniques - and achieving the same results (GOODMAN et al., 2016).

Replicability: is the characteristic added to a work that provides enough details to allow it to be replicated. A work is said to be replicated when it has provided details of models and methods it has employed, to such a level that the adoption of new data, new code (or software), or a new analytical look has been able to achieve results close enough to the original results, confirming the original work (ROUGIER et al., 2017). Disagreements revealed in a replication process can have different reasons

and lead to different findings, from the evidence of missing information in the process employed in the original work, failures or biases identified in the original work (PATIL et al., 2016), up to such good points as scientific discoveries in a new dataset, for example (NASEM, 2019).

Even though they have meanings that are relatively easy to understand (we say "relatively", because perhaps the ease of understanding may be directly associated with the context of the science in which the terms are interpreted), the topic surrounding the application of the concepts of reproducibility and replicability in scientific work is still discussed in a very divergent way.

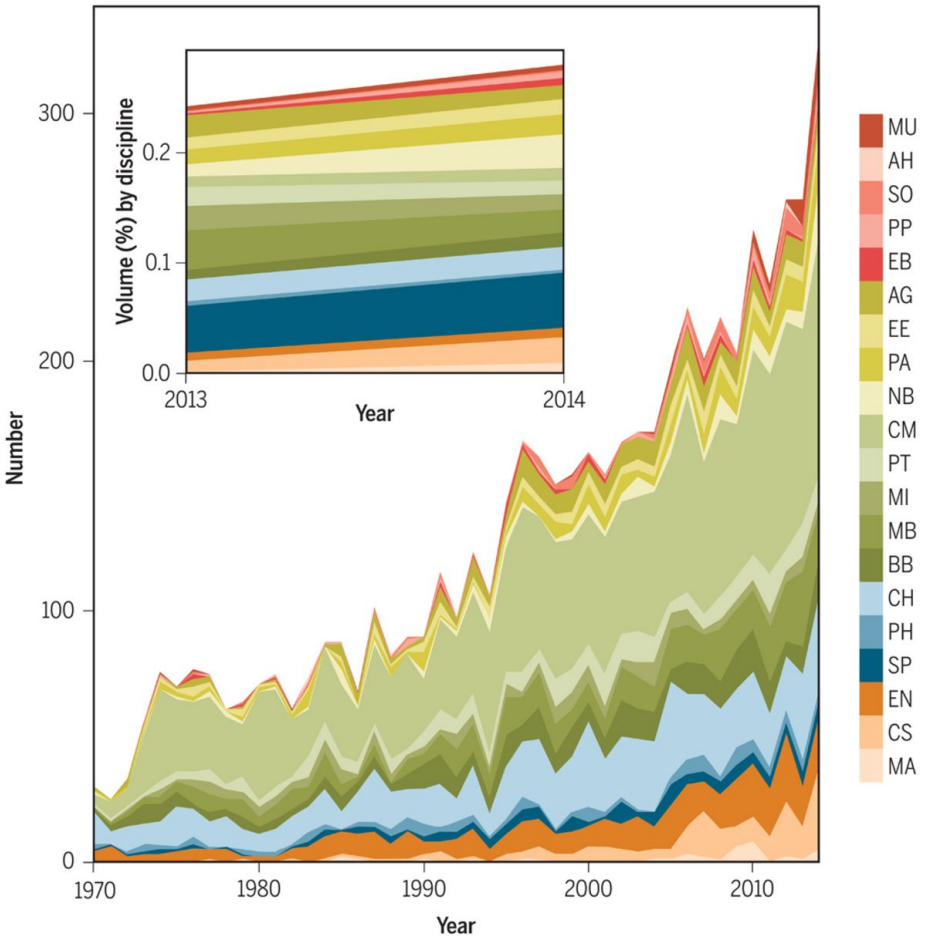
The concept of replicability is older than the reproducibility concept. Even before the technological advances of the last decades, which have allowed the use of computational elements - digital databases, software, and other tools and computational resources - in different areas of scientific research, the replicability was an expectation of scientific findings (ROMERO, 2019). Popper (2005) observed that

> "...non-replicable single occurrences are of no significance to science...".

The term *reproducibility*, on the other hand, suggests different conceptual structures in the literature, in different areas of science, where each area has its own structure regarding how studies and research are conducted (GOODMAN et al., 2016). For example, reproducing an experimental nature of work may involve different efforts and may have different meanings compared to reproducing a study with analytical nature. Different areas still have a different look at the importance of reproducibility and the credibility and validity that this characteristic brings to scientific works. Figure 3.1 displays the presence of reproducibility-related terms in the 44-year publication records of Scopus[1] papers, which mainly suggests a segmentation into four major areas, in increasing order of consideration of reproducibility in research: exact, biomedicine, economics, and humanities. While in some areas, a non-reproducible work may be questioned (HAIBE-KAINS et al., 2020), in others the practice is still just an additional distinguishing feature that has been slowly revealing and maturing in recent decades, as indicated in Figure 3.1, by Goodman et al. (2016).

[1] https://www.scopus.com/

Figure 3.1 - Number of publications registered in Scopus, between the years 1970 and 2014, that had, in the title or abstract, at least one expression indicating that there had been consideration of reproducibility of research or results.



Subject subtitles: MA, mathematics; CS, computer science; EN, engineering; SP, space science; PH, physics; CH, chemistry; BB, biology and biochemistry; MB, molecular biology; MI, microbiology; PT, pharmacology and toxicology; CM, clinical medicine; NB, neurobiology and behavior; PA, plant and animal sciences; EE, environment and ecology; AG, agricultural sciences; EB, economics and business; PP, psychology and psychiatry; SO, social sciences, general; AH, arts and humanities; MU, multidisciplinary.

SOURCE: Goodman et al. (2016).

The particularities of each area of science make it difficult to have a common interpretation of what is expected of a reproducible work by scientists from different disciplines. The absence of a standard began to raise efforts for reproducibility practices to be considered in works from all areas of science, even without specific criteria established. However, the absence of criteria combined with pressures to use prac-

tices that sometimes researchers were not yet ready to face made reproducibility still a fragile practice, not actually present in some works that claimed to be reproducible, or contributing to reveal instabilities in other works (NATURE, 2016).

Citing an example that suggests the complexity involved in the issue of reproducibility, the scientific validity of McKinney et al. (2020b), a paper in the Artificial Intelligence area applied to medical diagnosis, published in the journal Nature, was questioned by a medical community group (HAIBE-KAINS et al., 2020) because of the absence of transparency and details in the documentation of the methods and computational codes used in the paper, making it non-replicable and non-validable. The questioning was answered in McKinney et al. (2020a), in which part of the data, details of the methodology, and some computational codes are revealed and make evident obstacles to reproducibility and replicability such as privacy rules restricting data sharing, as well as ethical values involved. A curiosity about this example is that it reinforces the present levels of reproducibility mentions in computer, medical, and multidisciplinary publications showed in Figure 3.1.

Scenarios of crisis have been reported in both reproducibility and replicability. Research has highlighted failures in attempts to replicate or reproduce scientific papers (BAKER, 2016a; PENG, 2015; HUTSON, 2018; MIYAKAWA, 2020). Those failures can indicate two types of issues: either the work that cannot be reproduced did not present sufficient elements to make a full attempt at reproduction; or the reproduction itself highlighted flaws in the work. Both issues have had efforts to bring improvements and benefits to the scientific context.

3.1.1 Reproducibility failures

Science is based on evidence and progresses with each new finding. No scientific work, however, no matter how rigorous, is free from the possibility of human error; because science, even with advanced technological resources, is still made by humans and, consequently, is susceptible to failures that can undermine the credibility of published results.

Unnoticed errors can occur, for example, in the raw data pre-processing, in calculations, equations, statistical analyses, in computer codes, even in the closed software used, and thus be incorporated in the final results (BOYLAN et al., 2015; KING, 1986; ALLISON et al., 2018; MAKIN; XIVRY, 2019).

Biomedical sciences and chemistry, for example, which deal with clinical trials and

laboratory experiments, deal with transparency as an essential requirement to ensure reliability of published work, to identify, for example, human error when it occurs, and to allow replication, so important in these disciplines to validate methodologies. Scientists and journals demand this transparency through the practice of reproducibility (MCNUTT, 2014; GIBB, 2014; NATURE, 2013).

Failures can also occur at more abstract levels, such as conceptual failures (ALISSON, 2013), false positives (KAPPES et al., 2014), non-conclusive results, failures in theories (SCHLICH, 1993) when there is some kind of bias in experiments or data used in research, or even in the way ideas are conducted through the writing of scientific text and the misinterpretations they can provoke in readers (WOOD; NEZWORSKI, 2005).

Baker (2016a) lists **selective reporting**, **pressure to publishing**, **Raw data not available from original lab**, and **fraud**, among others, as factors that contribute to irreproducible research.

According to Redish et al. (2018), the occurrence of reproducibility failures has an essential role in scientific investigation, which makes it possible to compare conflicting results and try to reconcile them to search for conclusions and possible new scientific findings.

This whole scenario reinforces the importance of transparency efforts through replicability and/or reproducibility, when possible. It is clear that reproducibility is no guarantee of fail-safe, but reproducibility does strengthen the chances that they will be identified.

3.1.2 Reproducibility as a way to join efforts

Just as for Popper (2005) replicability was essential, as mentioned previously, for Cacioppo et al. (2015)

> "...is a minimum necessary condition for a discovery to be credible and informative."

However, different areas of science may have their own protocols to follow to satisfy this requirement.

Sharing analytical workflows is a practice already present in data processing and analysis work in biotechnology, bioinformatics, and biomedicine, and brings many

benefits such as making a ready-made environment available to the community, and providing the execution of one procedure for different input data (LAZARIS et al., 2017; GORGOLEWSKI et al., 2017; BEAULIEU-JONES; GREENE, 2017; GRÜNING et al., 2018; KULKARNI et al., 2018; YANG et al., 2019). However, this practice is still rare in other areas of science.

Cultural changes that have been taking place in the scientific community have made reproducibility strongly motivated and recommended by scientists, scientific journals, and other initiatives in favor of protocols that guide reproducibility in publications (BAKER, 2016b; NATURE, 2016; STODDEN et al., 2016; MUNAFÒ et al., 2017; NATURE, 2018; BERG, 2018). Some works also evaluates the efficiency of those protocols (MULLANE; WILLIAMS, 2017; STODDEN et al., 2018).

It can be noticeable that the topic of reproducibility can become very broad and complex, which can worsen when all areas of science are considered. Because for different areas the issue of reproducibility focuses on different aspects of a scientific work. In some, data reproducibility is more important (PAWLIK et al., 2019; TIERNEY; RAM, 2020), since the raw data used in a paper are not always available, or sometimes they are, but they have undergone some pre-processing that is unclear or has not been disclosed and/or shared; in others, reproducibility is almost entirely solved by making available the data and source codes used in the paper (STODDEN et al., 2018), and so on.

Since the concept of reproducibility that we intended to propose for this work is aimed at the entire process adopted from beginning to end, from this point in this work we will only consider the context of reproducibility, which involves workflows that use computational materials - such as data sets, methods, tools, platforms, codes and other computational resources - and where the main task is focused to data processing and analysis.

The references of reproducible computational workflows we have found, as already mentioned, show that biomedicine, biotechnology, and related areas (BEAULIEU-JONES; GREENE, 2017; GRÜNING et al., 2018; YANG et al., 2019) are the ones that invest the most effort and benefit from this practice. A search for reproducible works in areas such as space geophysics and space weather showed that the practice of reproducibility is still poorly explored in those areas.

3.2 Computational reproducibility

Many independent works produce data processing and analysis workflows focusing on the results that will be obtained at the end and do not consider systematically recording all details about the data, the computational resources, and the procedures adopted to obtain those results, in order to make them available to readers of the published work in the future. In this context, usually, the entire choice of computational material and other information cited takes into account only the personal or workgroup needs and the other local context in which the study was developed. If a change in this culture were in the direction of considering that work may need to be validated or can in some way be reused, the choices involved from the initial stages of the study could be adapted to a new development scenario that would make it reproducible.

Reproducible works resulting from this change in the development culture of scientific works, aligned with the *open science* trend, which aims to increase the transparency and sharing of the materials used in scientific research, can contribute directly or indirectly by increasing the visibility of research, enabling collaborative work, reducing the operational costs of research with the reuse of processes, and increasing the possibility of replicating published research, increasing access to knowledge and the results of scientific research (NOSEK et al., 2015).

The evolution of computing had great participation at the beginning of discussions about the reproducibility culture in publications in several areas. As it turns out, data, code, and computational methods are now part of most scientific work (STODDEN et al., 2016). Many times, the computational methods involved in the works have become, in an imperceptible way, the research protagonists, which before were hidden in the publication, because they were considered just tools, not realizing that sometimes, providing computational details such as code or the raw data preprocessing techniques could explain much more about the work than just the textual description about them.

Computational reproducibility initially had the idea of helping a scientific work to be reproducible turning available its data and computational codes. According to Baker (2016b),

> "If I can run your code on your data, then I can understand what you did."

This view was common among other authors and served as motivation for some journals to adopt data and code sharing policies. (HANSON et al., 2011; ELSEVIER, 2021; NATURE, 2021; SCIENCE, 2011)

The problem was that, even with those transparency policies, not all researchers were prepared to make their research data or codes publicly available. But in many cases, even when data and codes were made available, there were still several obstacles that made the works difficult or unfeasible to reproduce in practice. Some of the obstacles found are listed next (STODDEN et al., 2018):

- Requirement of advanced or inaccessible hardware or computing resources;
- Requirement of long and many files to be preprocessed, with the preprocessing codes or description not mentioned sometimes;
- Requirement of reformatting data files, which could take a long time to be done;
- Requirement of prior technical expertise, for very specific software or techniques;
- Manual coding using some specific programming language;
- Lack of minor codes or partial steps, and
- Requirement of libraries to be installed or other minor adjustments.

The expectation that still is held about works that involve predominantly computational methods and artifacts is that reproducibility is something that is easy to achieve and offer in them, because all they may do is make available their codes and data. But there is a consensus that it is not a simple task to produce a reproducible scientific work only with both materials, once they could involve several details, which are not always naturally considered in an effort over reproducibility (STODDEN et al., 2018; IVIE; THAIN, 2018).

However, as mentioned before, there has been a movement consisting of papers, policies, tools, and other scientific materials that provide guidance on how to conduct reproducible work, with the intention that increasingly more researchers effectively introduce themselves to this new practice. (NOSEK et al., 2015; STODDEN et al., 2016; FUENTES, 2016)

According to Baker (2016a), reproducibility is something learned and practiced until it becomes a natural habit. Some researchers learned the importance of reproducibility, and started to apply it to their own works, only after having a bad experience trying to reproduce someone else's work (BARBA, 2016); others say that reproducibility is a practice that has been acquired through the validation they do of their own work, because they can only accept their results after repetitions and checks that convince themselves that what has been done has validity and can be redone, without having suffered self-biased influences (POPPER, 2005).

A quote from Claerbout (2011) could be a reflection for scientists who have not yet been convinced of the gains that reproducibility can bring:

> "It takes some effort to organize your research to be reproducible. We found that although the effort seems to be directed to helping other people stand up on your shoulders, the principal beneficiary is generally the author himself. This is because time turns each one of us into another person, and by making effort to communicate with strangers, we help ourselves to communicate with our future selves."

We can use Claerbout (2011)'s quote to make an analogy to a computational program that a researcher wrote a while ago and which for some reason wants to be re-run. This problem would bring the first expected requirements over a program or a set of computer codes so that it could be reproduced. When we extend the need of re-execution to another researcher who wants it, other requirements may be raised.

3.2.1 Requirements for computational reproducibility

The obstacles mentioned up to now lead us to introduce some new terms over the computational codes or programs that make more evident the issues that could possibly be encountered when is needed to improve the reproducibility of computational scientific work, in addition to providing the codes and data used in it.

Re-runnability: A program is said to be re-runnable when it can be re-run how many times (at different times) it is needed. To be and remain re-runnable on the original or other computers over time and computational upgrades - software updates, programming language and libraries new versions, etc. - a program should provide enough details about the original running environment so that it can be re-run in an existing or created environment that is as similar as possible to the

original (BENUREAU; ROUGIER, 2018).

Repeatability: After is runnable, a program should achieve the same output that is originally generated. The repeatability is essential to validate not only the program itself, but also the published results it produced (BENUREAU; ROUGIER, 2018).

Reusability: A program is said to be reusable if it has enough documentation/comments that make it be easily run, repeated, or even modified, independently of when or where it is gonna be used and who is gonna reuse it - it can even be the same person who created it (BENUREAU; ROUGIER, 2018).

These three terms were defined over a program or a set of codes, but can also be applied to a computational workflow, in which we have basically a group of codes, with an execution direction established between them. The terms will be defined next can be thought to be applied over simple workflows.

Modularity: A workflow can be called modular when each module (code) provide an independent and well-defined task, bringing gains like avoiding large failures and allowing changes inside each module (code) or a whole module replacement for another module that can execute a similar task, using other methods or techniques, but respecting the predefined input and the output data and their formats (UHRIN et al., 2021; YANG et al., 2019).

Accessibility: This is a term that can be applied to data or to workflow. A workflow is said to be accessible if it can be made available on the cloud or another platform that can be easily accessed and where the workflow can be downloaded from or executed locally, or even remotely, in the online platform itself. At the same time, we can call the data as being accessible too; it happens when the data is easily made available for download or can be used online in the same platform which contains the workflow. But when the execution of a workflow depends on the availability of specific software, operating system or other types of computing resources, accessibility is present when the necessary resources are available in some way: either when they are free software, which can be downloaded or installed for free, or when all the resources necessary for the workflow's execution are being made available in some way attached to it (ASOREY et al., 2015; BEAULIEU-JONES; GREENE, 2017; CASSIDY; ESTIVAL, 2017; AFGAN et al., 2018).

Portability: It has a definition that meets the final part of what was mentioned in the *accessibility* features description. The portability refers to the possibility of a

workflow entirely execution, independently of the platform it is used for it (YANG et al., 2019; GRÜNING et al., 2018).

Other features, like scalability and versioning, are considered for reproducibility but will not be considered for this work, as they do not fit our needs for now.

Resources and tools are becoming increasingly available that make it possible to achieve the features mentioned in this section. The resources considered for use in this work are described next.

3.3 Reproducibility support tools

Understanding of how to make research reproducible is sometimes limited to making the source codes available in a repository. For some works, this single practice may even be sufficient, in cases where the codes are simple to execute and contain the work's main content. In such cases, the person wishing to reproduce that work is considered to be able to obtain the other elements necessary for the reproduction on his or her own. But in the case of research that uses a specific environment (consisting of specifics operating system, programming language, libraries, scripts, environment variables, and other dependencies) for the application execution, just making the codes available is not enough to offer reproducibility for a research project.

Nowadays, several free and open-source tools have emerged to make works available in such a way that they can be reproduced. We mention and briefly describe those that were used in this work in an attempt to make it available and reproducible.

3.3.1 Virtual environment manager

A virtual environment manager (VEM) is a tool that allows the creation of virtual environments in order to customize a computational software environment for a given project, allowing the installation of interpreters such as Python, libraries and scripts in specific versions, without the need of modifying the installed versions for other environments. Conda[2] was the one used in this work, as a part of Anaconda[3] - a popular Python distribution platform.

Conda is an environment management system that can manage packages and their dependencies in a specific environment created especially for a purpose or a project. This feature allows the installation of different sets of packages, in different versions,

[2]Conda's web page: `https://conda.io`, accessed in September 2021.
[3]Anaconda's web page: `https://www.anaconda.com/`, accessed in September 2021.

into different environments, without one interfering with another one.

3.3.2 Code repository

The code repository is a hosting platform that allows source code of projects and applications to be made available and shared in a collaborative and versioned way. Documents and data tied to the projects can be made available in the repository as well. One of the most popular code repositories, and the one that was used in this work, is *GitHub*[4].

3.3.3 Container manager

Container management allows automating the creation, deployment, and scaling of containers. Through specific commands, container management makes it easy to add, replace, and organize multiple containers.

Platforms like Docker[5] allow the management of packages (or container images) with the computing environment necessary for the execution of a given application. The environment, as described in the VEM section, may include code, libraries, and all system dependencies. The images become containers at runtime, when a sequence of commands can be executed to prepare the environment for use. The container images can be made available in repositories - like Docker Hub[6] - from where they can be downloaded, instantiated, and executed in any machine that also has the Docker installed. A container can be manipulated in a way that simplifies the configuration, execution, deployment, and management of an application in complex environments.

3.3.4 Computational notebook

A computational notebook (or just notebook) is a browser-based platform that, aggregated with scientific work, provides in a single IPYNB (IPython notebook) extension file a documented code split into interactive blocks, that can be executed individually. It includes the elements necessary for the development of a program: interpretation and execution of a computer code, its documentation, and the display of its results.

Its interactivity and web layout allow each step of a code to be more easily understood, its parameters to be easily changed and tested, and its results to be displayed

[4] *GitHub*'s web page: `https://github.com/`, accessed in September 2021.
[5] Docker's web page: `https://www.docker.com/`, accessed in September 2021.
[6] Docker Hub's web page: `https://hub.docker.com/`

in a raw form or through graphics, using the resources of the programming language being adopted. Jupyter[7] is the most popular notebook, generally used by users of Python and R languages.

3.3.5 *ReproZip*

This is a software that allows the packaging of an application in a self-contained package along with all of its data files, libraries, environment variables, and other resources used to run the application, making the work completely reproducible. Can be used in different types of applications, including different purposes tools - like for data analysis, scripts, notebooks, and graphics - in different programming languages.

ReproZip, after being installed in the environment in which the application will run, tracks the system calls used by the application during its execution to automatically identify which files should be included in the package. This process creates a metadata file listing all the calls and dependencies that were needed during the application run. Such files can be reviewed and edited before generating the final package file (CHIRIGATI et al., 2016).

By unpacking this package on a different machine, *ReproZip*[8] is able to automatically set up an environment just like the original so that the application can be run, without having to locate and install dependencies, or even having to run the same operating system.

The only condition imposed by the software is that the packaging must be done on a Linux operating system, but unpacking can be done on any machine, with any operating system, that has Docker installed or any of the many compatible unpackers plug-ins, or also on *ReproZip*'s cloud server, *ReproServer*[9] (CHIRIGATI et al., 2016).

[7] Jupyter's web page: `https://jupyter.org/`, accessed in September 2021.
[8] *ReproZip*'s web page: `https://www.reprozip.org/`, accessed in September 2021.
[9] *ReproServer*'s web page: `https://server.reprozip.org/`, accessed in September 2021.

4 NEW WORKFLOW DEVELOPMENT

This chapter has the general objective of describing the stages of the workflow that performs the prediction of solar irradiance. It is a set of tasks applied sequentially over the initial data, with the intention of extracting information and condensing it so that it can be used as input to a recurrent neural network to perform the irradiance prediction.

In this work, most of the original computational code was translated into another programming language, leaving aside only the visualization part of the results, which was done independently. Details of the translation steps and the experiences during and after its realization are reported.

At the end of the chapter, the structure and characteristics of the translated workflow will be presented, with the algorithms being almost loyal to the original workflow, after applying some concepts proposed to make it reproducible. The gains expected after the changes are also related.

4.1 Original workflow

The original workflow was developed by Vieira et al. (2011b) to provide the task of predicting short-term total and spectral solar irradiance - from six hours to three days - in near real-time, using images of the solar photosphere - continuous images and magnetograms (described in 2 as initial data. Briefly, the workflow downloads the images, which are processed so that their features are extracted and condensed to serve as input to a recurrent neural network, which performs the irradiance prediction.

It had been set to run continuously on the EMBRACE web page as a forecasting service, but, for different reasons, the service was discontinued. One of the main obstacles to the continuation of the service was the lack of human resources to continue with the management of this service on the web page.

The workflow's processing steps consist of:

1 **Downloading the magnetograms and continuum images for all the time window chosen and in the defined resolution;**

2 **For each time instant:**

2.1 Identifying and delimiting the solar disk region (Figure 4.3(a)) and discarding information outside this region (such as textual elements of image identification).

2.2 Creating a mask to divide the solar disk into eleven rings (circumferences) from the disk's center to its edge, as shown in Figure 4.3(b).

2.3 Processing the magnetogram:

2.3.1 Converting the image from RGB[1] to gray-scale[2]. (Figures 4.4(a) and 4.4(b)).

2.3.2 Identifying connected groups of pixels (active regions) within the solar disk - considering the "cross" neighborhood criteria for connectivity (Figure 4.1) - and labeling them numerically. This segmentation is performed with a *Scikit-Image* library function that also: differentiates the image's common background, closes small gaps, removes pixels that touch the image's edge, and also filters out very small objects (WALT et al., 2014). The function returns the image matrix with the pixels belonging to each active region labeled with a different natural number.

2.3.3 Extracting the coordinates, the area, and other properties of each active region identified. This task is performed by another *Scikit-Image* function, called *regionprops*.

2.3.4 Labeling each active region again, but now by area sizes, as shown in Figure 4.5, considering the thresholds informed in the left column of Table 4.2.

2.4 Processing the continuum image:

2.4.1 Transforming the image from RGB (Figure 4.5(a)) to gray-scale (Figure 4.5(b)).

2.4.2 Scaling the image colors to help with the identification of umbra

[1] An RGB image is represented with a matrix with dimension M X N X 3 (in this case, 1024 X 1024 X 3), where each of the 3 vectors corresponds respectively to red, green, and blue intensities of each pixel. Each pixel value varies from 0 to 255.

[2] A gray-scale image is composed of only one matrix M X N (in this case, 1024 X 1024), where each pixel can be represented by a single number, which ranges from 0 to 255. This value determines how dark the pixel appears (e.g., 0 is black, while 255 is bright white).

and penumbra regions.

2.4.3 Identifying dark regions, through each pixel color, and labeling them as penumbra (lighter shade) and umbra (darker shade), considering the thresholds informed in the right columns of Table 4.2.

2.5 Combining the labeled matrices of both the magnetogram and the continuous image into a single matrix, obtaining six categories of structures: four sizes of active regions, penumbras, and umbras. An example is presented in Figure 4.7.

2.6 For each of the six labels present in the new matrix:

2.6.1 Calculating the number of pixels referring to that label.

2.6.2 Computing the label's relative frequency, based on the total amount of pixels in the solar disk.

2.6.3 Computing the label's relative frequency, based on the total number of pixels in the image, in each of the regions formed by the eleven rings on the disk.

2.7 Saving the two resulting arrays: the first one with length 6: ratio of the total area for each of the six classes and the total area of the solar disk; and the second one with dimension $6x11$, formed by the ratio that is similar to the first one, but this time considering the total area of each class, within each of the eleven rings. The second matrix is the one chosen in the original workflow to be presented as input to the RNN.

3 Processing the previous steps for all images in the time window selected for training, validation, and testing of recurrent nets. Images that will only be used for activation should also be processed with the same steps.

4 Concatenating the matrices resulted from each image so far (of size $Nx6x11$, where N is the number of images processed) into a single matrix.

5 Making the choice of the irradiation type that will be used: TSI, or

any emission line (SSI); and organizing the matrix to serve as inputs for the RNNs. The input matrix is then associated with the respective desired irradiance values - which will serve as output for the RNNs - considering the time interval chosen for prediction.

5.1 Checking whether the resulted matrices values have outliers and eliminating them, if any, through one-dimensional interpolation.

5.2 Reading the irradiance data file and selecting the data according to the chosen irradiance type.

5.3 Choosing the labels and rings that will be considered for use as input to the RNNs. In this step, the first two classes of regions - small and medium1 - and the last outer ring of the disk are discarded, since they are considered to bring irrelevant information to the prediction. This resulted in an $Nx40$ matrix (four classes into ten rings), where N is the number of images processed.

5.4 Performing small interpolations, in case of missing data.

5.5 Preparing the input matrices, considering the temporal shifting between the inputs and the output, as a function of the chosen prediction interval.

6 Preparing the net's hyper-parameters, if training is to be performed, or loading the already saved net with its settings and training weights, so that it can perform the prediction.

7 Training, validating, testing, or activating, depending on the chosen action, and displaying and saving the results.

7.1 Displaying training and validation results, if applicable (training and validation errors)

7.2 Saving the net's settings and weights, resulted from the training phase.

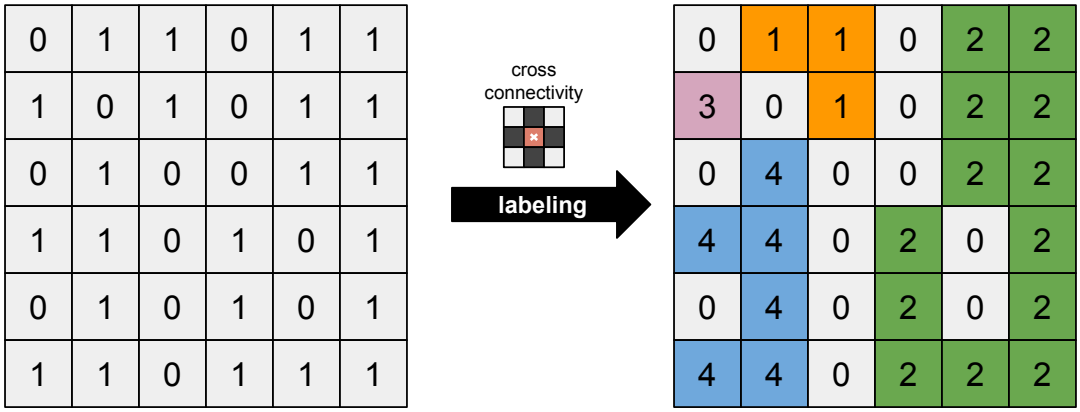
7.3 Displaying results of the test and activation phase, if applicable.

Table 4.1 - Workflow's steps and their corresponding codes.

CODES	STEPS
getTableDataScript.m	1
calc_mu_hmi.m	2.1 and 2.2
figure_mask.m	2.3.1, 2.3.2, 2.3.3, 2.3.4, 2.5 and 3
compute_sunspot.m	2.4.1, 2.4.2 and 2.4.3
gera_figura_areas.m	2.6.1, 2.6.2 and 2.6.3, 2.7, 3 and 4
model_mdi.m or model_eve.m	5, 5.3, 5.4, 5.5, 6, 7 7.1, 7.2 and 7.3
check_areas.m	5.1
read_tim_tsi.m or read_eve.m	5.2

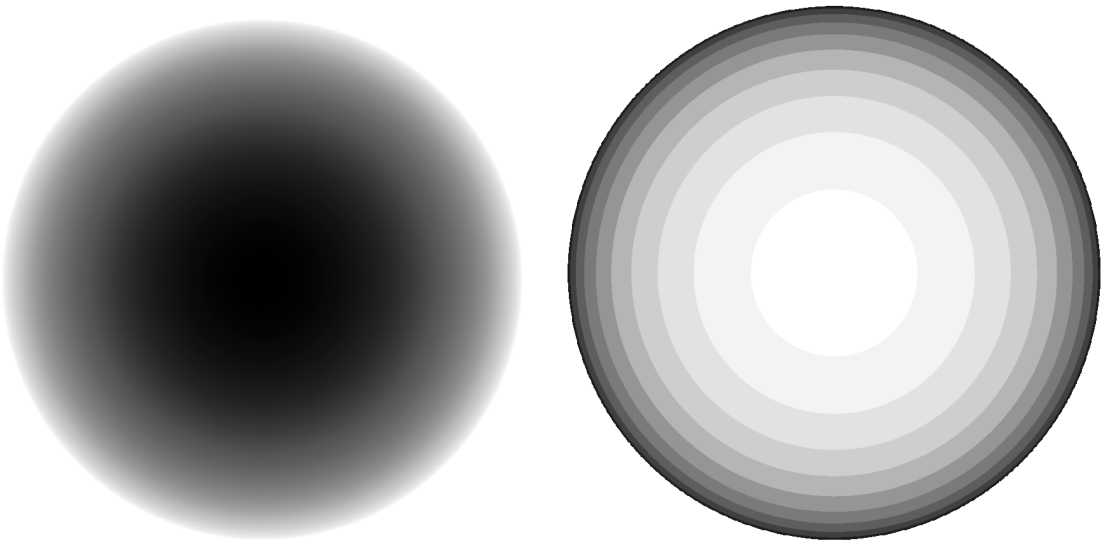
Up next, Figures 4.1, 4.2, 4.3, 4.4, 4.5, 4.6 and 4.7, mentioned in the description of the original workflow steps, are presented. And a brief sequential representation of the workflow steps can be seen in Figure 4.8. The figures refer to the time instant of 00:00h on 10/20/2014 and were plotted in *Spyder*[3] IDE (integrated development environment), for *Python* language.

Figure 4.1 - Among the eight neighboring pixels of a given central pixel, connectivity would be the connection of this central pixel to others which have the same color. The "cross" connectivity limits the neighborhood only to four adjacent pixels: the right above one, the right below one, the just left one and the just right one.



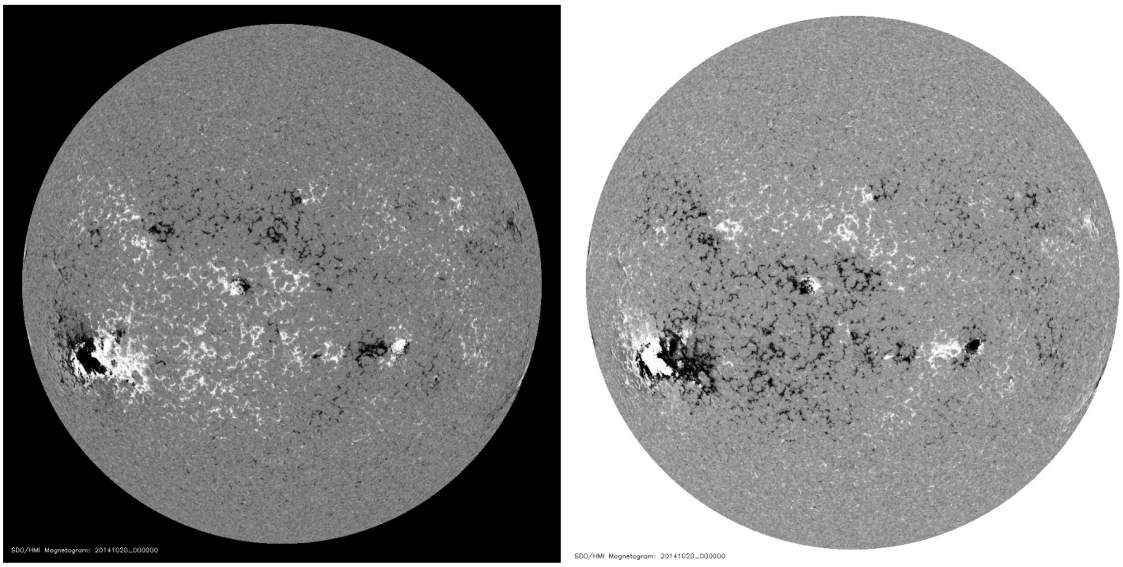
[3] *Spyder* webpage: https://www.spyder-ide.org/, accessed on July 19th 2021

Figure 4.2 - On the left, the solar disk area identification and limitation; and, on the right, the disk partitioned into rings - from the core to the edge of the circumference.



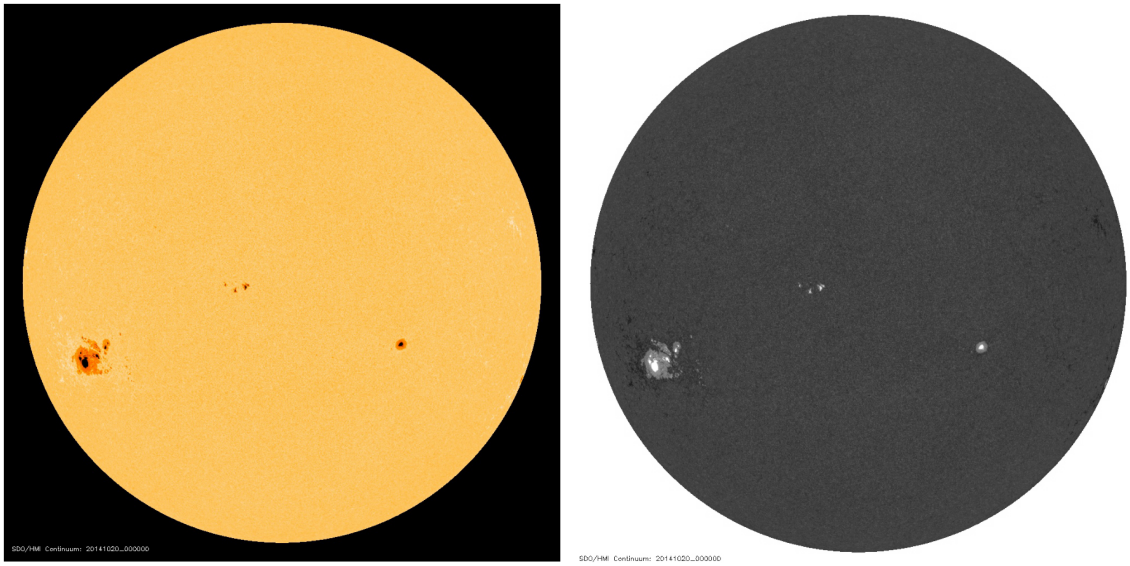
(a) Mu (b) MuRings

Figure 4.3 - On the left, the magnetogram in the downloaded version, in RGB; and, on the right, the same image after being transformed into gray-scale.



(a) Mag (b) MagGray

Figure 4.4 - On the left, the continuum image in the downloaded version, in RGB; and, on the right, the same image after being transformed into gray-scale.



(a) Continuum (b) ContinuumGray

Figure 4.5 - Magnetogram with labeled pixels, based on the identified connected subregions' area size.

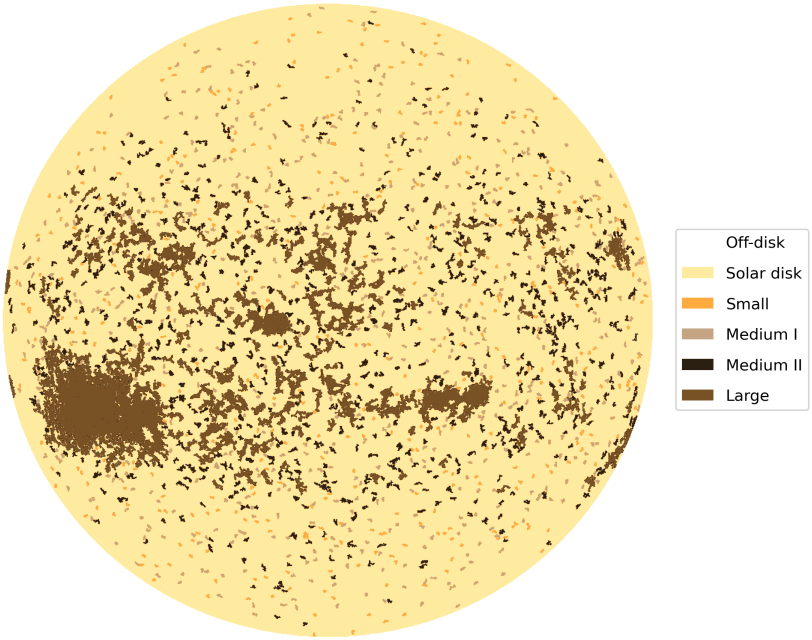


Figure 4.6 - An example of continuum image with dark regions labeled according to the pixels coloration, classified in umbra or penumbra. We observe that this figure was generated exceptionally for this document, because in the execution sequence, the sunspots classification is performed overlapping the active regions classification.

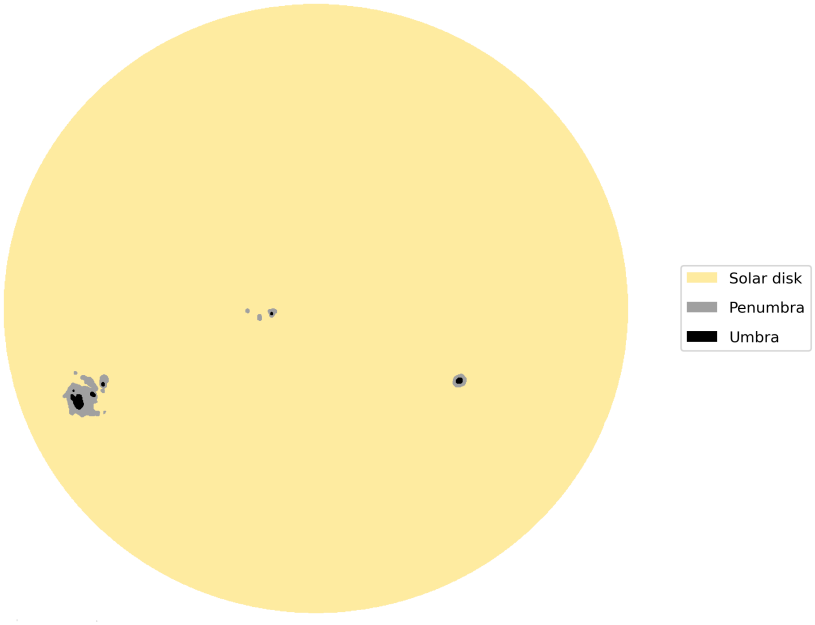


Figure 4.7 - The union of the magnetogram classification with that of the continuous image, with a total of six classes. It can be seen that the umbra and penumbra regions overlap the large active regions.

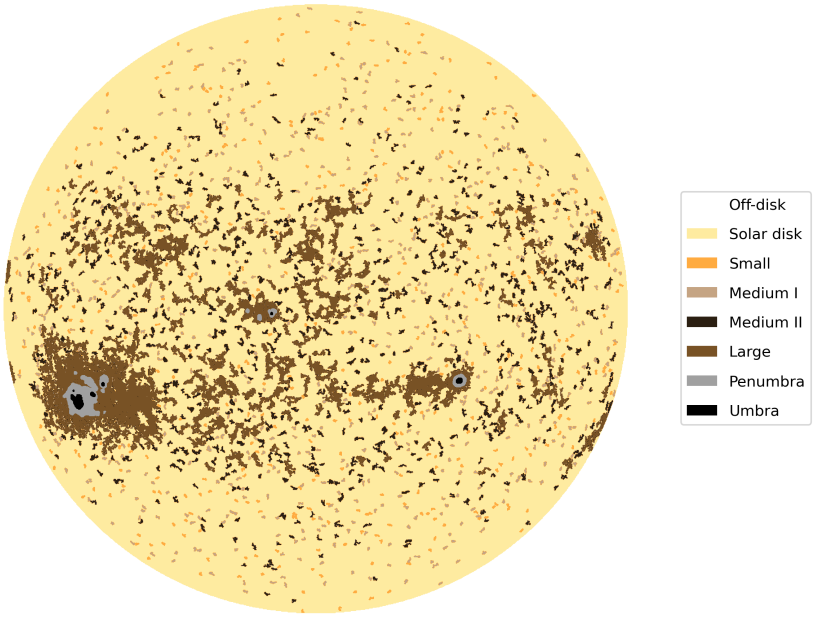


Figure 4.8 - A brief sequence of the workflow steps.

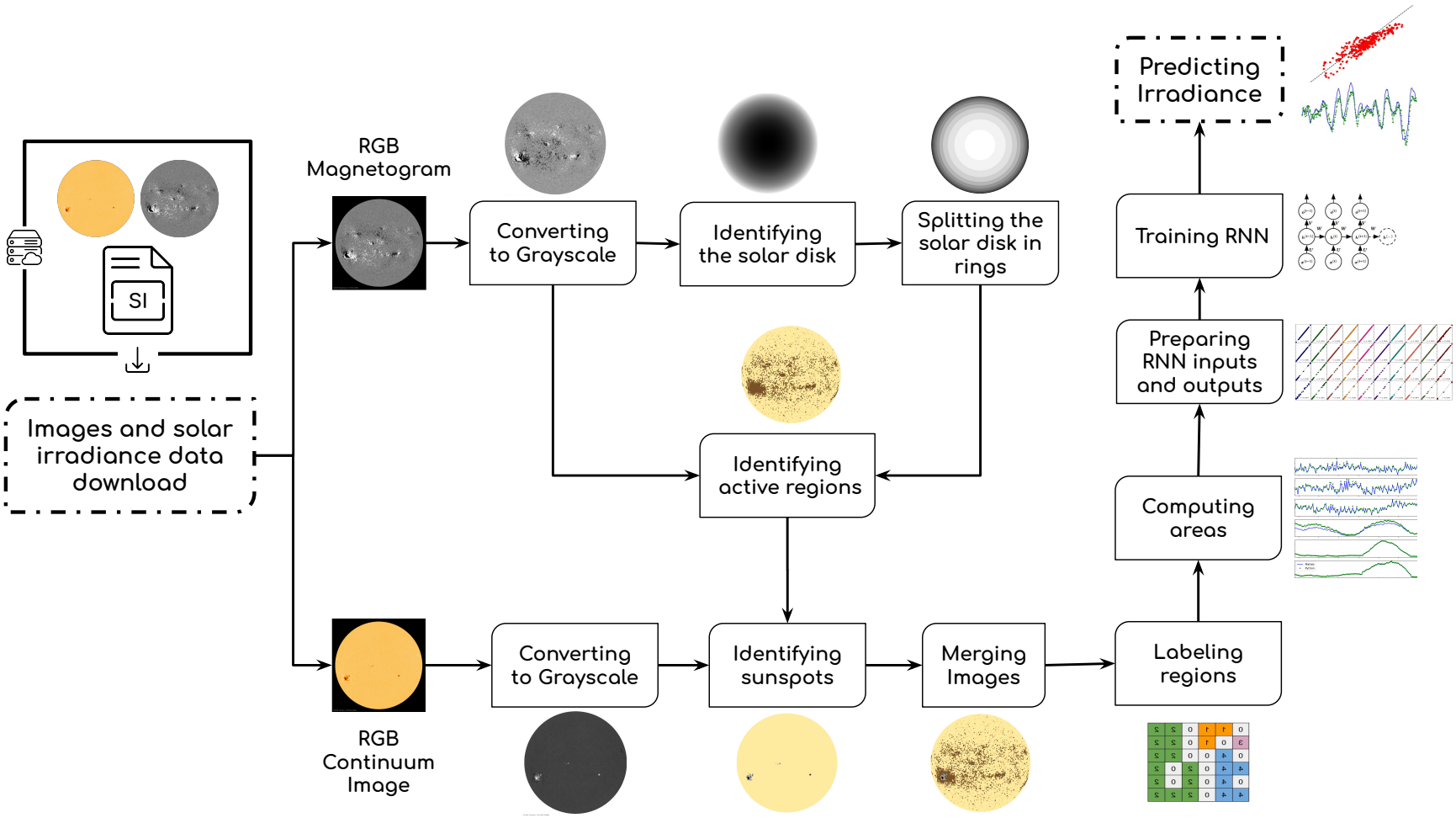


Table 4.2 shows the thresholds used to classify the active regions in magnetograms and the sunspots in continuum images. The left side of the table shows the thresholds used to classify the active regions identified in the magnetograms by their area size. Taking an image of 20/10/2014 as an example, its total solar disk has the size of 712448 pixels. The table's right side shows the thresholds used to classify the umbra and penumbra regions in continuum images. Before the classification, a convolution function is applied over the gray image and the resulting image has its range of pixels standardized. Taking now as example the continuum image for the same date and time taken for the magnetogram, the range obtained before the umbra and penumbra classification was from -26.75 to 2.5 (the darkest pixel to the brightest).

Table 4.2 - Thresholds used to classify the identified active regions by their area size, for which the total solar disk has the size of 712448 pixels for the image of 20/10/2014; and to classify the sunspots regions of umbra and penumbra, for which the pixel values vary from -26.75 to 2.5 (the darkest pixel to the brightest).

Magnetograms	**Label**	**Continuum Images**	**Label**
$area \leq 16.6883$	small	$pixel \leq -4$	penumbra
$16.6883 < area \leq 24.8066$	medium1	$pixel \leq -15$	umbra
$24.8066 > area \leq 89.0673$	medium2		
$area > 89.0673$	large		

4.1.1 The original workflow specifications

The original workflow development was not intended to share or make the codes and process available for other researchers to reuse. For this reason, practices that would make this possibility easier and more feasible were not considered at that time.

We can list as its main specifications:

a) The computational codes implemented for the original version of the workflow steps were written in the *Matlab*[4] proprietary language, with the use of proprietary resources (functions and toolboxes), part of which was specifically available only for the version of software used;

b) The operating system used in the development was Windows[5], a proprietary operating system, which requires a specific software version.

[4] *Matlab* webpage: `http://www.mathworks.com/`, accessed on July 2021.
[5] Windows webpage: `http://www.microsoft.com/windows/`, accessed on July 2021.

c) The codes have only a few lines of comments, which results in a considerable amount of time taken to understand the logic and tasks behind the codes;

d) The process produces partial and final outputs in a specific format (*.m) - that can only be read by *Matlab* itself.

e) The *Matlab* codes concentrate 100% of the workflow procedure, including any kind of pre-or post-processing.

Once the goal of this work is not to evaluate the reproducibility of the original workflow, but to replicate it and make it as reproducible as possible, we will not dive further into its characteristics.

The requirements that are believed to be mainly necessary to enable the computational reproducibility of a scientific experiment were mentioned in Chapter 3 and served as a guide during the development of a new version of the solar irradiance prediction workflow.

Based on the original workflow specifications, it was primarily believed that the computing resources such as the programming language and the operating system, when free, increase the chances of reproduction. Because of that, the new workflow was developed in a free programming language - *Python* - and to run in a free operational system - *Linux*. All the auxiliary software and resources used were also of free use and will be mentioned in this chapter.

It was also believed that the partial and final files generated by the procedure should also have a free, standardized format that could be easily handled by other languages and platforms. The CSV (comma separated values) is a universally used file format that makes the communication between different software and languages easy, which is why it was chosen to be used for partial and final results storage.

Other adaptations carried to bring reproducibility to the workflow's new version will be also mentioned in the next sections.

4.2 The new workflow development: replicating the original one

The new workflow development, with the objective of repeating the entire process of the original workflow and obtaining the same final results by using the same input data, can be considered an attempt at replicability, once the computational tools

used for that are not the same used in the original work. In order to report on the replication process, the steps involved in this process will be described below, as well as the observations made, and the obstacles encountered.

4.2.1 Translation: steps, validation, and obstacles

A substantial part of the effort to build this new version focused on translating the original procedure into the *Python* language. The translation itself required preparatory steps before the writing of the codes could begin. These pre-steps and the steps that involved the translation will briefly be listed next. It is worth noting that the theoretical study of the solar irradiance context was necessary throughout the development of this work, as well as learning the *Python* programming language - steps that can influence a general process of reproducibility.

Step 1. Study of the Sun-Earth relationship, which enabled the basic understanding of the general physical processes involved in this relationship, and in which context this work's theme fits on;

Step 2. A specific theoretical background for the comprehension of the procedure adopted in the workflow;

Step 3. Understanding each macro step of the workflow;

Step 4. Search for a *Matlab* license to be able to run the original codes and check partial and final results;

Step 5. Installation of *Matlab* and the adaptation of the computational environment - directory structure, paths, etc. - that would allow re-execution of the complete procedure, checking the availability of packages and tools for the version obtained;

Step 6. Establish a data period for the first execution of the procedure and download those data needed (images, TSI, and SSI measurements) to run it in the installed software;

Step 7. Understanding the purpose of each source code, its internal logic, and its input(s) and output(s);

Step 8. Study of the *Python* language, starting from the basic level, to understand concepts such as the present data types, their structures, and differences, their methods, the language syntax, etc.;

Step 9. Start of code translation;

Step 10. Understanding the *Matlab* closed functions used in the codes and searching for *Python* packages that had equivalent functions that could replace the original ones;

Step 11. Testing to validate the substitute functions;

Step 12. General testing for comparing the results of the original procedure and the translated one with respect to TSI;

Step 13. Finding a compatible recurrent neural network to perform the prediction;

Step 14. Tuning the network hyper-parameters and conducting experiments to look for similar performance for TSI prediction;

Step 15. Execution of tests for validating results and of experiments, to obtain network performance for some SSI emission lines.

Each step presented distinct characteristics regarding duration, complexity, and expected return. For the purpose of analysis from the point of view of reproducing a scientific workflow, some of those particularly chosen steps will be reported in more details in order to have a better understanding of the issues involved in translating a set of codes from one programming language to another and when the first one is proprietary, as in this case.

The study and understanding of the application area (Step 1.), the space weather, was a step that, despite having been concentrated more in the pre-translation phase, demanded constant retaking due to the emergence of questions and the search for answers, especially for a better understanding of the workflow business rules. Linked to this need, the Step 2. and Step 3. also required constant restarts and a longer time than expected, so that the understanding of all the workflow steps was sufficient enough to enable the understanding of the context and the validation of the partial results obtained.

Step 4. and Step 5., related to the search for an available *Matlab* license in the institution, brought some obstacles. To be able to re-execute the workflow, a different machine from the one that was originally used to develop it had to be prepared for this. However, the *Matlab* license originally used was no longer accessible and available to be installed again, which required a new license to be found. The only option at the time was a license that INPE had available, which was installed with

the expectation that no compatibility problems would be encountered during the workflow execution since it was a newer version than the one used to develop the original workflow. The installation followed the security norms and protocols of the institute's IT sector, which gave the right to access and use the program only through the internal computer network, not allowing access from any other remote location outside the institute, which in part limited its use. After installing the software, and during the attempt to start re-executing the codes, it was verified that the version in question did not include some packages and toolboxes necessary for re-execution, and its acquisition was not financially viable at that moment. This situation required another license to be searched for. Finally, after weeks, a new license of the software was found and installed, so that the re-execution of the workflow could be resumed and, this time could be concluded to the end.

Regarding the learning of the new programming language, mentioned in Step 8., it was a step without major difficulties and that was worthwhile, because *Python* is a language increasingly popular in the scientific environment, as confirmed by observing the languages used in scientific papers in the astronomy area, and noting its frequent presence in themes and lectures in events in the area, during the participation in some of them. At the same time, this task brought the challenge of understanding the language and its characteristics, while simultaneously using it to encode and translate, without having dedicated a specific time for learning it. What made learning *Python* significantly easier was the fact that it is a high-level language, easy to learn, and very similar to *Matlab* in some characteristics, for example, dynamic typing, being imperative and functional. On the other hand, some points, such as performing matrix operations, took a little more effort, as they required a greater maturity in understanding the available data types and their methods, as well as which were the most appropriate for each translation need of the original variables. The general learning of the language had the great advantage of having a community of *Python* users, noticeably larger today than the community of *Matlab* users, who participate in web pages where you can find answers to almost any kind of question.

A concern that was part of the whole translation process was related to the search and replacement of *Matlab*'s closed functions by equivalent functions implemented in *Python*, mentioned in Step 10.. The first challenge was the fact that the functions in *Matlab* are not open source, which did not allow, at any time, comparisons between algorithms used in them and those used in *Python* functions, which are open source. Therefore, equivalence was verified, for the most part, by considering the function

names, their descriptions in the documentation of both languages, and in comparing the results that the functions offered in the output.

The comparison of the values of the variables and of the partial results to analyze the equivalence between the functions (Step 11. was done through plots or basic statistical measures, such as mean, minimum, maximum, and average. It is worth pointing out that at the same time that some of the functions, already evaluated as being equivalent, presented equal output values in both languages, others presented similar values with small distances in precision. This step was essential for the progress of the translation work, to provide confidence in the process being carried out, and to reinforce the correct understanding of the business rules that were intended to be replicated in the new workflow. Some of the comparisons can be seen in Table 4.3 and in Table 4.4, which list some variables of the code used to identify the solar disk circumference. Another comparison can be seen in Figure 4.9, which compares the sum of the pixels of the different sizes of active regions, and the umbra and penumbra regions of the sunspots, identified in 240 images of October 2014.

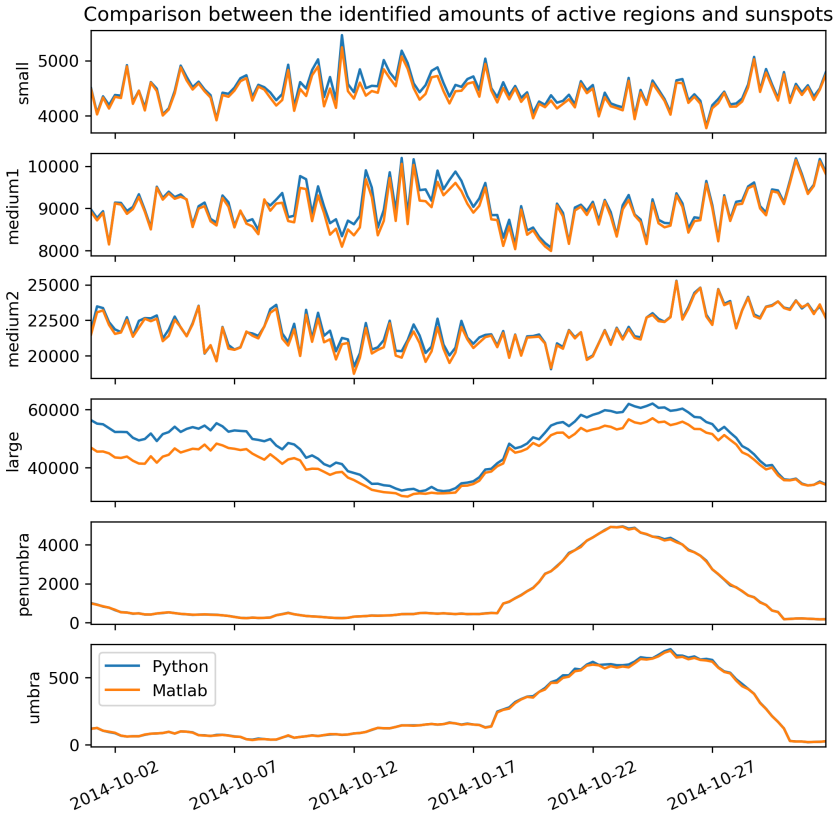
Table 4.3 - Comparison of some variables *Matlab* and *Python* values.

Variables	centerX	centerY	mRadius	areaDisk	timeStamp
MATLAB value	513.32	512.28	476.21	712448	735892
Python value	511.28	512.32	476.21	712448	735526

Table 4.4 - Basic statistic measures for comparing variables.

	Variables	mu	mu_rings	area
Matlab	**min**	0	1.0	140.36
	max	1	11.0	25170
	avg	0.4530	4.31	66.02
Python	**min**	0	1	140.36
	max	0.9999	11	25170
	avg	0.4529	4.81	66.04

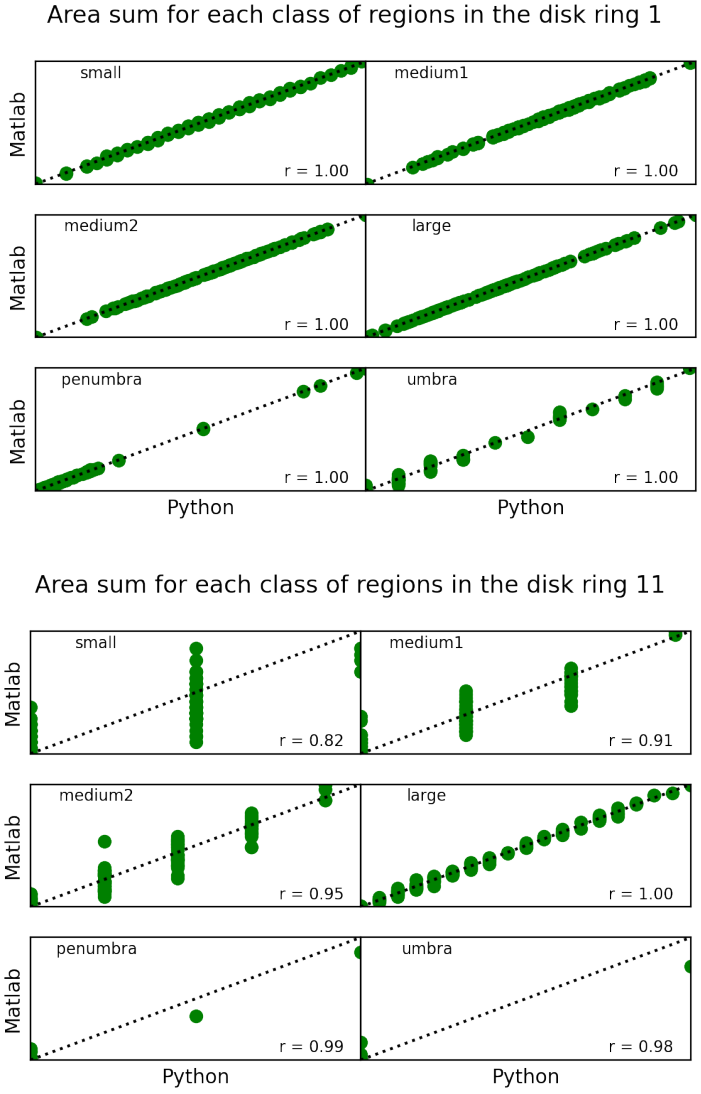
Figure 4.9 - Examples of *Matlab* and *Python* outputs comparison: identification of pixels amounts for each size of active region, and for each region of sunspots (umbra and penumbra), on October 2014.



The *Python* workflow's almost final results, obtained just before preparing them to serve as input for the RNN, were also compared to those obtained in *Matlab*. The Step 12., referring to this task, reported the accumulation of small precision deviations between variables throughout the procedure so far. Such deviation does not appear in most of the matrices generated, but is clearly present in a few of them, referring mainly to the solar disk outermost rings and the presence of umbra in that region, as can be seen in Figure 4.10. The figure shows a good correlation between *Matlab* and *Python* results for the innermost ring (top plot), and a clear deviation when comparing the outermost rings (bottom plot). Even so, the good correlation between most of the matrix columns, seen in the Figure 4.11 plots, leads to the conclusion that the translation was satisfactory. Figure 4.12 correlates the sum of the same areas, but without taking into account the rings split; the figure also shows the maximum correlation level between *Matlab* and *Python* outputs. The columns which gave more discrepancies were already excluded from the original workflow and were not considered as input for the prediction step, as described in workflow

Step 5.3. On the new workflow, they were also discarded.

Figure 4.10 - Correlation between *Matlab* and *Python* output matrices. They represent the sum of the areas of each class of regions on both solar disk rings: the innermost and the outermost, for the time period of October 2014. The correlation regarding the other nine rings can be found in APPENDIX A - A.1.



The pre-prediction results, i.e. the RNN input and output (reduced to an $Nx40$ matrix, as described on the workflow step 5.3) were also compared to validate the values. Figures 4.13 and 4.14 show the matrix comparison.

Figure 4.11 - Linear correlation coefficient between *Matlab* and *Python* output matrices of the sum of areas on each ring of the solar disk. Time period of October 2014.

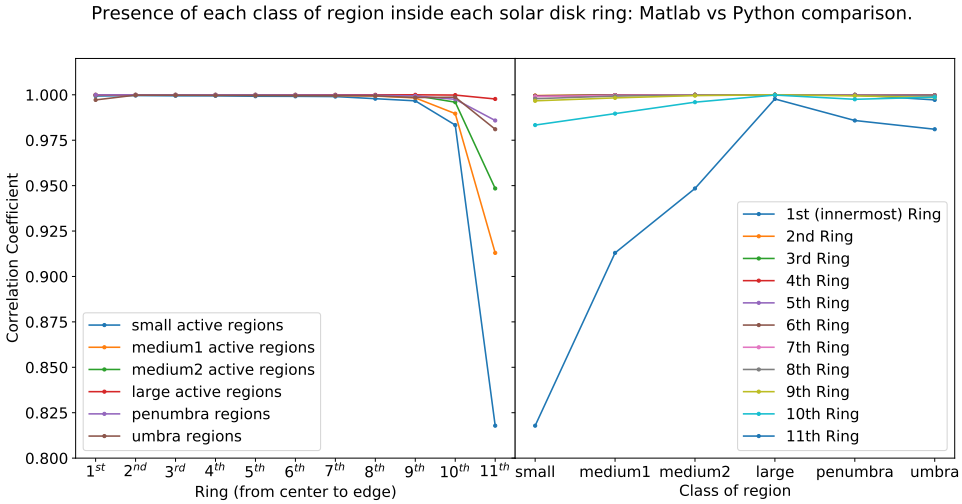


Figure 4.12 - Linear correlation coefficient between *Matlab* and *Python* output matrices of the sum of areas without taking into account the rings split. Time period of October 2014.

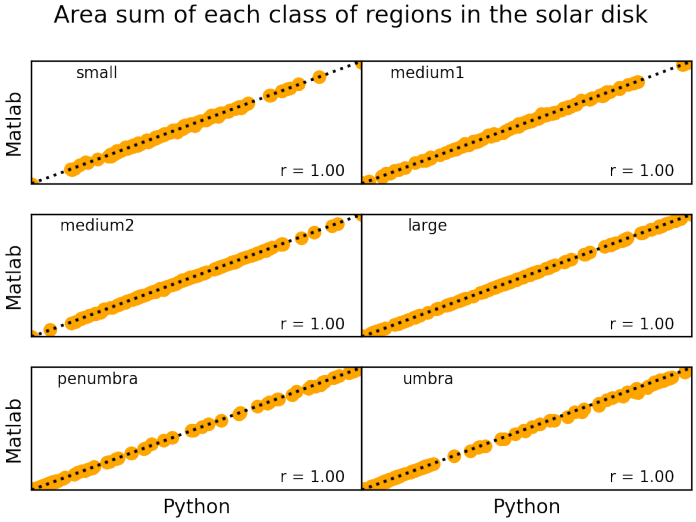


Figure 4.13 - Comparison between RNN input features of *Matlab* and *Python*. Time period of October 2014.

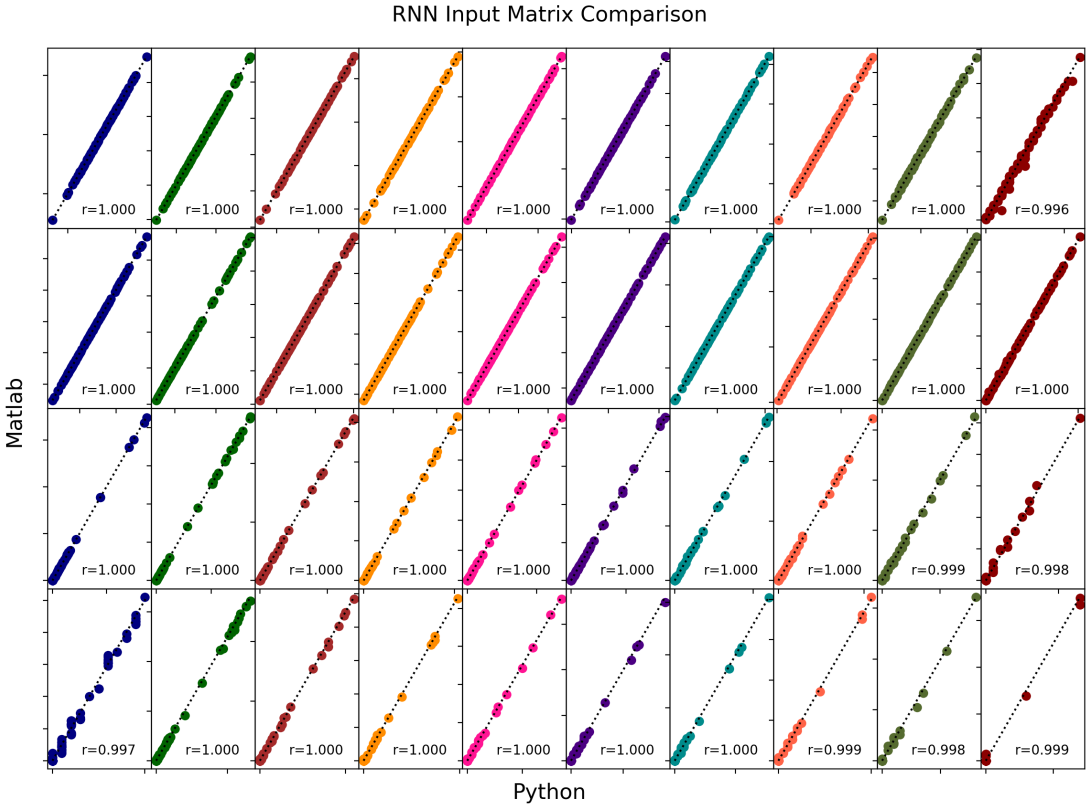
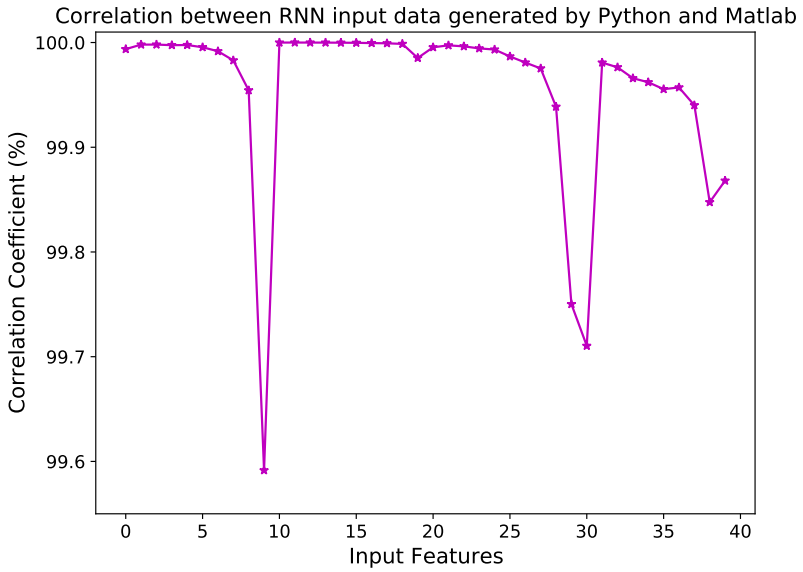


Figure 4.14 - Linear correlation coefficient between each RNN input feature generated by *Matlab* and *Python*. We can observe that for all 40 features the correlation is high. And even the observed outliers still show high correlation (greater than 99.5%). The data time period is the full month of October 2014.



Chapter 6 focuses on the results obtained by Step 13., Step 14. and Step 15., in which TSI and SSI prediction experiments are performed, exploring different RNN architectures, and for different prediction intervals.

4.2.2 Additional auxiliary codes

In addition to having rewritten the functionality of the original workflow, other auxiliary codes were developed to assist in the preparation of data, and also to generate graphs with the results of the experiments. These codes were aimed at some tasks such as to:

- Optimize the data download;
- Cover the data gaps by replacing them with higher resolution images, for time instants close to the missing ones;
- Check data continuity;
- Read the reconstructed data series and generate series with them for the same periods of data used for testing;
- Compile the results of the experiments.

4.3 Making the new workflow reproducible

Among the reproducible characteristics added to the new workflow, we believe that the programming language in which it was developed can be considered as the main one, which will allow the workflow to be reproduced using free platforms. The migration to a free language brought along, consequently, facilities in making the workflow available. This will enable the workflow to continue to be used and improved after the conclusion of this work, both by the group that developed it and by other researchers and students who may be interested in reproducing or reusing it, without concerns about the acquisition of paid licenses.

To achieve this goal, therefore, it was decided to use free tools, starting with the operating system where it will run. It would be ideal if the application could run on any operating system, but mainly on *Linux*, which is free.

4.3.1 Computational elements

Some resources were used to make available the workflow and all the computational elements needed for its execution. Before listing the resources, let us list the elements

that constitute the workflow:

- *Python* codes: The workflow's computational codes are split into two groups:
  - PY files: The files necessary for processing the images and generating the inputs and desired outputs that will be presented to the neural networks are in extension .PY, thus they can be executed directly, without interaction. But they will also be made available in .IPYNB (*iPython* notebooks) to be viewed and executed interactively by using *Jupyter*.
  - IPYNB files: The codes for setting up, training, and activating the neural networks were written in IPYNB to run on *Jupyter* notebooks.
- Data:
  - Workflow's input images (in JPEG format), being one pair of images for each time instant, were separated into sets by time periods; some sets were used for training and others for testing the networks.
  - Irradiance files in text format (TXT or CSV), one file for each irradiance type. The files have a certain period of data that must be observed before their use. The spectral irradiance files contain measurements for the period from 2011 to 2014. The TSI file contains measurements for the period from 2003 to 2019. Other data periods can be freely downloaded from the appropriate web repositories, already mentioned in this document.
  - There are also partial files generated at different stages of the workflow, which are available in CSV format but can also be re-generated by running the application.
- Anaconda was used to create and manage a virtual environment, with:
  - *Python* interpreter in version 3.6.
  - All the libraries used for *Python*, in their appropriate versions.
  - *Jupyter*, for creating, loading, and running notebooks.
  - *Spyder*, as an IDE, for managing the *Python* codes.

4.3.1.1 Virtual environment *rworkflow*

Before starting the code translation process, a virtual environment (VE) was created (through *Conda* for *Windows*), which we named *rworkflow*, specific for the development of the new workflow. The *Windows* operating system was used because of the need of executing the original workflow codes in *Matlab*, simultaneously with the translation, and *Matlab*'s version found was for *Windows*. The VE was initiated by choosing *Python* in version 3.6, and gradually, when necessary, libraries in the newest versions were also added to the environment. The specific VE creation allowed the same versions of *Python* and libraries to be taken as reference for that environment reproduction in other computing platforms after translation.

4.3.2 Using reproducible support tools

With the support of the resources presented in Chapter 3, an effort was made to make this work reproducible.

As a first step, the *rworkflow* environment was recreated in *Linux*, also with the support of *Conda*, through installing each package in the proper version. Next, the application was tested on the new operating system to check if any required system dependencies were missing. This process was a bit laborious, since the VE dependency list, generated through *Conda* on Windows, listed several dependencies linked to the operating system. Filtering out the true dependencies, and searching for the same library versions, were a time-consuming process and that did not guarantee that the application would run completely. After attempts to find the essential group of dependencies, the adaptation was finally accomplished.

With the application running properly, the VE created for *Linux* provided the basis for further actions.

Although it was a necessary step, recreating the VE on *Linux* brought inconveniences that motivated its encapsulation, in order to make it available for reuse, without requiring efforts for re-installing all dependencies. Two solutions meet this need: the creation of a container image using *Docker*, and a packaged file generated by *ReproZip*.

4.3.2.1 *Github* project

On *Github*, all the computational elements used in the development of the workflow were made available: the main codes, the notebooks, the data sets for training, vali-

dation, and testing, as well as auxiliary codes and notebooks for preparing the data and generating the graphs of results. The files have been organized with objective directory names in order to make the contents of each clear. A "read me" file has also been provided with guidelines on how to proceed with the run and some further details.

4.3.2.2 *Docker* image

A *Docker* image was created with *Python* and libraries, following the versions used in the work; and made available on the *Docker Hub* so that it can serve as a base environment for anyone who wants to reproduce the work or reuse it totally or partially. The *Docker Hub* path will be made available on *Github*, where the codes and other files can be downloaded for use with the *Docker* image.

4.3.2.3 *ReproZip* package

The *ReproZip* package was created while the workflow was running for the TSI six hours prediction. It encapsulated in a single compressed file (.RPZ): files, source codes, libraries, and all other system dependencies that were required during the execution;

With this, it became possible to reproduce the workflow on any operating system. Note that the *ReproZip* needs to be installed in the *Linux* environment to perform the mapping of an application's dependencies and the pack file generation, but its unpacking and re-execution can be performed under any other operating system that has the *Docker*, or another compatible container manager, installed. The reproduction can also be performed in the cloud, using the *ReproServer*.

Perhaps a disadvantage of *ReproZip* is that in the mapping process, the software only looks at the resources called during the application's execution. If some optional resource was not called in the execution during packing, that resource will not be considered in the re-execution as well.

In this work's workflow case, different processing is performed for the generation of each of the forecasts made. Therefore, considering that TSI predictions were made for six distinct time intervals, and the three irradiance emission lines, for three time intervals each, there would be a total of fifteen different networks for prediction, which would require fifteen different RPZ files for the execution of each type of prediction. If we also consider making available the two gated architectures, there would be 30 *ReproZip* packages to be made available, which would demand

no less than 15GB of space, considering the minimum of 500MB occupied by each compressed file.

Since we don't have that much space to make all 30 RPZ files available, we will limit ourselves for now to making the ReproZip file available only for the TSI forecast six hours ahead.

4.3.2.4 Application versions

The standard structure of the workflow codes receives a period of data to be processed, with a start date and an end date, and performs all image processing and the recurrent network training, for the chosen irradiance type and prediction time interval.

A new version was created specifically for TSI forecasting, with few changes, that allows running data from a single instant of time. This new structure meets the need of irradiance prediction for two scenarios: near real-time prediction, i.e., as the SDO images are made available, the net, with the saved training weights, predicts the TSI. This version was created in principle, only for the forecast six hours ahead of the TSI, which is what was available on the EMBRACE site. The actual TSI, as it became available, could also be compared to that estimated by the net, so that the net's accuracy could be constantly evaluated.

Both versions were made available on *Github*.

4.3.3 Computational arsenal repositories

The following computational arsenal will be made available on *Github*:

- Data in .JPG (images) and .CSV (irradiance) extensions *Python* codes in .PY extension
- Notebooks *Jupyter* in the .IPYNB extension
- *Docker* file in the .YML extension
- Compressed file *Repozip* in .RPZ extension
- "Read Me" file explaining how to execute the workflow

Since all the data used in this work can be freely downloaded from the already

mentioned institutions websites, only part of them will be made available on the repositories, according to the space limit provided for each *Github* account.

The repository on *Github* can be accessed at: `https://github.com/amitamk/siprediction.git`.

The *Docker* image can be accessed on the *Docker Hub*, at: `https://hub.docker.com/r/amitamk/siprediction`.

4.3.4 Characteristics aggregated

After making the workflow available through the previously mentioned resources, it is believed that the workflow has succeeded in adding the following features:

- **Rerunnable:** Since your environment with the software dependencies has been made available and can run on any platform.
- **Repeatable:** After using the created environment, the entire workflow was run on a different machine, which received no initial preparation and received no files other than the provided file package.
- **Reusable:** All the source codes and notebooks have their functions described, and there is a document next to the computational arsenal explaining the order of execution of the workflow and the inputs and outputs of each step (or module).
- **Accessible:** Because all the essential elements for the execution of the workflow have been made available in the cloud so that they can be downloaded from anywhere in the world. In addition, the workflow's execution does not depend on any paid computing resources.
- **Portable:** You could say that the workflow is portable, since it can run on the main operating systems - *Windows*, *Linux*, *macOS* - that could have *Docker* running on.
- **Reproducible:** Since all the previous characteristics could be met, we understand that the new workflow is reproducible.

4.4 Conclusions

The process described in this chapter has shown that the replication of an existing workflow can encounter barriers of various kinds: software, hardware, license,

programming language, incompatibility of functions, etc.; and that the practices involved in the process of making a work reproducible are important so that many of these obstacles can be avoided. It is worth pointing out that, in the case of this migration work, there was contact with its developer when it was necessary to seek a solution to some of the problems; but when the reproducibility of a work cannot count on the support of its developer, the barriers may not permit the work to be reproduced.

Although reproducibility is an important characteristic, this work showed that it is not merely a feature to be added to a work. Every step during development requires this condition to be considered. But it is also believed that this practice can be exercised, be improved through each work, becoming a natural habit.

It is worth mentioning that a reproducible work requires the effort of whoever is going to reproduce it, toward mastering or learning, at least basically, how the adopted resources work.

There is no doubt about the benefits that the practice of reproducibility can bring to science, and so the exercise in offering it in works is a worthwhile effort.

5 RECURRENT NEURAL NETWORKS

This chapter will initially give the theoretical background of the technique used for solar irradiance prediction, the recurrent neural network (RNN), and present its different architectures. It will then list the main variables involved in the training, validation, and testing phases, such as data, quality measures, as well as the pre-training settings and hyper-parameters of the networks.

5.1 Machine Learning

Before discussing Machine Learning (ML), it is worth mentioning a larger and earlier area, that is Artificial Intelligence (AI). A definition for AI is given by McCarthy (2007) as:

> "... the science and engineering of making intelligent machines, especially intelligent computer programs."

For the meaning of "intelligence", it is taken as a basis the characteristics of human intelligence, which are desired for machines to have. Some of artificial intelligence concepts and techniques are modeled on human biological mechanisms, such as the human brain or biological evolution.

However, what makes a machine intelligent, or what is the exact meaning of "artificial" are still questions that are not so consensual among AI researchers. Possibilities and future scenarios for the field are discussed, and some questions about the large field of AI still have no common answers (DAS et al., 2015; MCCARTHY, 2007).

But if on one hand, the meaning of artificial intelligence may have become a philosophical and complex discussion (FETZER, 1990; MCCARTHY, 2007), on the other hand, the term machine learning has a more well defined meaning, which is to exploit the computer's "ability" to learn, through specific algorithms.

Machine learning is certainly the most popular area of artificial intelligence, which has grown independently of the issues behind AI concepts. It has raised interest from areas that have data to be mined, analyzed, and understood; especially in an era when so much data is generated without an adequate information extraction. Moreover, they are part of our daily life, running on different platforms and electronic equipment, performing tasks such as natural language processing, pattern recognition, and predictive analysis, among several others (BONACCORSO, 2017).

Machine Learning algorithms are mathematical models that have been increasingly used both in academia and in business and industrial areas due to their availability, application possibilities, and positive results. They are techniques that help humans in decision making and can bring benefits such as reduced task execution time, detailed analysis of large volumes of data, black-box mapping, among others. They can be separated basically into supervised, unsupervised, and reinforcement learning algorithms.

Supervised Algorithms: As the name implies, these algorithms are those that need supervision, i.e. they need to be trained on data for which the input (or set of inputs) and respective output is already known. During training, the errors between the desired outputs and the ones calculated by the algorithms serve as a basis for the "learning" processes in order to achieve generalization - the scenario in which the algorithm is sufficiently experienced to handle new data. Thus, after training, the testing phase will allow outputs to be calculated for unknown inputs that were not presented in the training phase. Among the most common tasks of supervised algorithms are regression (or prediction), where outputs are usually continuous values, and classification, where outputs are discrete classes or values.

Unsupervised Algorithms: In this type of algorithms, there is no supervisor, no correct answer, or error between what is desired and what is obtained by the algorithm, as is the case in supervised algorithms. The unsupervised algorithms can be applied to labeled or unlabeled data and aim to group them according to the similarity between them, which is usually calculated through distance measures and centroids. Among the tasks performed by these algorithms are segmentation (or clustering) and similarity detection (BONACCORSO, 2017).

Reinforcement Learning: These algorithms are inspired by the idea of reinforcement learning from areas such as psychology, neuroscience, and also computer science, in which an agent learns to behave by interacting with an unknown and dynamic environment. Its learning occurs through trial-error, receiving punishments and rewards, without being necessary to predetermine how a certain task should be performed (KAELBLING et al., 1996). This paradigm basically falls into three types of algorithms: those with the dynamic programming approach, those that use samples from Monte Carlo methods, and those that deal with temporal difference learning (SUTTON; BARTO, 2018).

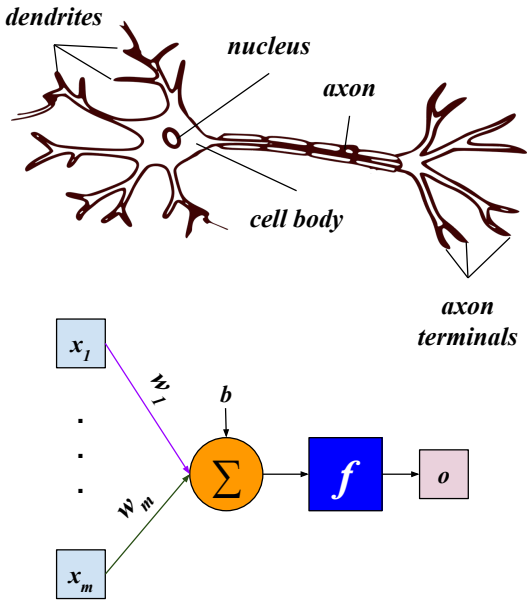
Among the most traditional and popular ML techniques are Artificial Neural Networks (ANNs) with their different architectures.

5.2 ANNs

The origin and evolution of ANNs are based on the computational model of artificial neuron proposed by McCulloch and Pitts (1943), inspired by the neuron model of the biological nervous system. The biological concepts of nerve impulse, associated inputs received from other neurons, the synaptic connections between neurons, and non-linearity, for instance, were brought into the artificial model in the form of each neuron's inputs, its weighted sum, and its activation through the transfer function. The internal processing of a neuron is common to all ANN architectures and is summarized in the weighted sum (with the vector of weights W) of the neuron's inputs (x_1 to x_m), added to a bias b, over which an activation (or transfer) function f is applied, thus generating the output o of that neuron, described in Equation 5.1, and shown in the representation in Figure 5.1, which compares this "artificial" structure to the biological neuron's basic structure.

$$o = f(Wx + b) \tag{5.1}$$

Figure 5.1 - Artificial neuron internal process (bottom) compared to a biological neuron basically described structure (top).



Gradually ANNs evolved to different architectures, with the three types of algorithms previously mentioned (supervised, unsupervised, or reinforcement learning)

becoming techniques that are widely used in several areas to solve different classes of problems with different approaches. (ABRAHAM, 2005; WALCZAK, 2019)

In this work, we want to perform the prediction task, which can also be referred to as regression, and the supervised ANNs are a better fit for this task. Among the supervised neural networks, the most widely used is the multilayer perceptrons (MLP) network, one of the first architectures to become popular, combined with the *backpropagation* training algorithm (GARDNER; DORLING, 1998; ROJAS, 1996).

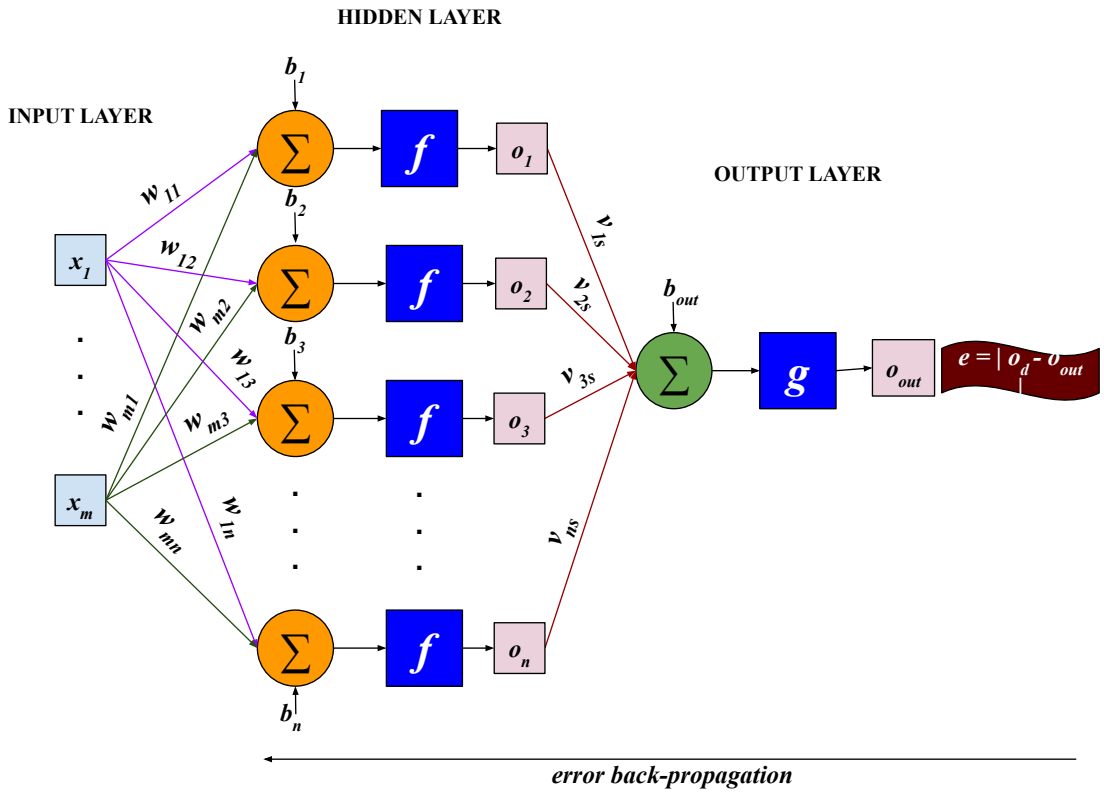
MLP is a simple interconnected neurons ANN model, which has:

- an input layer - which is a non-processing layer, where the inputs are presented
- one or more hidden layers
- an output layer

MLP works on mapping non-linearity between input and output data. It is the superposition of simple but nonlinear transfer functions that allows the MLP to perform extremely nonlinear mappings (GARDNER; DORLING, 1998).

Figure 5.2 is an MLP representation, with the m input features (x_1 to x_m) being presented to a single hidden layer, where it is processed in each hidden neuron with the vector of weights W, added to each respective bias b, and activated with the function f. Next, the hidden layer's outputs are processed in the same way on the output layer single neuron, this time with the vector of weights V and activation function g, when the error e between the network's output o_{out} and the desired output o_d is computed and prepared to be propagated backward, as with the *backpropagation* learning algorithm.

Figure 5.2 - Single hidden layer MLP representation with the error computed in the output layer and ready to be *backpropagated*, aiming the weight vectors update.



The *backpropagation* is a feed-forward neural network training algorithm based on the backward propagation of descending gradients, leading to an efficient weight update reaching all of the network's layers. (GUO, 2013) Taking an MLP with one hidden layer as an example, the network *backpropagation* training process can be summarized in the following steps (GARDNER; DORLING, 1998) (a more detailed description of the *backpropagation* algorithm can be found in Rojas (1996)):

Step 1. The network layers' weights are initialized;

Step 2. The input samples are presented to the hidden layer;

Step 3. The sample is propagated through each hidden neuron, where an output is computed and forwarded to the output layer;

Step 4. The hidden neurons' outputs are propagated likewise through the output layer. The outcome of the output layer, also called network output, is com-

puted and an error signal is calculated by comparing the network output to the desired (target) output;

Step 5. The error signal is used to compute the gradient, which is propagated backward through the network;

Step 6. The weights are adjusted (updated), during the *backpropagation* in order to decrease the error, and

Step 7. Step 2. to Step 6. are repeated until the error is acceptable.

A recurrent neural network (RNN) is an extended type of MLP with supervised training, in which the processing units (neurons) have recurrent connections. The following section describes and RNN and how the scope of this work meets the RNN features. Note that, from now on, traditional MLPs will be referred to as "traditional networks".

5.3 RNN

RNNs are a class of supervised ANN with dynamic behavior, which considers the existing dependency between the input instances. This is what differentiates them from the traditional feed-forward networks, which treat the inputs independently from one another.

This flexibility motivates the adoption of RNNs in problems in which data is sequential, like language processing problems (such as speech recognition and language modeling), machine translation (GULLI; PAL, 2017), price monitoring (PINHEIRO; SENNA, 2017) and time series in general. In all these cases, taking the sequential context of the inputs into consideration can be a differential in the forecast or regression quality.

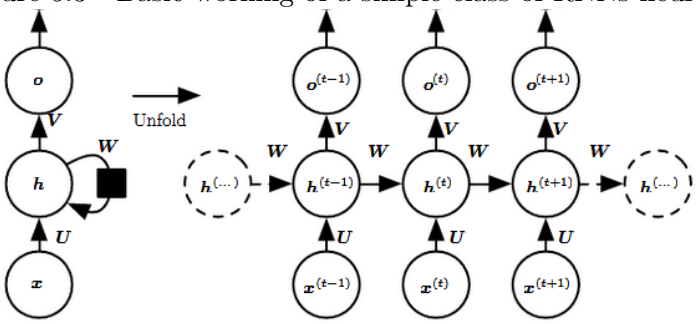
In order to generate internal memory devices, RNNs also have connections between neurons of the same layer, also different from the traditional networks, in which the connections exist only between the different layers. Those connections are an extension in which feedback connections are included to the units, adding a time index t to the process (GOODFELLOW et al., 2016).

Essentially, the RNNs follow the basic functioning of MLPs, regarding the internal processing of neurons. However, besides the traditional matrix of weights, which is applied over the neurons entries, the RNNs have two other weight matrices: one is

used exclusively in processing the feedback from the previous state, and the other is used over the current state to calculate the output of the neuron.

In Figure 5.3, it is possible to visualize the basic process that occurs in a recurrent **neuron** (or recurrent **unit**, as usually called in RNNs) to calculate its current state h in the instant of time (or state) t. It is computed given the unit's input $x^{(t)}$, and its previous state ($h^{(t-1)}$). In the sequence, the weighted current state $h^{(t)}$ generates the unit output ($o^{(t)}$). On the left of the picture, there is a briefer representation; and, on the right, the unfolding of the process considering instants of time from *t-1* to *t+1* is shown. In addition to this information, the figure also distinguishes the three weight matrices, being U the input layer weight matrix, V the output layer weight matrix, and W the weight matrix applied to the feedback signals (GOODFELLOW et al., 2016).

Figure 5.3 - Basic working of a simple class of RNNs neuron.



SOURCE: Goodfellow et al. (2016).

Equations 5.2a and 5.2b formalize the operations indicated in Figure 5.3, where we observe the middle cell receiving and processing the input $x^{(t)}$, along with the previous state $h^{(t-1)}$, generating the current state $h^{(t)}$ and from it computing the unit output ($o^{(t)}$). Therefore, when that same cell receives the next entry $x^{(t+1)}$, it has the memory of the previous state $h^{(t)}$. And so the process continues, successively, until it processes all the input instances.

Equations 5.2 show the weighted sums on which the activation functions f and g are applied, with their respective weight matrices; and the computation of each recurrent unit output.

$$h^{(t)} = f(Wh^{(t-1)} + Ux^{(t)}) \tag{5.2a}$$

$$o^{(t)} = g(Vh^{(t)}) \tag{5.2b}$$

The training algorithm normally used in RNNs is *backpropagation* through time (BTT), adapted from the traditional *backpropagation* used in multilayer perceptrons, which performs gradient descent on a complete unfolded network. 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 which influenced them with the feedback through time. Thus, in BTT, the errors calculated in the output layer, besides being able to be *backpropagated* to the first layers, as happens in *backpropagation*, can still be arbitrarily *backpropagated* to other times instants. This process is called "unfolding" and is represented in Figure 5.3 when the recurrent weight is spatially duplicated for an arbitrary number of time steps (GUO, 2013). A detailed description of the BTT algorithm can be found in works like Werbos (1990).

In BTT, the large number of steps through which the error is retro-propagated generates a drastic reduction in the value of the gradient, approaching zero, resulting in the so-called vanishing gradient problem. The weights stop suffering significant updates, not contributing to the learning process anymore (HOCHREITER, 1998).

The operation of the simple recurrent nets architectures tested in this work is based on the functioning described so far and does not deal with the vanishing gradient problem, which limits the memory offered by this type of net to the 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), described next and also tested in this work.

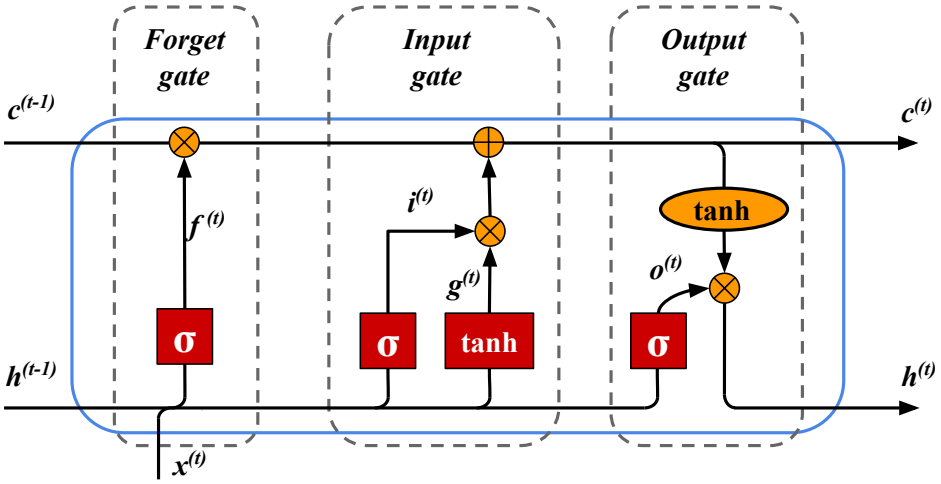
5.3.1 LSTM

LSTM is a special type of RNN, proposed by Hochreiter and Schmidhuber (1997), 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 (HOCHREITER, 1998) .

The LSTM improves the concept of memory, already offered by the basic RNN, by introducing "gates" to the recurrent units, which were included targeting the cell

state protection and control (YU et al., 2019; SAKAR et al., 2019). Different versions of LSTM have been suggested since the first one (GERS et al., 2000; SCHMIDHUBER et al., 2007; JOZEFOWICZ et al., 2015; GREFF et al., 2016; YU et al., 2019), but the version usually used nowadays and also used in this work has three gates: input, output and forget gates.

Figure 5.4 - LSTM recurrent unit gates processing.



SOURCE: Adapted from Yu et al. (2019).

The recursive unit's computation including such gates can be found represented in Figure 5.4, and is described by Gulli and Pal (2017) as follows:

Forget Gate $f^{(t)}$: Defines how much of the previous state $h^{(t-1)}$ should be allowed to pass through (Equation 5.3).

$$f^{(t)} = \sigma(W_f h^{(t-1)} + U_f x^{(t)}) \tag{5.3}$$

Input Gate $i^{(t)}$: Defines how much of the newly computed state for the current input $x^{(t)}$ will be let go (Equation 5.4).

$$i^{(t)} = \sigma(W_i h^{(t-1)} + U_i x^{(t)}) \tag{5.4}$$

Output Gate $o^{(t)}$: Defines how much of the internal state $h^{(t-1)}$ is desired to be

exposed for the next layer (Equation 5.5).

$$o^{(t)} = \sigma(W_o h^{(t-1)} + U_o x^{(t)}) \tag{5.5}$$

These three gates' outputs are activated by the sigmoid function falling in a range from zero to one. After multiplying the output by a second vector, the function will define how much of it will be passed through it.

Internal Hidden State $g^{(t)}$: It is computed based on the current entry $x^{(t)}$ and the previous state $h^{(t-1)}$ of the neuron, using the function *tanh*, similar to what happens in the basic RNN's unit (Equation 5.6). However, in this case, $g^{(t)}$ will be adjusted by the input gate output $i^{(t)}$, during the so-called "cell-state" computation.

$$g^{(t)} = tanh(W_g h^{(t-1)} + U_g x^{(t)}) \tag{5.6}$$

The cell-state ($c^{(t)}$) is given by Equation 5.7, in which a combination is made between the long-term memories and the most recent ones, through the Forgot and the Input gates. This way, their values are weighted in order to ignore the desired memories (with value 0) or make them relevant (with value 1).

$$c^{(t)} = (c^{(t-1)} \otimes f^{(t)}) \oplus (g^{(t)} \otimes i^{(t)}) \tag{5.7}$$

The output $h^{(t)}$ of the neuron is finally calculated as a function of $c^{(t)}$, 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 (Equation 5.8).

$$h^{(t)} = tanh(c^{(t)}) \otimes o^{(t)} \tag{5.8}$$

The works cited in this section have affirmed the efficiency of the LSTM network. Although the learning capability of the LSTM is superior to that of the basic recurrent network, the gate structure inevitably increases the computational cost. To relieve this load, a new and more simplified gate structure has been suggested, the GRU.

5.3.2 GRU

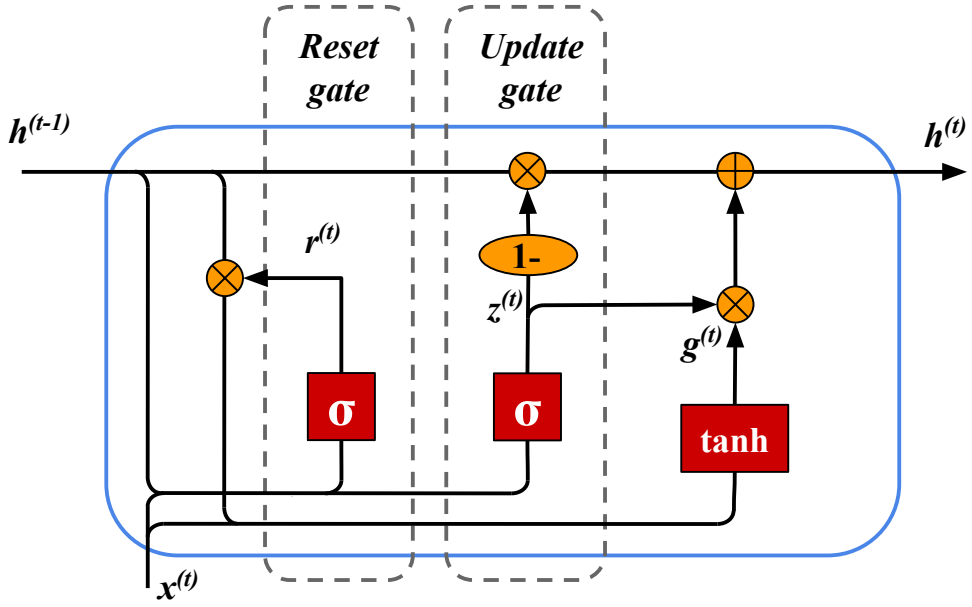
It can be considered that the GRU, proposed by Cho et al. (2014), is a variation of LSTM and very similar to it, but having a considerably simpler gates structure than LSTM. GRUs are composed of only two gates:

- **Update** $z^{(t)}$**:** Integrates the LSTM forget and input gates, which is able to be trained to define how much of the older information should be kept (GULLI; PAL, 2017), and

- **Reset** $r^{(t)}$**:**: Able to be trained to merge the new information entries with the previous memories (GULLI; PAL, 2017).

Figure 5.5 describes its architecture and Equations 5.9 given by Gulli and Pal (2017) summarize the computations that take place in GRU cells.

Figure 5.5 - GRU recurrent unit architecture.



SOURCE: Adapted from Yu et al. (2019).

$$z^{(t)} = \sigma(W_z h^{(t-1)} + U_z x^{(t)}) \tag{5.9a}$$

$$r^{(t)} = \sigma(W_r h^{(t-1)} + U_r x^{(t)}) \tag{5.9b}$$

$$c^{(t)} = tanh(W_c(h^{(t-1)} \otimes r^{(t)}) + U_c x^{(t)}) \tag{5.9c}$$

$$h^{(t)} = (z^{(t)} \otimes c^{(t)}) \oplus ((1 - z^{(t)}) \otimes h^{(t-1)}) \tag{5.9d}$$

It is worth mentioning that all equations regarding RNNs presented until now have the option of providing a bias to the sums, in the same way, that was indicated in Equation 5.1 for the ANN basic neuron's processing.

According to some works, like in Jozefowicz et al. (2015), LSTM and GRU are networks that have shown themselves to be very similar in terms of performance. Although, according to other works, like in Sakar et al. (2019), the single GRU cell is not as powerful as the LSTM, being limited to solve some language processing problems. In this work, both of them, as well as the Simple RNN, are tested to see which one brings better results for TSI and SSI forecasts.

To proceed with the *Python* language in the final phase of the workflow, it was necessary to choose a library that offered all three RNN architectures. For this work, we decided to adopt the most popular *Python* open-source software library: *Keras*.

5.3.3 *Keras* RNNs

Keras[1] is a modular *Python* library that runs on top of *Tensorflow* or *Theano* backends (GULLI; PAL, 2017). It is the most popular library among the *Python* user community, which has complete and accessible documentation, guides, and examples (KERAS, 2021).

Keras is said to be a modular user interface because its core data structure is made of models and layers, where a model could be of a type already available in the library or also a linear stack of chosen layers customized by the user in a friendly way (KERAS, 2021). For instance, ANNs can be models built by layers as blocks. More detailed information about *Keras* available layers and how to build a model

[1] *Keras-Tensorflow* webpage: https://www.tensorflow.org/guide/keras?hl=pt-br, accessed in July 2021.

or use a pre-existing model can be found on *Keras* documentation (KERAS, 2021).

Keras has seven types of recurrent layers available, and we chose the ones that follow the architectures described previously to be used in this work:

a) **SimpleRNN** layer;

b) **LSTM** layer, and

c) **GRU** layer.

The default hyper-parameters adopted by *Keras* for the three layers will be informed in the next section, which will list some of the nets' parameters that need to be set before the training process is run.

5.4 RNNs' training parameters and metrics

The search for satisfactory performance in a neural network depends on performing a series of tests - which we will call "experiments" to avoid confusion with the term "tests" from the testing phase of RNNs. Even when there is certainty about an ANN architecture, experiments are essential for hyperparameter tuning, to obtain the best set of hyper-parameters that will improve the network performance.

The experiments seeking this set started, in this work, right after the selection of the data used in training, validation, and testing - phases that will be briefly described in the following subsection. And it is during these two initial phases - training and validation - that the hyper-parameters are adjusted. Some of them, the ones explored in this work, will be described next.

5.4.1 Training, validation, and test

In addition to the characteristics of neural networks that are similar to biological neural systems mentioned earlier in this chapter, another characteristic of ANNs with the supervised algorithm is the way learning occurs: through the phases of training, validation, and testing. We can make an analogy to traditional school teaching, where a student must learn some content of a subject. First, the content is presented, with questions and their respective answers, so that the student trains himself and understands the logic behind the content; then, a list of exercises is given, so that the learning acquired during the first stage is validated; and finally, an evaluation is applied, so that the acquired learning is tested to reach a conclusion

if the learning was effective or not. With neural networks, the "learning" occurs in a similar way, with the following steps:

Training step: Initially input examples and their respective desired outputs are presented, that is, data that has already been observed is shown to the network, in which the output of each input sample is already known. By comparing the output computed by the network with the real and desired output, the error is calculated and used as the basis for adjusting the neurons' weights. These steps are repeated until the error reaches an acceptable minimum, as described in this chapter when the *backpropagation* algorithm is explained. The training process is when the network fits the data, aiming for a good generalization. Generally, 80% of data are used for training, and the other 20% are held off for test (or validation) (PRADHAN; LEE, 2010; KALANTARY et al., 2019; NETO et al., 1995).

Validation step: Occurs after or during training (cross-validation), with a data set that is independent of the one used for training. The difference between validation and training processes is that the validation error is not propagated back and is not used in the adjustment of the weights. Validation can be done in two ways: at the end of the training, or at the end of each iteration of training, when it is called cross-validation. Either type allows analyzing the training effectiveness by checking if the network is adjusted well enough to provide a good generalization, that is, the network is not biased. This can help to identify the over-fitting, which occurs when a network is so specialized in the data presented during training that it cannot perform well on, or generalize, the unknown validation data. With cross-validation, validation can be followed up with training, giving indications of generalization or over-fitting. The validation step is also useful to tune the network's hyper-parameters aiming the model final version (RIPLEY, 2007).

Test step: If the validation step was used to tune the network, a test set is used to assess the training fit quality through an unbiased estimate of generalization error. The testing phase performs a final check with a new data set, to confirm (or not) that the mapping was successful and that the network is ready to be activated, i.e. to be used to make consecutive predictions without the need for evaluations (RIPLEY, 2007; KUHN et al., 2013).

Some works consider the validation as an optional step (RUSSELL; NORVIG, 2002; KUHN et al., 2013), focusing more on training and testing, which was initially performed in this work. After some inconsistencies in the results (that will be reported in the results chapters), it was decided to proceed with the three stages, with a data

set for training and validation (in the proportion of 80% and 20%, respectively) and another new set exclusively for testing. In the final experiments for nets' performance evaluation, an extra set was used exclusively for the test phase to reinforce the quality of the results. In *Keras* architectures the use of cross-validation is optional, and we opted to use it.

Before starting the training phase, some of the network parameters were chosen to be varied in a search for values that would provide the lowest validation and testing error. They will be briefly defined in the next subsections, and the experiments on varying them will be described in the results chapter.

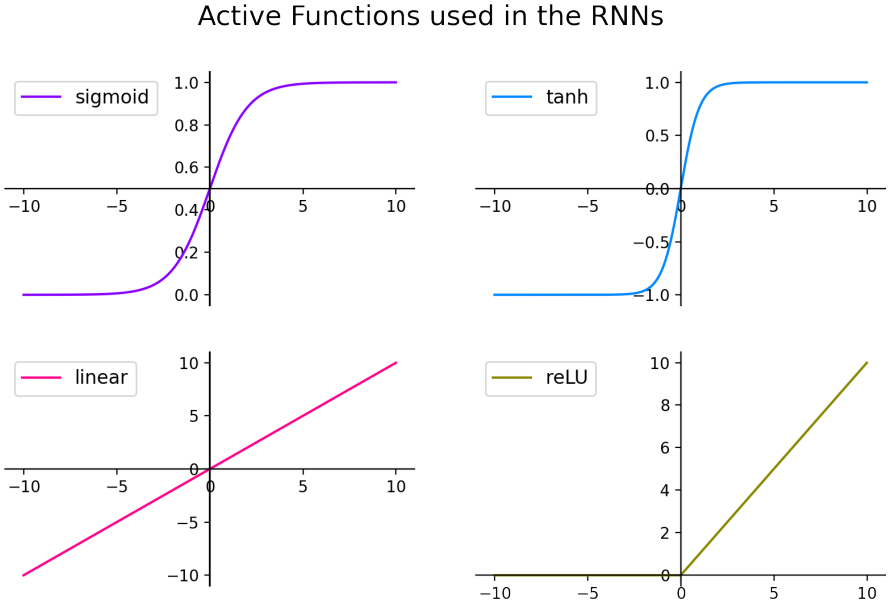
5.4.2 Activation functions

An activation function or transfer function is basically a linear or non-linear mathematical function used over the internal signal of an artificial neural unit to keep it in the desired range. In any artificial neural network, an activation function is applied over the sum of the inputs multiplied by their corresponding weights. Without such a function, it would not be possible to recognize non-linear mappings between input and output data (SHARMA; SHARMA, 2017).

There are several functions that are used in ANNs, but some of them are more commonly used in RNNs, depending on the units' internal process part. Table 5.4.2 presents some of them, including the ones that are present by default in the *Keras* RNN models, with their derivative function and the values' range each one gives as output. Those functions can also be found plotted on Figure 5.6, in which the range $x = [-10, 10]$ was taken initially as a sample to the plot, with the intention of displaying their pattern.

Function	**Equation**	**Derivative**	**Range**
Hyperbolic Tangent	$f(x) = \frac{2}{1+\exp^{-2x}} - 1$	$f'(x) = 1$	$(-1, 1)$
Linear	$f(x) = x$	$f'(x) = 1$	$(-\infty, \infty)$
Sigmoidal	$f(x) = \sigma(x) = \frac{1}{1+\exp^{-x}}$	$f'(x) = f(x)(1 - f(x))$	$(0, 1)$
Relu	$f(x) = \begin{cases} 0 & \text{for} x < 0 \\ x & \text{for} x \geqslant 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for} x < 0 \\ 1 & \text{for} x \geqslant 0 \end{cases}$	$(0, \infty)$

Figure 5.6 - Some activation functions plotted for the range $x = [-10, 10]$.



As seen in the description of RNNs, the two activation functions generally used by default in the internal processing of recurrent units are the sigmoid and the hyperbolic tangent, both of which are also defined as standard in *Keras* recurrent models. In the output layer (*Keras Dense* layer[2]), a linear function is usually used.

5.4.3 Data scaling

In the context of ML techniques, scaling an input or output data means re-scaling its values into a common predefined range, in order to avoid computation between features in very discrepant ranges.

The scaling is not a mandatory pre-process, but some works, like Sola and Sevilla (1997) and Jin et al. (2015) indicates that an adequate data scaling process can reduce estimation errors significantly, besides reducing the time needed in the training process to obtain satisfactory results.

There are basically two types of scaling:

- **Normalization**: when the data is scaled to a predefined range. The range

[2] *Keras Dense* layer documentation: `https://keras.io/api/layers/core_layers/dense/`, accessed in August 2021

$[0, 1]$ is the one usually used, but other ranges can be adopted. A normalized value x_n, in the range $[0, 1]$, of an element x_i of vector X is given by Equation 5.10, where X_{min} and X_{max} are, respectively, X's minimum and maximum values.

$$x_n = \frac{x_i - X_{min}}{X_{max} - X_{min}} \tag{5.10}$$

- **Standardization**: when the data is scaled to have zero as mean and one as standard deviation. A standardized value x_s of the element x_i is given by Equation 5.11, where μ and σ are, respectively, X's mean and standard deviation values.

$$x_s = \frac{x_i - \mu}{\sigma} \tag{5.11}$$

5.4.4 Loss and accuracy metrics

On the task of comparing the networks' performance, it is important to define some functions that will be used during the training step to calculate the loss in each training epoch or during validation aiming optimizing the networks' fitting process.

The loss function is used to compare the network processed values with the real desired values in the output layer, for each input sample, in each iteration. A frequently used loss function is the mean squared error (MSE), represented in Equation 5.12, which was chosen to be adopted in this work to measure the LSTM and GRU performance internally during training and validation steps. The decision to use MSE as the loss function resulted from the experiments performed to define the hyperparameters, also explained later in this subsection.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5.12}$$

where n is the training set size (or test set size, when used to evaluate the prediction), y_i is the real and desired output value for the i^{th} sample, and $\hat{y}_i$ is the network predicted output value for sample i.

To evaluate the prediction quality, in the test step, we initially chose to compare four different measurements; two of them to measure accuracy (the level of correctness) and the other two to measure the average loss rate in the predictions. MSE was

chosen as a loss function because it punishes the predicted values that are more distant from the real values.

The accuracy measurements were the linear correlation coefficient (denoted as R) and the coefficient of determination (denoted as R^2), which indicates fit goodness. R^2 is a percentage measure of how well the unseen test samples are likely to be predicted by the model (WALT et al., 2014) and can be estimated as denoted in Equation 5.13.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{5.13}$$

where $\hat{y}_i$ is the predicted value of the i-th sample, y_i is the corresponding real value, and $\bar{y}$ is the real set mean value, defined in (Equation 5.14).

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{5.14}$$

The other loss function, besides the MSE, was the mean absolute percentage error (MAPE), estimated by Equation 5.15), a forecast model accuracy measure that provides, through a percentage value, a relative perspective of the test loss.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{5.15}$$

where y_i is the real value and $\hat{y}_i$ is the predicted value.

5.4.5 Optimizer function

The other relevant function that had to be chosen for the experiments was the optimizer function, which works with the *backpropagation* technique on a seek to achieve the minimum loss and the best fitting weights. The basic optimizer technique the *backpropagation* commonly uses is the gradient descent function, which can be represented in two versions: the one that uses the stochastic gradient descent (SGD) function (BOTTOU, 2012), and the one that uses the adaptive moment estimation (ADAM) (DA, 2014). The ADAM function is derived from SGD and the main difference between both is how they treat the learning rate: while SGD uses only a single learning rate for all features present in the chosen data set within a batch, ADAM adapts the learning rate to each feature. Once this work uses forty input features, we concluded that the ADAM optimizer would be the best alternative initially to train the LSTM and GRU networks.

5.4.6 Batch size

Batch size is the number of training samples presented in each training epoch. It influences the network's weights adjustment, once its effectiveness depends on which samples are presented in each epoch.

There are basically three options of batch size setting (ACADEMY, 2021):

- **batch mode:** when the batch consists of all the training samples, i.e. the batch is unique and has the same size as the training sample.
- **mini-batch mode:** when the batch has a size between one and the training sample size.
- **stochastic mode:** when the batch size is one, and therefore the gradient is computed and the weights are updated after each sample presentation.

Some authors have observed higher accuracy for smaller batch sizes (MASTERS; LUSCHI, 2018; ACADEMY, 2021). Following this finding, no batch mode tests were performed. But the other two options were included in the experiments, which will be described in the next chapters.

There are other hyper-parameters that have been poorly explored, for not having initially indicated significant improvement to the performance of the networks, but which could be properly explored in more specific works. This is the case of the parameters briefly described below.

5.4.7 Dropout

It is a regularization technique[3] that seeks to improve generalization and reduce overfitting. Its strategy consists of randomly dropping some units along with their connections during network training (SRIVASTAVA et al., 2014). In *Keras*, the fraction of units to be dropped is defined through the dropout rate, which is a value between zero and one.

The strategy of using dropout suggests that this technique gives results on larger neural network configurations, with many processing units or many layers (ZAREMBA et al., 2014; SRIVASTAVA et al., 2014; GOODFELLOW et al., 2016). This characteristic does not match the profile of the network configurations used in this work.

[3]Regularization technique is a strategy added to the training algorithm aimed at improving the test error. (GOODFELLOW et al., 2016)

5.4.8 *Lookback*

By default, input instances are presented to a supervised neural network individually, along with their corresponding desired output. That is, for a one-time ahead prediction, only the input at time t (*lookback*=1) is presented along with its corresponding desired output at time $t+1$. A *lookback* greater than one would be to present a history of more than one input, yet for only one desired output. For example, for a *lookback* of three, inputs at time t, $t-1$, and $t-2$ would be presented to the network to predict the output at time $t+1$, and inputs at $t+1$, t, and $t-1$ would be presented to predict the output at $t+2$, and so on.

Initially, some experiments were performed varying the *lookback* number (some of which will be presented in the next chapter) and showed a reduction in training and validation error as the *lookback* number was increased. (MURALIKRISHNA et al., 2020b; MURALIKRISHNA et al., 2020c) However, these first experiments were limited to the training and validation phases only, and no testing phase was performed until then. In later experiments - which included the testing phase - differently from what was expected, increasing the *lookback* number did not result in improvements in prediction. The reason why the testing phase did not reproduce the resulting profile obtained during training, in this case, is still unknown.

6 RUNNING THE WORKFLOW: EXPLOITING THE SOLAR IRRADIANCE PREDICTION WITH RNNS

Once the complete workflow was developed, it became possible to validate its main task: the prediction of solar irradiance. This chapter will describe the first group of experiments, whose main objective was to validate the prediction of TSI and three SSI emission lines - two of helium and one of hydrogen - at different prediction time intervals, while searching for the best recurrent neural network architecture and its best hyper-parameters set.

The prediction quality and experiments of the original workflow can be checked at Vieira et al. (2011b), and it is important to clearly state that the purpose of this work is not to make a comparison between the results obtained there and the ones obtained here. Despite the fact that we are dealing with RNN packages developed for different programming platforms, the prediction was performed using the same ML technique, and we are not aiming to compare distinct packages of the same technique.

Some of the experiments shown in this chapter were presented and published in conferences proceedings (MURALIKRISHNA et al., 2020a; MURALIKRISHNA et al., 2020c; MURALIKRISHNA et al., 2020b), but until that moment only the networks' training and validation steps' results had been reported. The main results presented in this chapter are the results of an article submitted to the JSWSC (Journal of Space Weather and Space Climate[1]), in which the test phase was included. All the experiments that will be presented in this chapter were regenerated, when necessary, with the testing phase included.

The chapter will start by describing the networks' pre-training setup, which involves data selection and fault coverage, when necessary, and then will inform the networks' settings which were adopted as default for most of the experiments. Following this, the experiments will be listed and their results will be discussed.

6.1 Data sets

The RNNs, as a supervised technique, compare real data to the values calculated by the network in the output, to adjust the internal weights and repeat the processes until the error between real and predicted is acceptable. To do so, they go through

[1] Journal of Space Weather and Space Climate webpage: https://www.swsc-journal.org/, accessed in August 2021.

three phases: training and validation, which are used to reach convergence; and testing, which allows the network's performance to be evaluated for another still unseen data set.

Given this, it is important to carefully select the data sets that will be used during the training, validation, and testing phases, so that they will represent well enough the data that is expected to be predicted in future network activation. Therefore, the experiments started with defining the data periods and its temporal resolution used in the experiments.

Regarding the data resolution, we have photosphere images and irradiance data in different resolutions available for download.

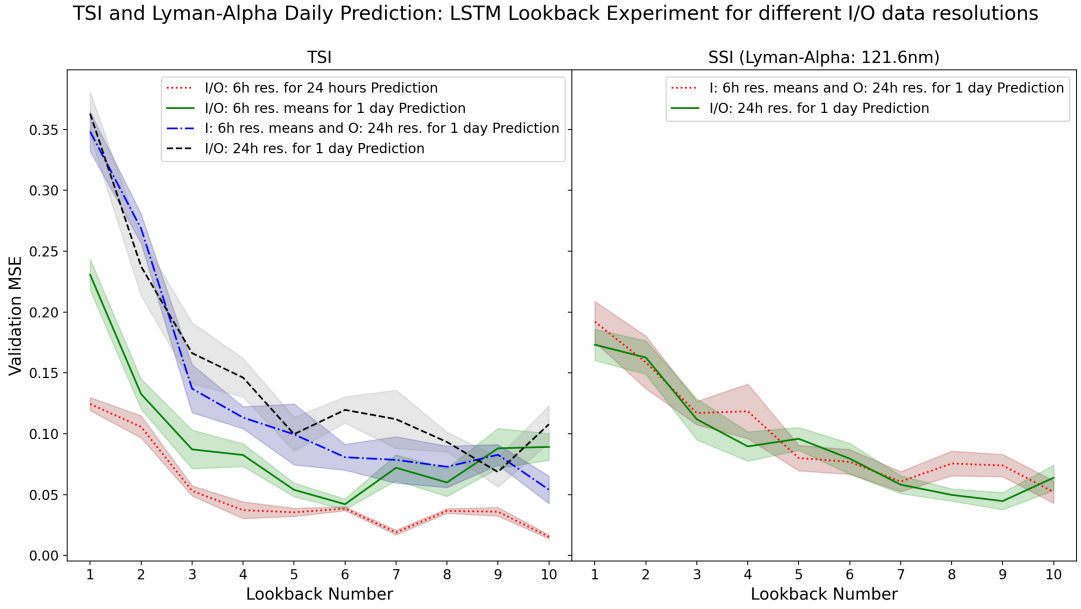
The minimum image resolution effectively processed by the workflow is the six-hour resolution, since images in a higher resolution would have to be interpolated to fit the six-hour resolution. For predictions of six hours and multiples of six (twelve and eighteen hours) ahead, this resolution was unavoidable. However, for daily forecasts, there was the option of using the one-day (or 24 hours) resolution for the images.

For the output data, there were the options of six-hour or 24-hour resolution for the TSI, and only the 24-hour option for the SSI emission lines.

Taking these options as a basis, an experiment was conducted to decide which resolution would be the best to use in order to proceed with the other experiments. The plot in Figure 6.1 shows the comparison between different combinations of resolution for both input and output in a simultaneous *lookback* variation experiment. For TSI input and output data, we see better results using six-hour resolution for both; and for SSI, we don't see a significant difference between the two options. The *lookback* variation results analysis will be done afterwards with other *lookback* results.

After analyzing the Figure 6.1, we decided to use six hours resolution for input data in all the experiments. For the SSI and TSI series, two different resolutions were taken: for TSI, the resolution was of six hours for all the experiments, and for SSI, the resolution was of 24 hours (daily average).

Figure 6.1 - *Lookback* variation experiment with different combinations of input and output data resolutions for TSI and SSI.

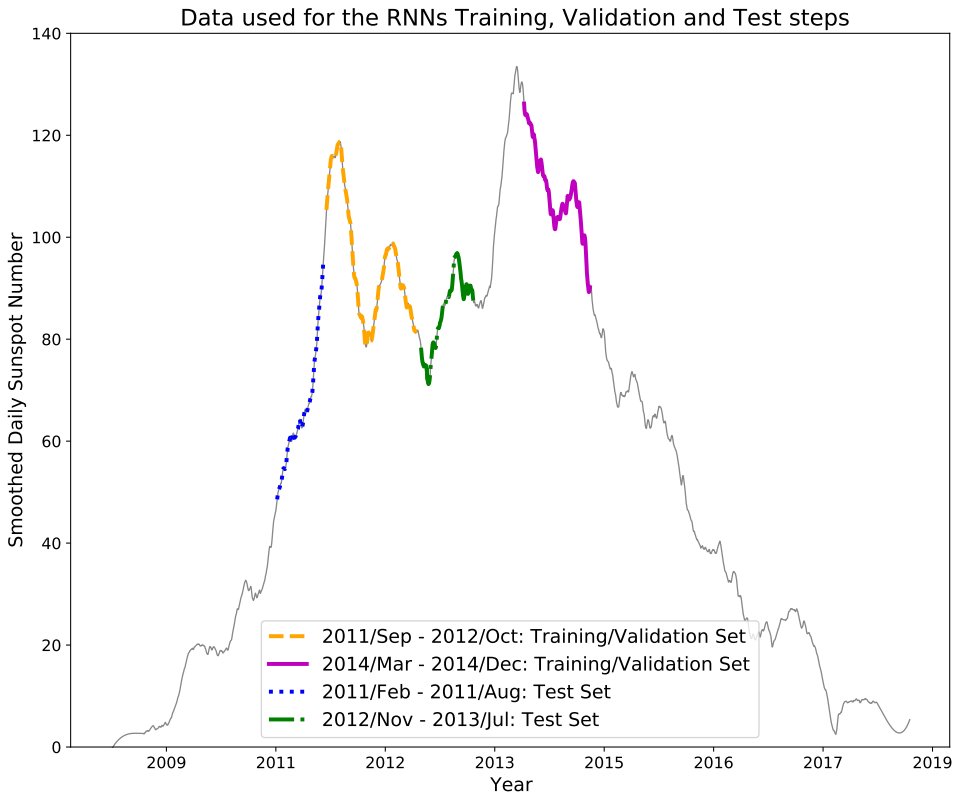


In order to avoid large failure periods and, thus, also avoiding the need for large interpolations, the periods of the data sets (Figure 6.2) were chosen taking into consideration the continuous availability of both: the input images and the time series used in the output. For the output data, interpolations were used only to fill a maximum of four consecutive time instants of failure. On the other hand, missing images were replaced with 15-minute resolution images of up to three hours back and forward. However, image replacements and data interpolations were required for less than 0.5% of the data.

As data, a time window around the last solar cycle (24^{th}) maxima was chosen. It is composed of data sets between 2011 and 2014. The Figure 6.2 shows those data sets and the purpose each one was used for: to train and validate or to test the recurrent network models.

For those experiments that reached the test phase, at least two data sets were used: one continuous or merged data set split randomly in training and validation sets, and one other set separated exclusively for the test step. The networks' performance was initially analyzed by looking at the validation loss, which is used during training in the cross-validation process to evaluate whether the weights adjustment is going in the direction of the smallest loss but avoiding the over-fitting problem.

Figure 6.2 - Data intervals used for the RNNs training, validation and test steps, considering the period near the 24^{th} solar cycle maxima.



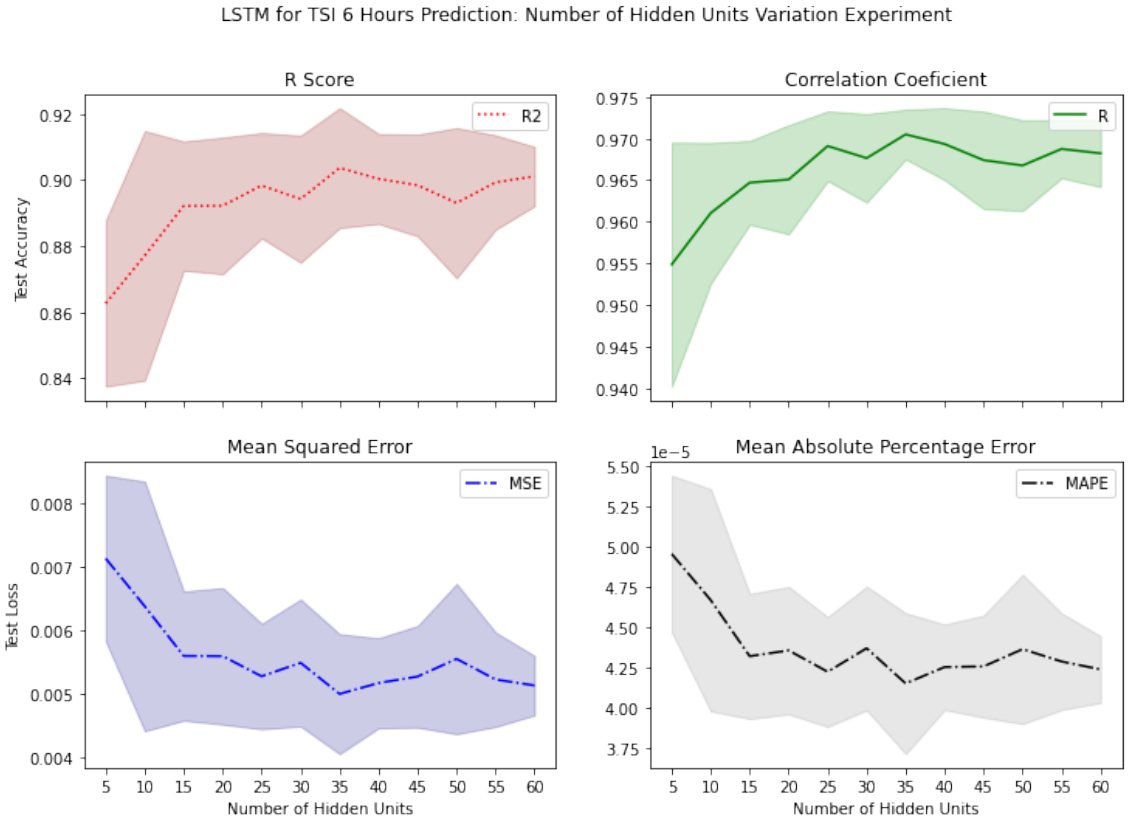
6.2 Defining initial hyper-parameters

Before actually starting the experiments, some individual and arbitrary training processes were conducted in a search of an initial configuration set that would provide convergence on the training processes and also reduce the prediction loss. These tests were performed on the network that predicts TSI 6 hours ahead, and what appeared to be the best parameters at that moment were used on this network and extended initially to all the other networks, which performed the other predictions. Gradually, more systematic experiments were performed in order to tune the networks and find new values for those settings.

For instance, to define an initial number of hidden units in the recurrent layers, an experiment was carried out with the LSTM net varying the amount from five

to 60 units. The average loss (MSE and MAPE) and average accuracy (R and R^2) obtained in each training run (20 runs for each value of hidden unit) can be seen in Figure 6.3, along with the standard deviation obtained in each group of executions. The averages are represented by the strongest lines in the center of each plot, and the standard deviations can be viewed by the weakest colored areas above and below the average lines. The four plots show that 35 hidden units brought better performance initially.

Figure 6.3 - Initial experiment varying hidden units in an LSTM for predicting TSI 6 hours ahead.



SOURCE: Muralikrishna et al. (2020c).

In the same way that the hidden units were varied, other experiments were also performed. However, those experiments were not performed in a systematic way and were not recorded properly, because they were the first experiments. Some of such experiments were made by:

- Adding dropout layer, with different dropout rates;
- Varying the number of training epochs;
- Adding one more recurrent layer;
- Varying the batch numbers;
- Varying the time steps (*lookback* number), among others.

Those experiments were essential for defining an initial set of hyper-parameters that were used in the systematic experiments that followed. Some of them were still varied again in a second set of experiments, that will be reported in the next sections.

The main settings and hyper-parameters values resulted from this initially experiments are listed on Table 6.1 as those were used as default in most experiments of this chapter. If different values were used for such hyper-parameters, it will be explicitly mentioned.

Table 6.1 - RNNs default hyper-parameters.

Settings/Hyper-Parameters	**Value**
Input Features	**40**
Output Size	**1**
Scaling method for Input Data	Normalization [-1,1]
Scaling method for Output Data	Normalization [0,1]
Train/Validation ratio (%)	**80/20**
Use of bias	**Yes**
Number of training run for each experiment	**20**
Hidden Layer	1
Hidden Units number	35
Hidden Layer Activation	relu
Recurrent Step Activation	**sigmoid**
Output Layer Activation	**linear**
Training Loss Function	MSE
Training Optimizer	ADAM
Dropout rate (%)	0
Batch size (% of the training samples)	5
Time step	1
Epochs	100
Loss metrics	**MSE, MAPE**
Accuracy metrics	**R and R^2**

Some of those hyper-parameters values were tested again further searching for best fit values, but initially, we chose to standardize the hyper-parameters for all experiments intending to make the comparison between the experiments more meaningful.

Other hyper-parameters' values were just defined and not changed anymore, like some that came as standard in the *Keras* models. They were kept with the standard values from the beginning to the end of the experiments, with no changes. The activation function used over the recurrent step is one of that hyper-parameters that were not changed, which by default in *Keras* layers is the sigmoid function. The *Keras* default option of using bias is another example that was not changed.

Other setup options, such as the training/validation ratio, were defined once and kept unchanged until the end. The data sets used to train the networks were split into 80% for training, and 20% for validation. Another practice defined from the beginning, which remained unchanged throughout the experiments, was the performing of twenty training sessions for each different configuration run.

The hyper-parameters that were unchanged in all experiments are in bold in Table 6.1.

6.3 Simple RNN vs LSTM vs GRU

The experiments were started by comparing the TSI prediction performance six hours ahead through three different *Keras* architectures, described in Chapter 5: Simple RNN, LSTM and GRU. Simultaneously with the comparison of the architectures, some network hyper-parameters were also tested in order to conduct the following experiments using the best values that are found along the way. Thereafter, new experiments were conducted according to the results that were being observed.

6.3.1 Hidden units number variation

This first experiment re-evaluate the number of hidden units to see how the different architectures would behave with this variation. The units were increased with step one, from 1 to 50, initially; and from ten units, the increasing step used was five, until reaching 50 units. The experiment had the intention of finding the number that gave the minor test loss and checking if the different architectures deal equally with the hidden units number variation.

The Figure 6.4 shows those results for TSI, where they are split in two boxplots at different MSE ranges to reveal the spreads in the larger numbers of hidden units

results too. The results for nine and ten hidden units were repeated in the second boxplot for visual comparison.

When adopting one and two neurons, Simple RNN and GRU showed a large variation in performance over the twenty training rounds compared to LSTM. From the adoption of three hidden units, GRU followed LSTM, which kept errors in a smaller range of variation during the increase of hidden units. In general, it can be observed that the Simple RNN presented a greater instability when compared to those that deal with the vanishing gradient problem, the LSTM and the GRU, and generated more dispersed results, and also had not a continuous performance increase pattern that followed the hidden units increase, as occurred with the other two architectures.

Figure 6.4 - Comparison between Simple RNN, LSTM and GRU RNN architectures, with hidden units number variation experiment in a TSI 6 hours forecast.

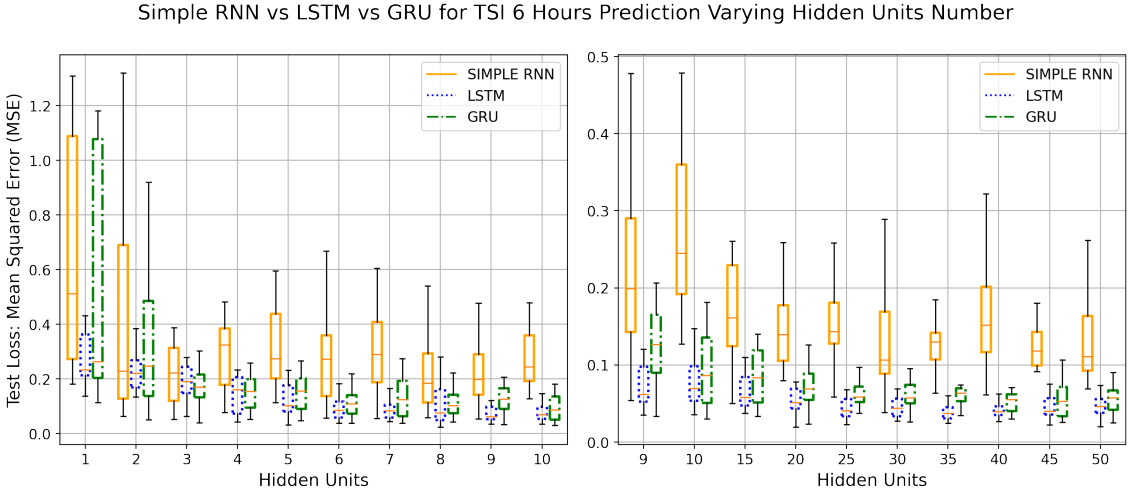
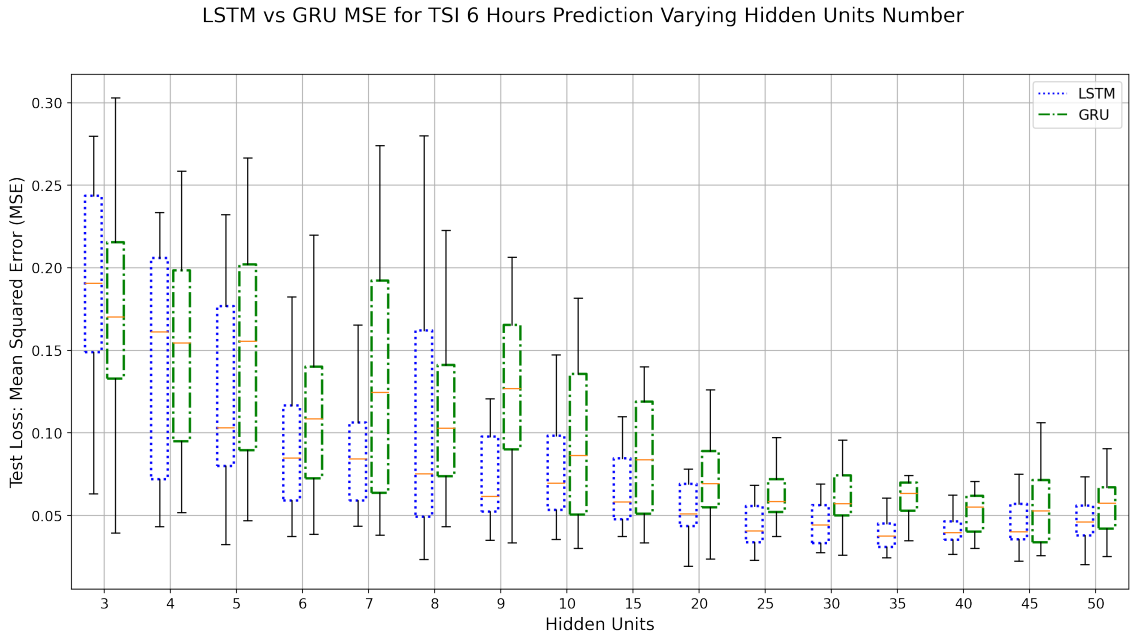


Figure 6.5 shows the same experiment's result, but this time focusing only on the results of LSTM and GRU networks, starting with three hidden units. As observed, both networks showed similar performance in all cases, but LSTM shows a more defined trend of error decrease as the number of hidden units increases. GRU also presents that pattern, but not as well defined as LSTM. More hidden units could be used to valuate this MSE decreasing pattern, but for 40 to 50 hidden units, it can be seen that the MSE neither varies nor decreases.

Figure 6.6 shows the same results of Figure 6.5 but now showing more clearly the distribution of each training group. Looking at the figure, we observe that GRU

Figure 6.5 - Comparison between LSTM and GRU RNN architectures, with hidden units number variation experiment in a TSI 6 hours forecast.



generates a little more distributed errors during the training groups in most of cases. But there is no continuous or general pattern that differentiates the two networks' performance.

The same hidden units variation experiment was made for the emission line of $121.5nm$. Figure 6.7 shows the first result, with a boxplot comparing the three RNN architectures. Although overall we once again observe better performance for the gated networks, in few cases, the Simple RNN shows similar performance to both.

Figure 6.8 shows once again that the gated networks in general have very close performance. To extract from the experiment a better value of hidden units for the next SSI experiments, the LSTM performance was taken as a basis, through the performance measures shown in Figure 6.9, in which four, five and 35 units seem to represent the best results.

Figure 6.6 - Comparison between LSTM and GRU RNN architectures, with hidden units number variation experiment in a TSI 6 hours forecast, this time with plots of the distribution of each run set.

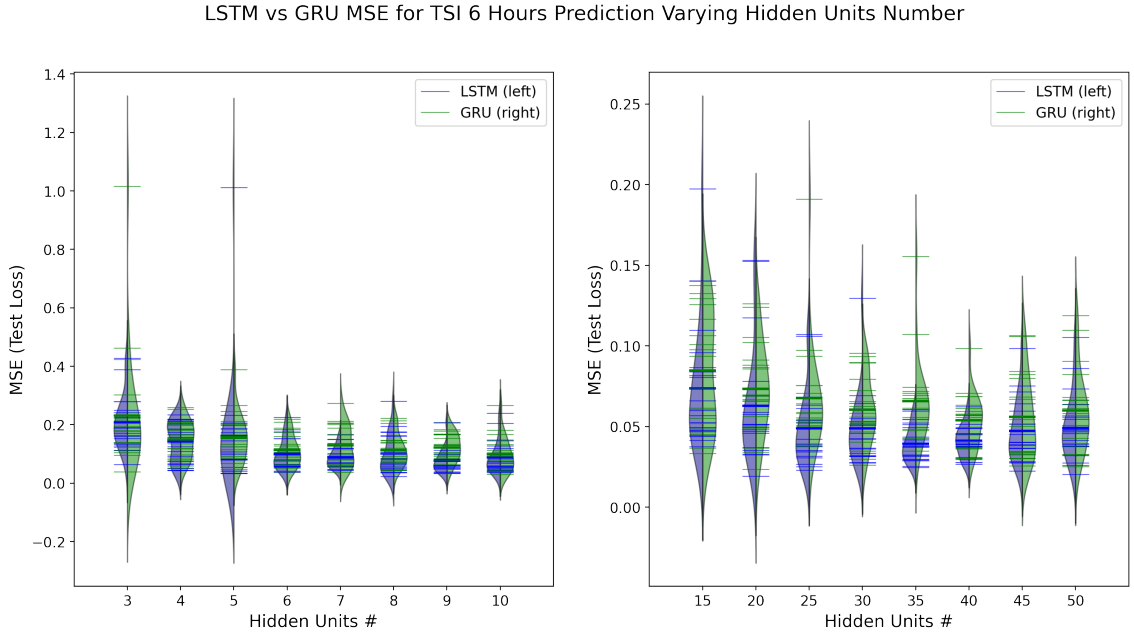
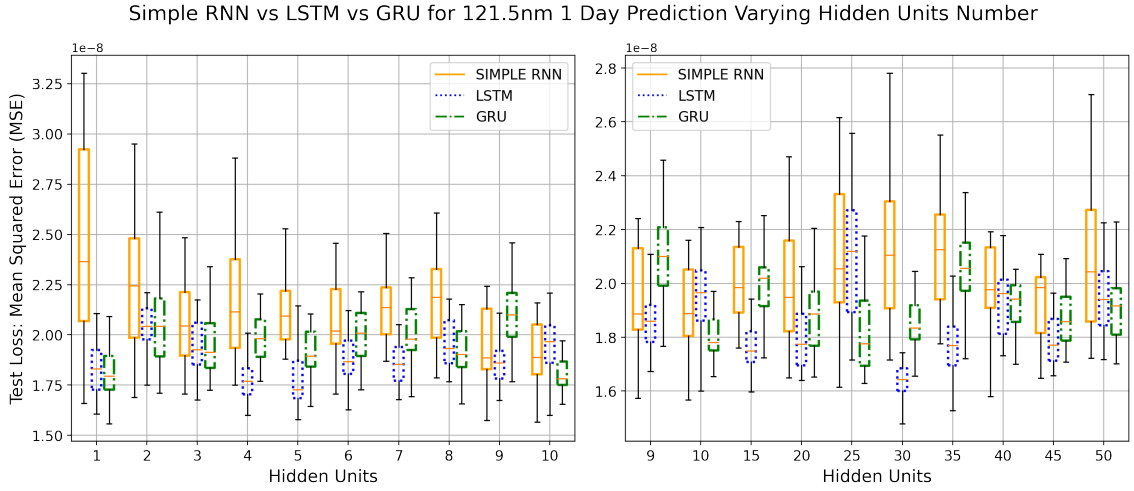


Figure 6.7 - Comparison between the three RNN architectures, with hidden units number variation experiment with emission line of 121.5nm to 1 day forecast.



6.3.2 Different activation functions for input

For this experiment we tried *tanh* function as the input activation, that was already the default option for *Keras* RNNs layers, and in additional we tried two other

Figure 6.8 - Comparison between the gated RNN architectures, with hidden units number variation experiment with emission line of 121.5nm to 1 day forecast.

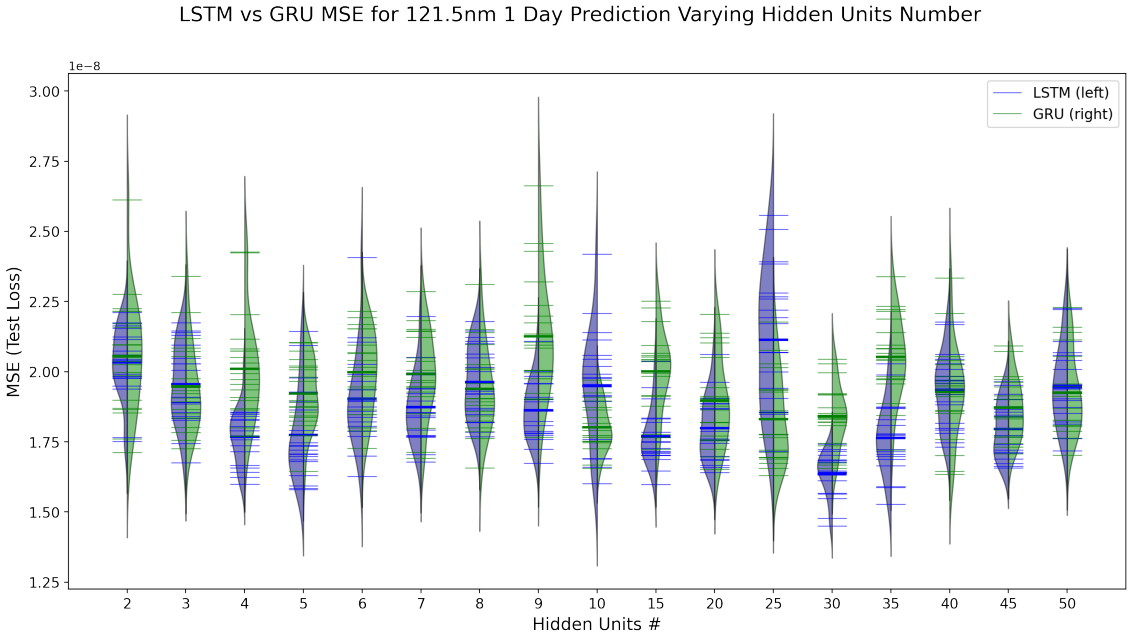
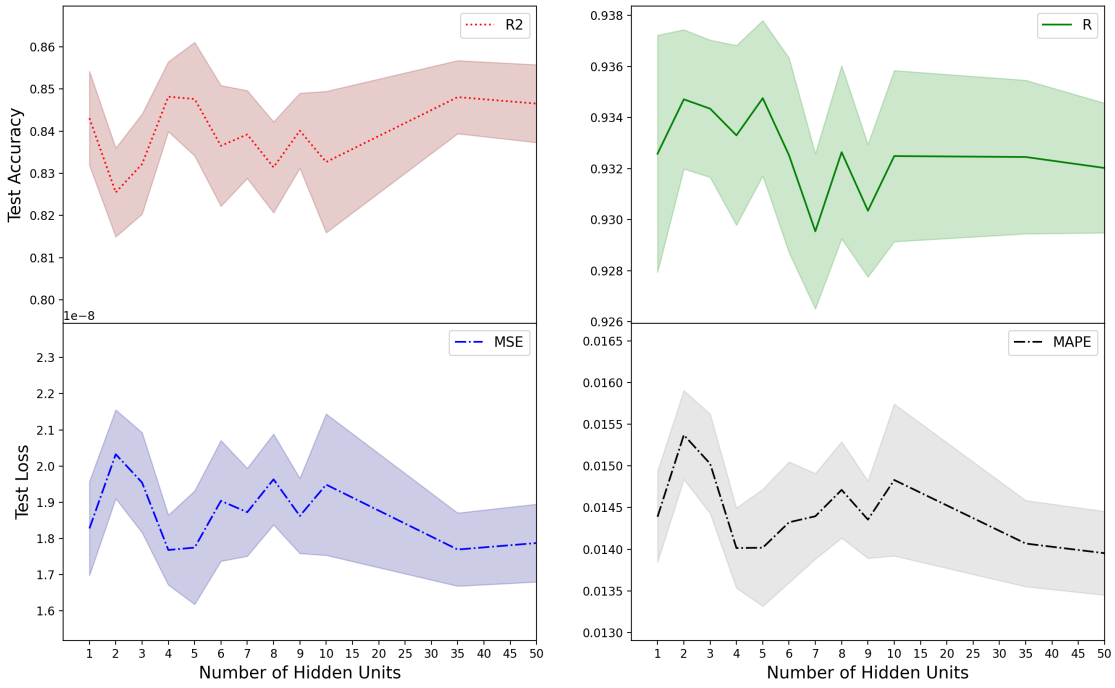
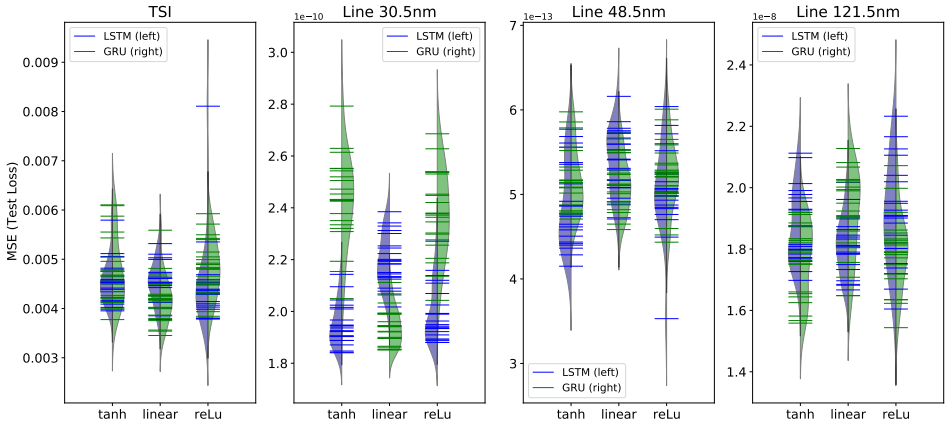


Figure 6.9 - LSTM hidden units experiment performance measures for 1 day forecast of emission line $121.5nm$.



functions: $reLU$ and *linear*. Figure 6.10 presents the comparison between those three functions for TSI and the three emission lines.

Figure 6.10 - Comparison between three different activation for the RNN layer input. The experiment also compared the gated nets and shows the results for TSI six hours prediction and for the SSI emission lines one day prediction.



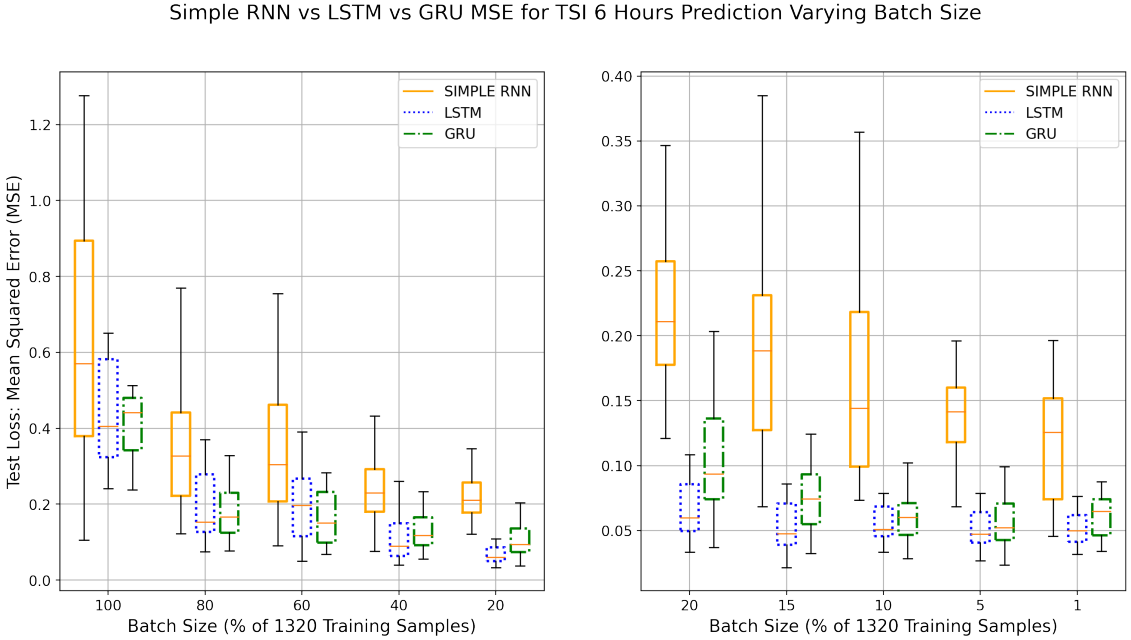
Overall, the three activation functions did not show a significant difference between them in the results. The 30.5nm emission line showed a curious result regarding the difference between the performance between LSTM and GRU: the activation functions that generated better results with LSTM were just the ones that generated lower results with GRU, and vice versa. Such contrast brought with it the greatest difference between LSTM and GRU results, when comparing to the responses of the other emission lines, for which the little difference between the gated networks was confirmed.

6.3.3 Batch size variation

Still looking to tune the networks, and comparing the three RNN architectures performance at the same time, other experiment was carried out, this time varying the batch size. The different values for batch size were calculated as percentages of the training set size, which for this experiment consisted of 1320 samples. The initial batch size value was 100% of the training set size, which was gradually (with a step of 20%) reduced until 20%, and from there on, with a 5% step until reaching 1% of the samples. In a same way of the hidden units experiment, first we compared the three nets results (Figure 6.11), splitting in two boxplots, and once noticing the

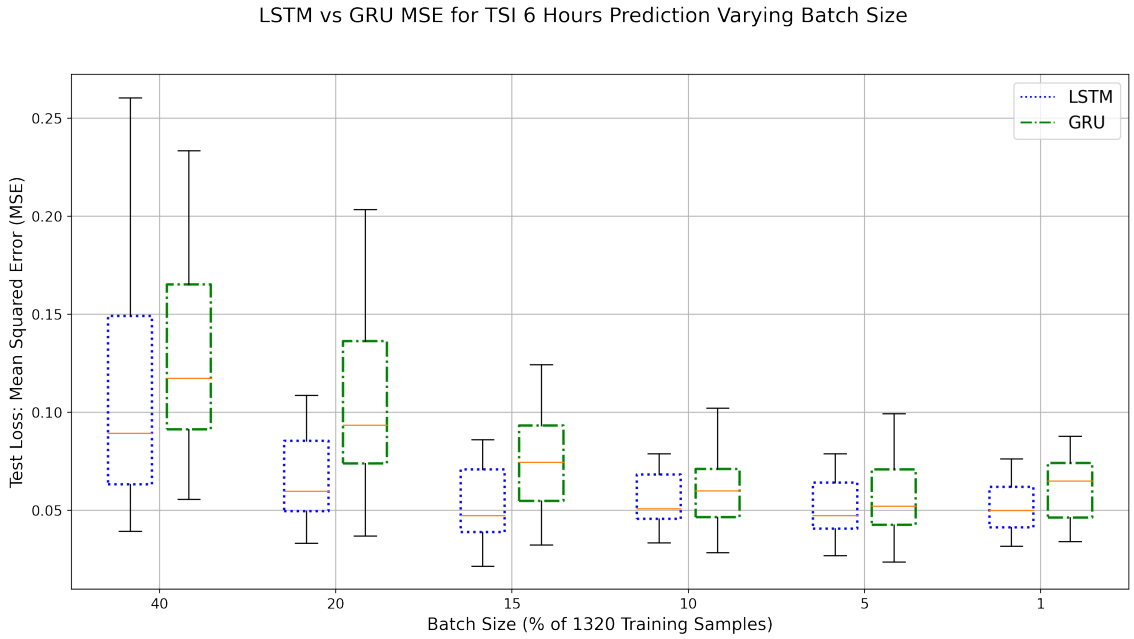
LSTM and GRU gave lower values for MSE, we compared these two architectures (Figure 6.12). Close to what was observed in the first experiment, the Simple RNN had worse performance comparing to the LSTM and GRU nets, which had similar performance at most of the cases. Looking generally for both gated nets, once again they did not have a significant difference in performance but once again the LSTM was slightly more stable and more performing than the GRU.

Figure 6.11 - Comparison between Simple RNN, LSTM and GRU RNN architectures, with batch size variation, for TSI 6 hours forecast.



Regarding the batch size variation, Figures 6.11 and 6.12 show lower MSE values, for the three nets, something beside 1% and 10%. Looking only at the gated nets, they also showed the same trend from Simple RNN. When we look only to the LSTM net, we see small errors also for batch rate of 15%.

Figure 6.12 - Comparison between LSTM and GRU RNN architectures, with batch size variation, for TSI 6 hours forecast.



To verify if batch size is tied to a percentage of the training set size or to a predefined value, regardless of the size of the training set, two new experiments were performed with LSTM only. In the first experiment, data sets of three different sizes were used for training, with pre-defined batch size values varying between 32 and 1024 samples. The result is shown in Figure 6.13, split in two boxplots: the left one with all batch sizes, and the right one only with those that gave the best results. The best results were observed in almost the same range of that observed on the first batch size experiment, confirming that a good value of batch size can be used as a percentage and between 1% and 10%, or more specifically between 1% and 5% of the training set. The second experiment was aimed at checking whether the first results could vary by assuming a larger number of training epochs, considering the possibility that different batch sizes could take more or less time to converge. The results, given this time by a GRU net, reinforced the result observed in the first experiment, as shown in Figure 6.14, in which on the left we have the three training set sizes and on the right we compare only the two bigger sets. As in the first results, the smaller have not shown significant differences between the different batch sizes, and the other two sets once again showed better results for batches from 1% to 5% of the training sets.

Another notable result observed was the better performance for larger training sets, which will also be adopted from now on.

Figure 6.13 - Pre defined values of batch size variation experiment in an LSTM net predicting TSI 6 hours ahead with three different sizes of training sets.

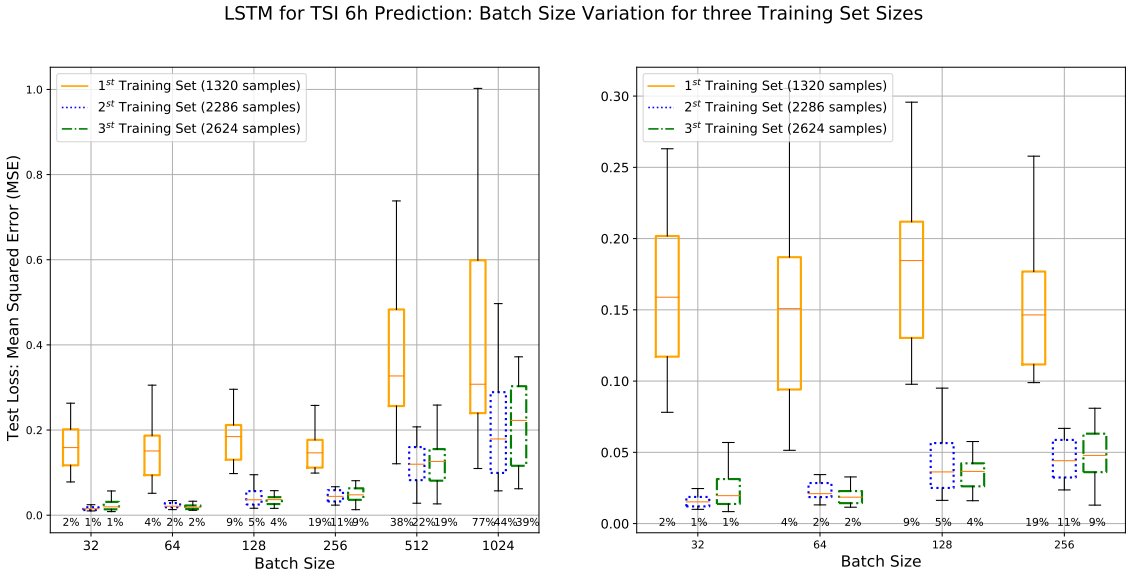
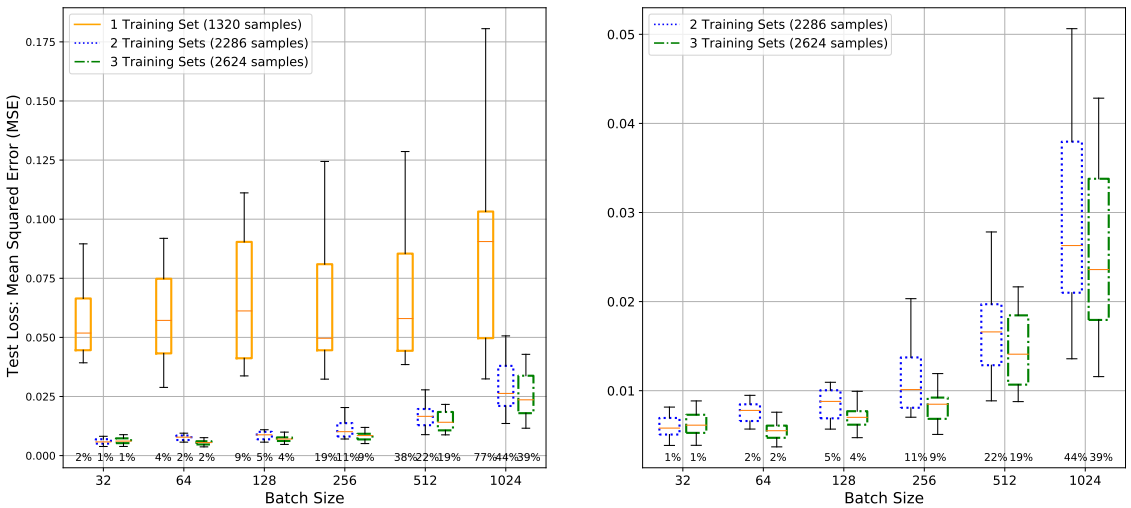


Figure 6.14 - Pre defined values of batch size variation experiment in a GRU net predicting TSI 6 hours ahead with three different sizes of training sets, with 200 training epochs.



For next experiments, the same batch rate chosen for TSI experiments will be used also for SSI experiments.

6.4 Data scaling

Before presenting input and output data to the network's input layer, we have the option of scaling them, assuring that they will be in a standardized or normalized range of values.

Figure 6.15 shows that the input data, which is the same for TSI and SSI predictions, is in a low-valued range, in the interval $[0, 0.03[$.

Figure 6.15 - Distribution of each of the RNNs' 40 input parameters. All the data periods (Figure 6.2 used in this work were merged for this plot.

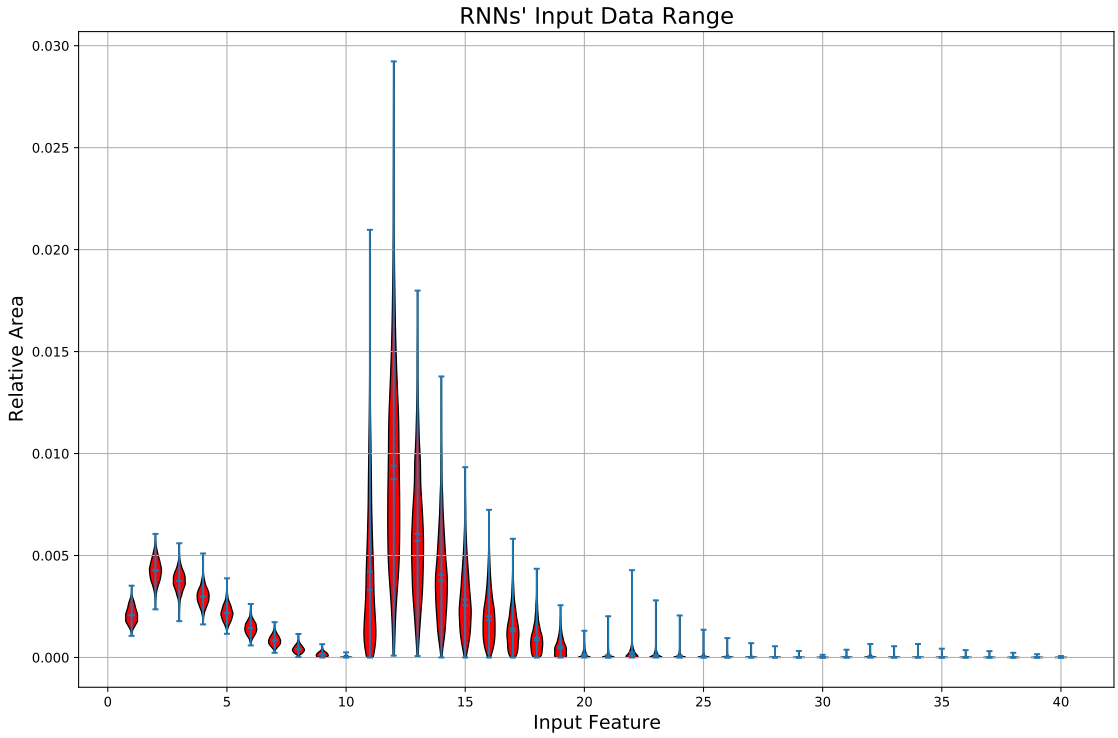
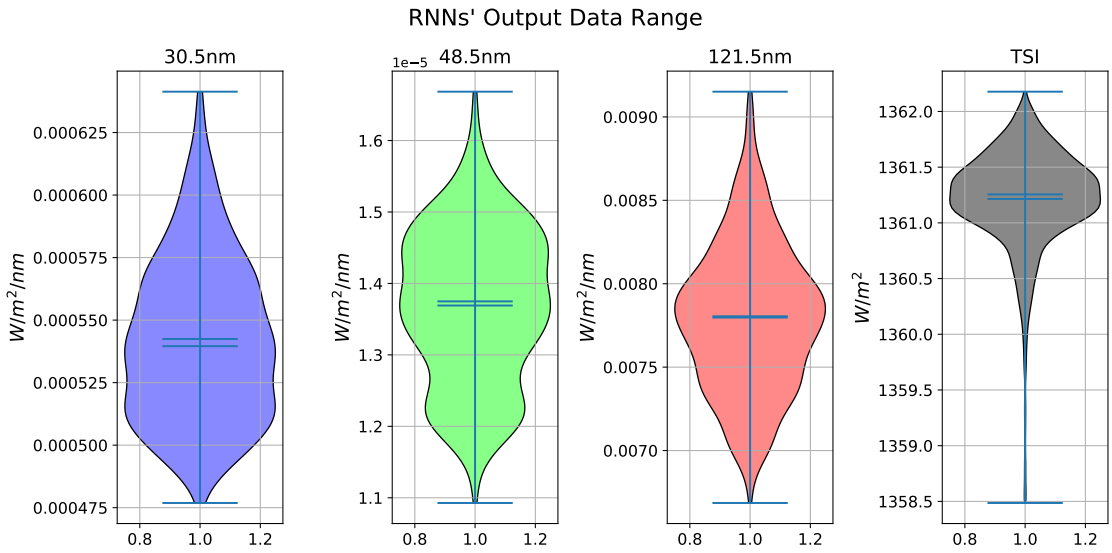


Figure 6.16 shows the range for TSI and each spectral emission line. Although they are in distinct ranges, all three emission lines show very low values, between $10^5 W/m^2/nm$ and $10^2 W/m^2/nm$, different from the TSI values pattern, which has high values, varying approximately from $1358.5 W/m^2$ to $1362 W/m^2$.

Figure 6.16 - RNNs' output data distribution.



In a scaling experiment for TSI, three tests were performed initially: without scaling either the input and output data, scaling only the output (because it is in a higher value range) and scaling both.

In the experiment without any scaling, the final accuracy obtained was very similar to the experiment with both data sets scaled, but required twelve times more epochs to it. In the tests where only the output set was scaled, even with a larger number of epochs, the results were not satisfactory, with estimation errors significantly larger than those obtained with the scaled sets.

The results of the experiment without scaling the data showed that scaling both input and output data would be the most appropriate for this problem. Therefore, the next step was to test the best scaling function between three options: standardization, normalization in range $[0, 1]$ and normalization in range $[-1, 1]$. An experiment was carried out with all possible combinations between the three scaling options and the input and output data, which resulted in nine different combinations. The performance for all of them can be seen on Figure 6.17, where we can see higher MSE for the two centered groups with standardized inputs and normalized output.

Figure 6.17 - TSI 6h MSE for all combinations of scaling functions applied for RNNs' input and output.

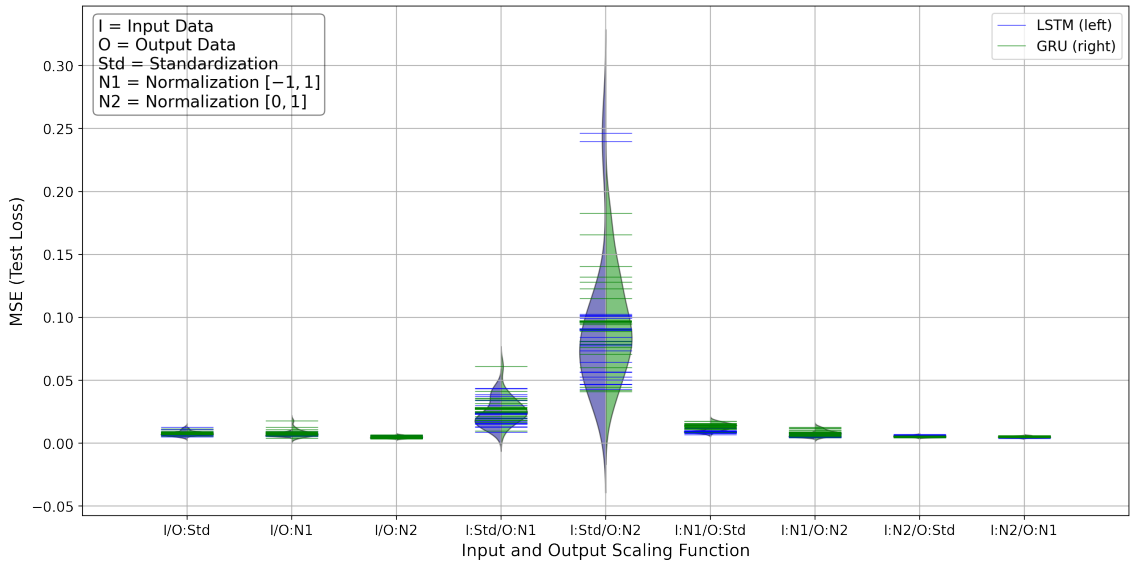


Figure 6.18 shows the same scaling experiment, this time excluding those two worse groups of results. In this figure we can see better the distribution of all the six combinations, from which the groups with the input normalized in the range $[0, 1]$ gave results with the same pattern, with error values less distributed and among the smallest comparing to the other ones.

An experiment was also performed to define the best scaling function to prepare the SSI values for the networks. Since the TSI experiment showed normalization in the range [0, 1] as the best function for the input data, in this experiment we will test only the best function for the SSI (output) data. Its result can be viewed in Figure 6.19.

For the spectral outputs, there was no better scaling function in common among them. Overall, none of them stood out positively or negatively. The $30.5nm$ line prediction almost kept the same performance for the three functions, showing a more spread out distribution in the tests for the range [0,1] normalization, with minimum and maximum values standing out from the other functions. The same behavior of normalization in range [0,1] was observed for the other emission lines, which seemed to present the lowest result for the Standardizing function. Although no results were outstanding, we decided to employ normalization [0,1] for the prediction of the $30.5nm$ line and standardization for the other emission lines.

Figure 6.18 - TSI 6h MSE for selected combinations of scaling functions applied for RNNs' input and output, with the two groups which standardized the input and normalized the output excluded.

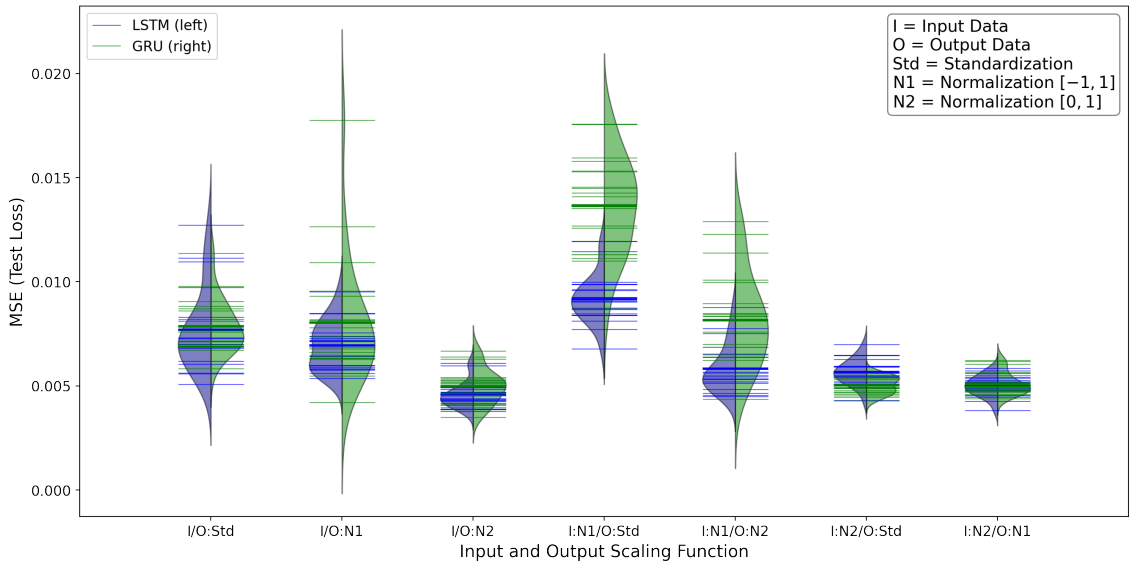
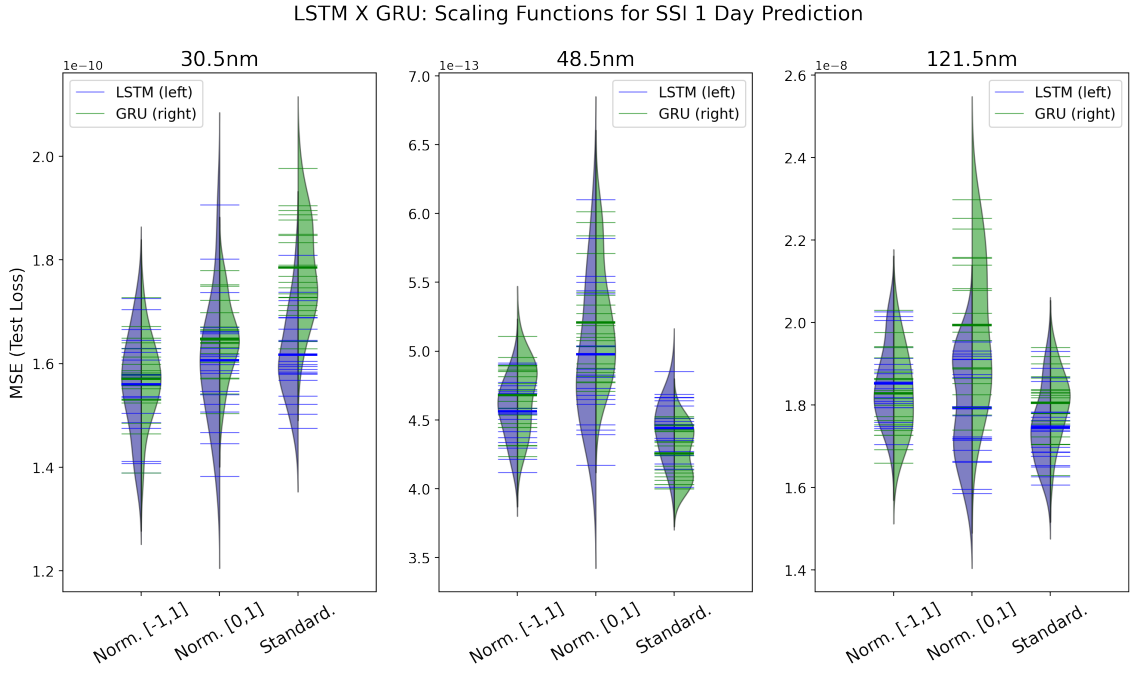


Figure 6.19 - Scaling experiment for SSI emission lines output.



6.4.1 Increasing the prediction intervals

A closing experiment with the three architectures was conducted in order to verify how Simple RNN performs for longer interval forecasts. To this purpose, the TSI was predicted for intervals from six hours to three days (72 hours), and the three emission lines of SSI were predicted for intervals from one to three days. The TSI results, in Figure 6.20, clearly confirms the predominance of LSTM and GRU nets performance over Simple RNN. The SSI results, in Figure 6.21 also shows lower MSE values for the gated RNN in most of the boxes. Both gated nets still presented very close performances in both experiments. The performance results obtained in these experiments with different prediction intervals, despite already indicating the best networks, can still be improved after adjustments in some hyper-parameters.

Figure 6.20 - The three RNN architectures performance for TSI prediction for six to 72 hours.

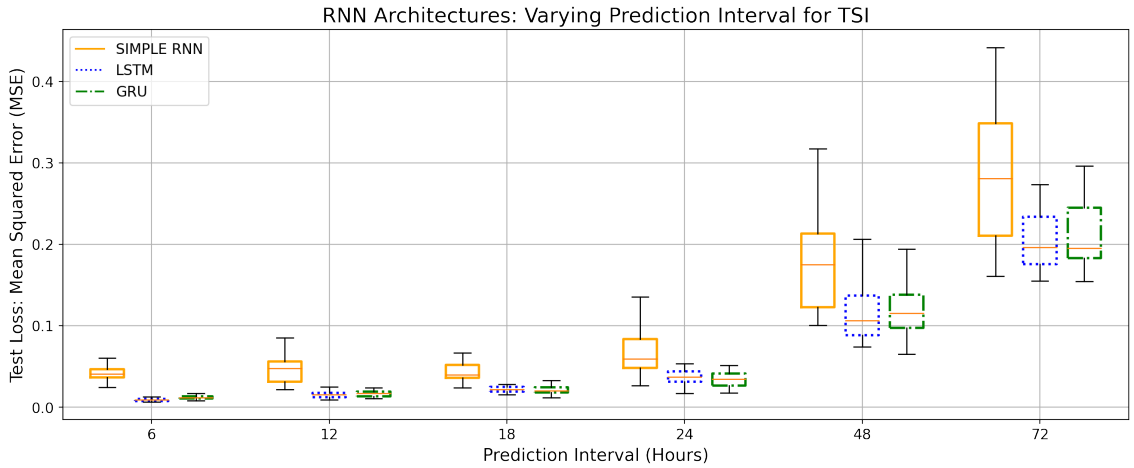


Figure 6.22 is a part of the results already presented for TSI on Figure 6.20, but this time focusing on the gated nets. In a similar way of TSI experiments, Figure 6.23 is a part of SSI experiments showed on Figure 6.21, this time directed to compare only the gated architectures. Even though the forecasts loss of the three emission lines were reached different scales, it can be seen that, as with the TSI experiment, the SSI experiment also shows that the performance of the networks decreases as the forecast interval increases, as would be expected of a forecasting system.

Figure 6.21 - The three RNN architectures performance for some of spectral lines prediction for one to three days.

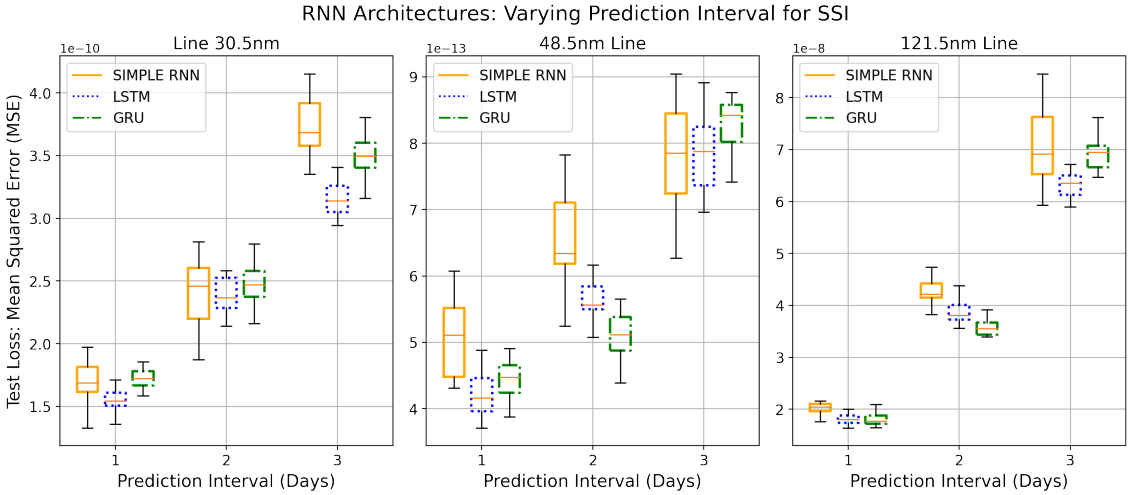


Figure 6.22 - LSTM and GRU performance comparison: TSI prediction for 6 to 72 hours.

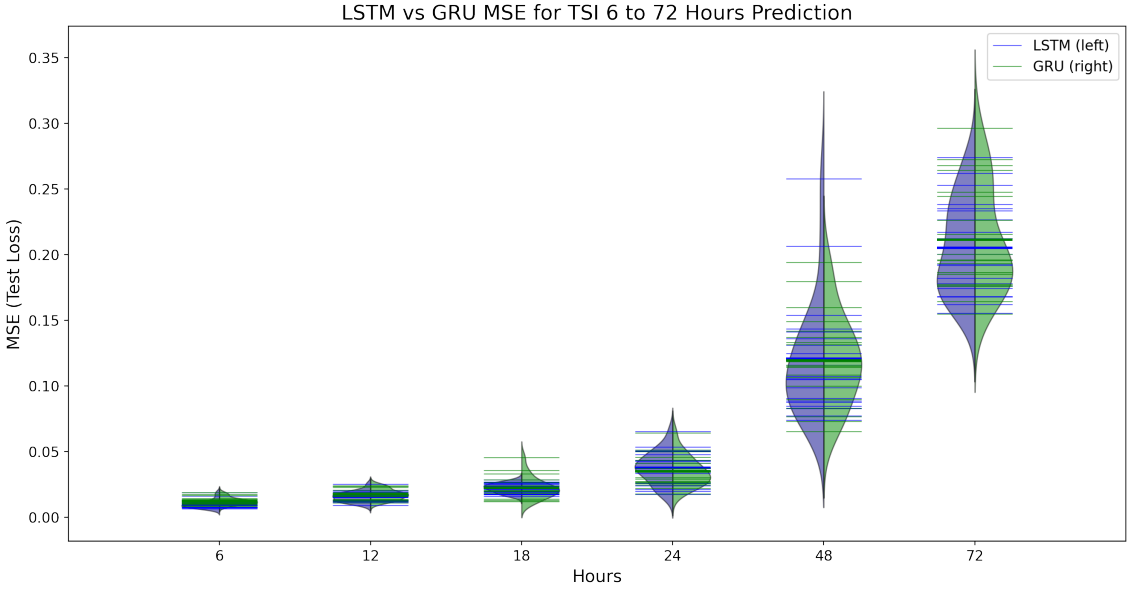
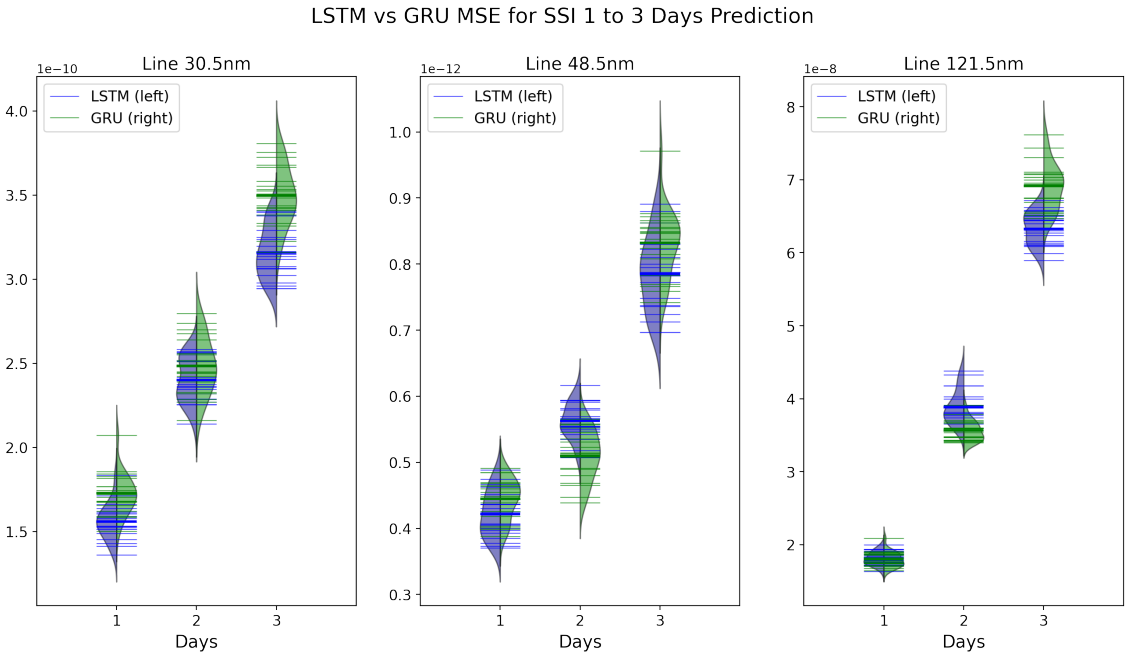


Figure 6.23 - LSTM and GRU performance comparison: SSI three emission lines prediction for 1 to 3 days.



Those results confirm that the two architectures do not differ significantly in their performance, maybe we can say that LSTM brings slightly better results than GRU in most of the results. Noting definitely the better performance of the gated architectures, for the next experiments we discarded Simple RNN and chose at least one of the gated nets to perform the predictions.

For the two irradiance types, the experiments consisted of training the two gated networks for each time interval and testing the trained network with at least one test set.

6.5 TSI predictions

Based on the hyper-parameters variation results reported in the previous section, a new hyper-parameter list was defined for the experiments that came next, which specifically measured error and accuracy of the predictions for each time interval. Table 6.2 will show only the parameters that were changed to TSI predictions, the others were remained with the values set in Table 6.1.

Table 6.2 - TSI experiments default hyper-parameters.

Settings/Hyper-Parameters	Value
Scaling method for Input Data	Normalization [0,1]
Scaling method for Output Data	Normalization [0,1]
Hidden Units number	9
Hidden Layer Activation	tanh
Batch size (% of the training samples)	1
Epochs	50

The next figures refer to the TSI six hours prediction performance: Figure 6.24 shows LSTM performance and Figure 6.25 presents GRU performance. Both have, on the left, the training and validation curves obtaining convergence; and, on the right, the training and validation predicted sets confronted in a scatter plot. They show, respectively, LSTM and GRU nets' convergence in the first training epochs, and the proximity between the real and predicted values in both training (with data known to the net) and validation (with data not used in the training supervision) phases.

Figure 6.24 - TSI 6 hours forecast training and validation performance with LSTM.

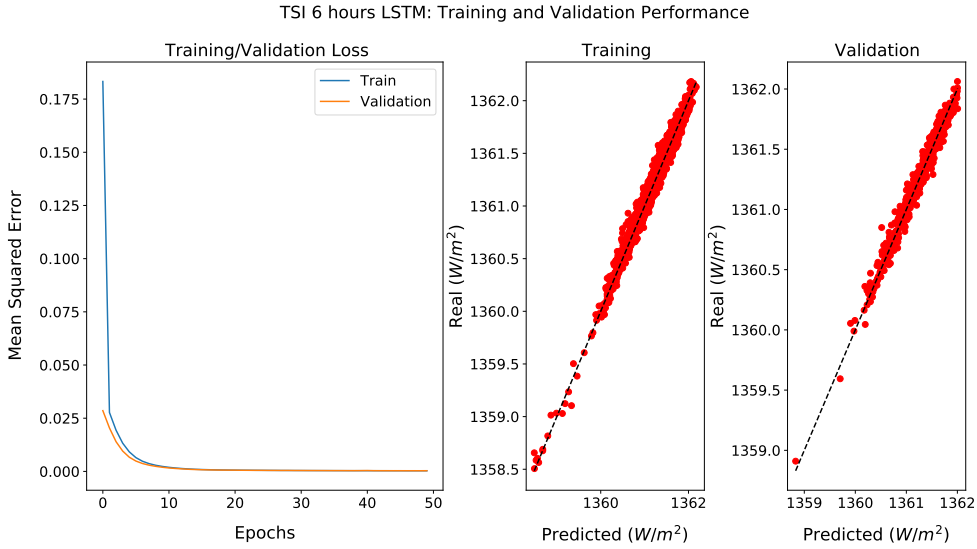
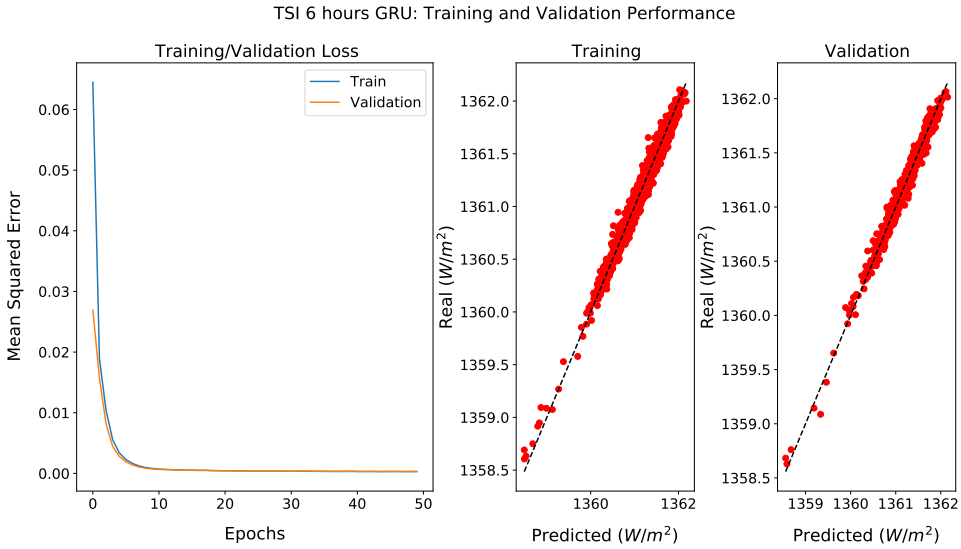


Figure 6.25 - TSI 6 hours forecast training and validation performance with GRU.



Figures 6.26 and 6.27 show two plots of LSTM for two different test data sets predictions, and Figure 6.28 and 6.29 shows the same data sets predicted by GRU.

Figure 6.26 - TSI 6 hours forecast test performance with LSTM net. Data set of 105 days.

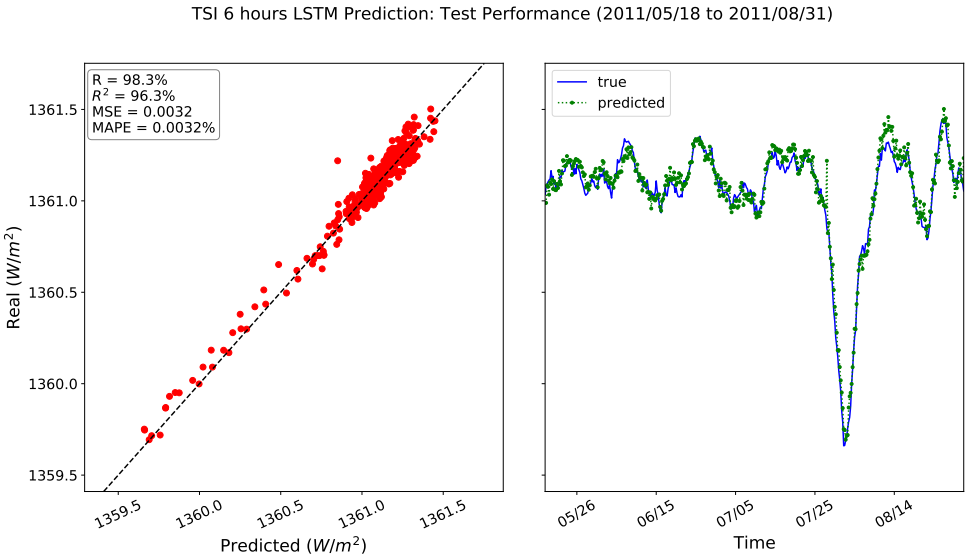


Figure 6.27 - TSI 6 hours forecast test performance with LSTM net. Data set of 28 days.

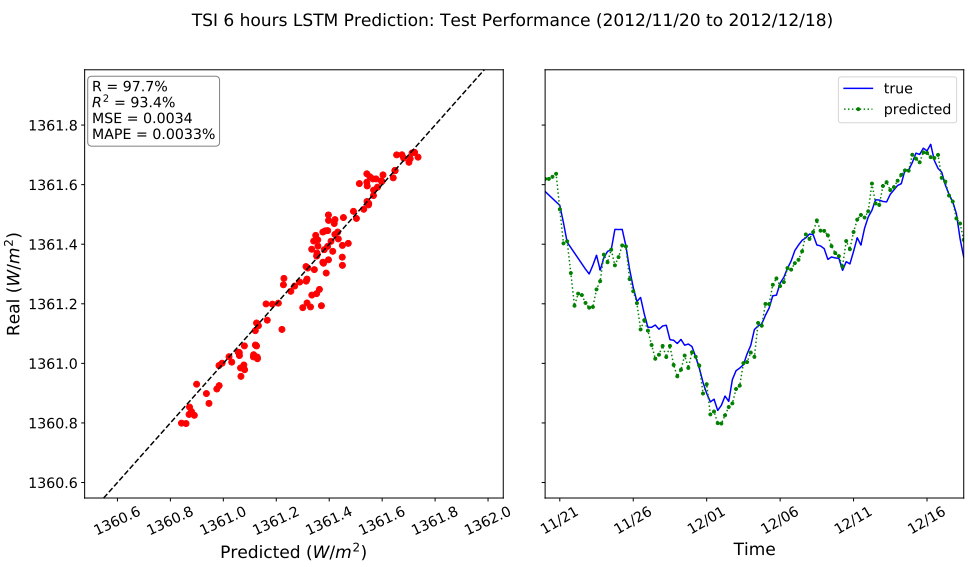


Figure 6.28 - TSI 6 hours forecast test performance with GRU net. Data set of 105 days.

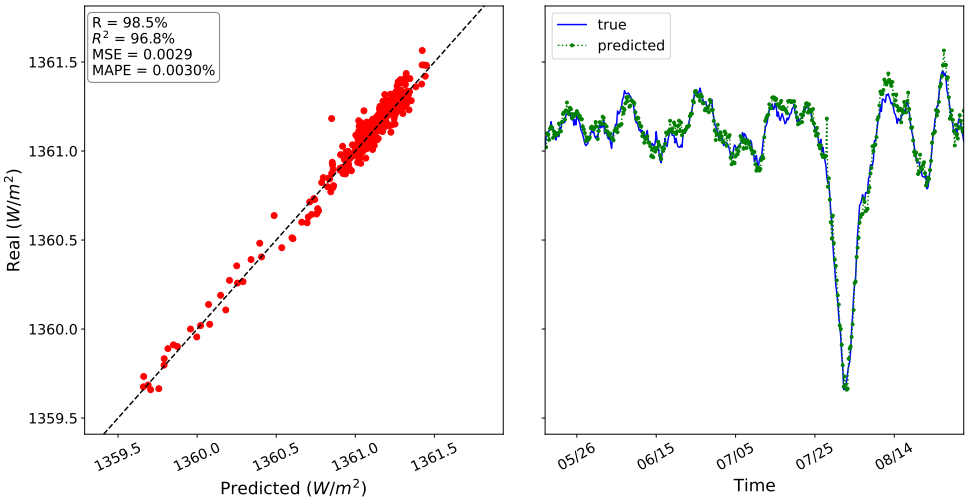
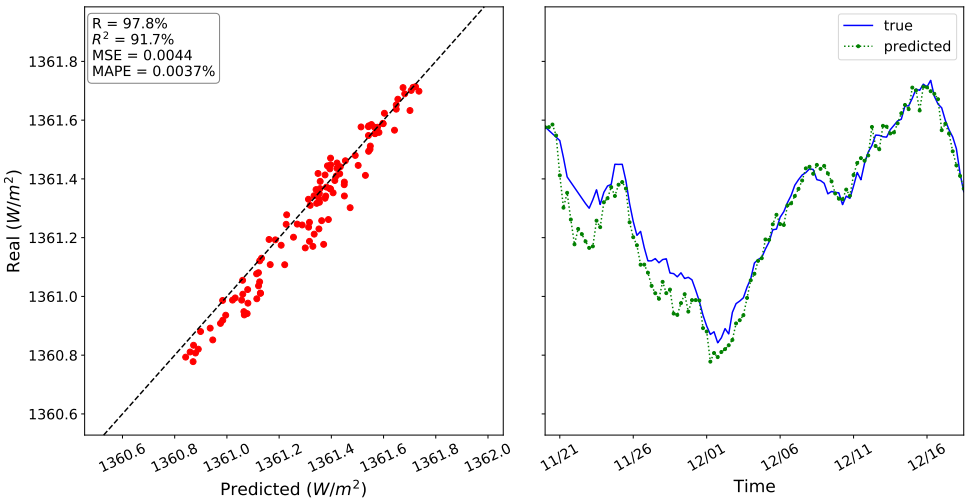


Figure 6.29 - TSI 6 hours forecast test performance with GRU net. Data set of 28 days.



The two predicted test sets have two differences between them: the data period and the series length. While the first is 105 days long, the second is only 28 days long. This brings also a difference in the amplitudes between both series. This reflects in the small discrepancy also present in the error and accuracy measurements of the two results. Even considering these differences, both the error and accuracy measurements, as well as the comparative graphs between the actual and predicted values, show good prediction quality for the two data series.

The results obtained by GRU are very close to the results obtained by LSTM, and it is not possible to choose one of the two just based on the results presented. Perhaps for the choice of one of the architectures, more sets of tests, of different amplitudes should be foreseen to verify, on average, which architecture presents better results.

Considering the close results obtained so far, and in order also not to exceed the number of figures, for the following time intervals, only the LSTM training and validation plots and its results for one of the test sets will be shown. The GRU result and the prediction for the second test set of both nets will be available in the APPENDIX A - A.2.

Figures 6.30, 6.32 and 6.34 show the LSTM training and validation performance for twelve, eighteen and 24 hours prediction, respectively. And Figures 6.31, 6.33 and 6.35 show the LSTM test performance for the same intervals, respectively.

Figure 6.30 - TSI 12 hours forecast training and validation performance with LSTM.

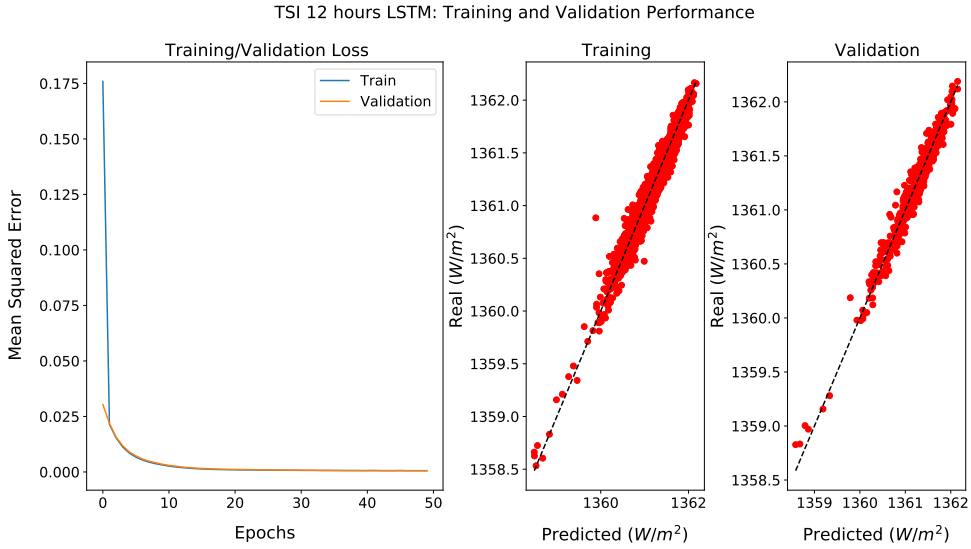


Figure 6.31 - TSI 12 hours forecast test performance with LSTM net.

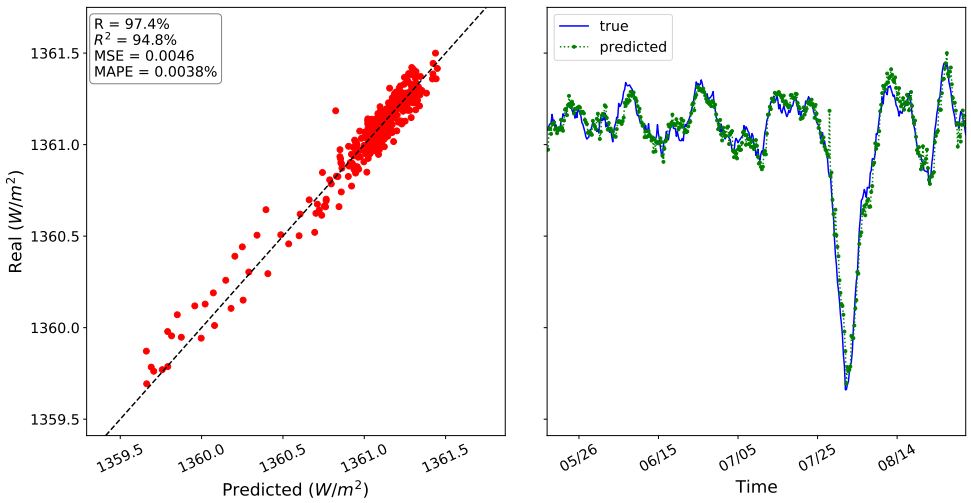


Figure 6.32 - TSI 18 hours forecast training and validation performance with LSTM.

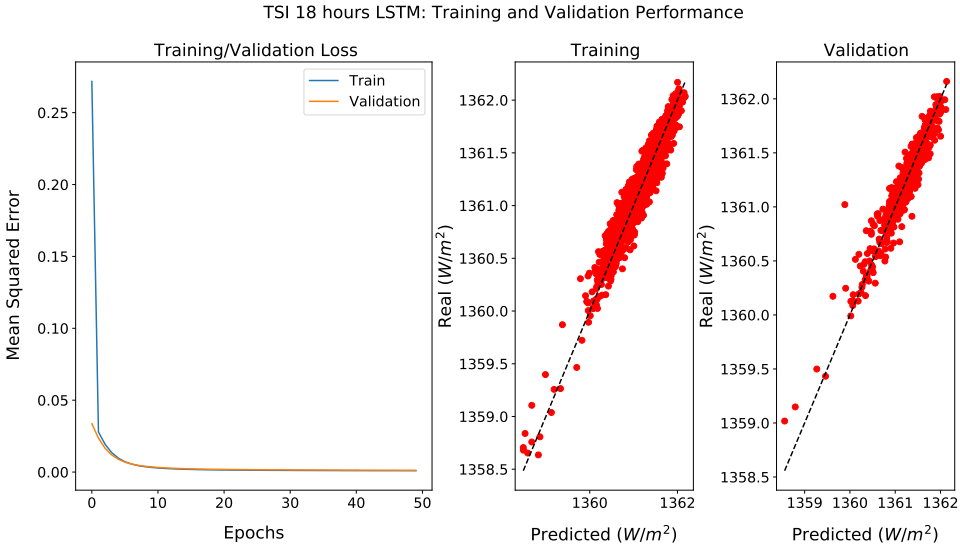


Figure 6.33 - TSI 18 hours forecast test performance with LSTM net.

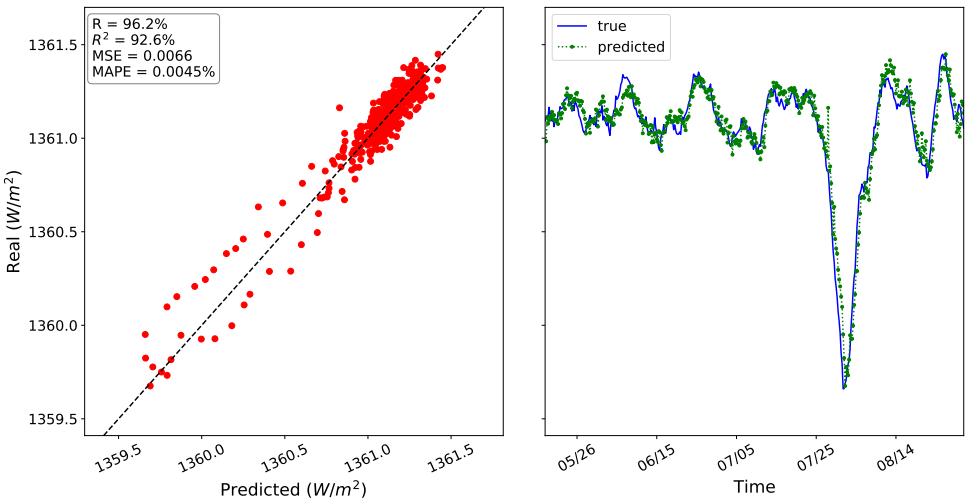


Figure 6.34 - TSI 24 hours forecast training and validation performance with LSTM.

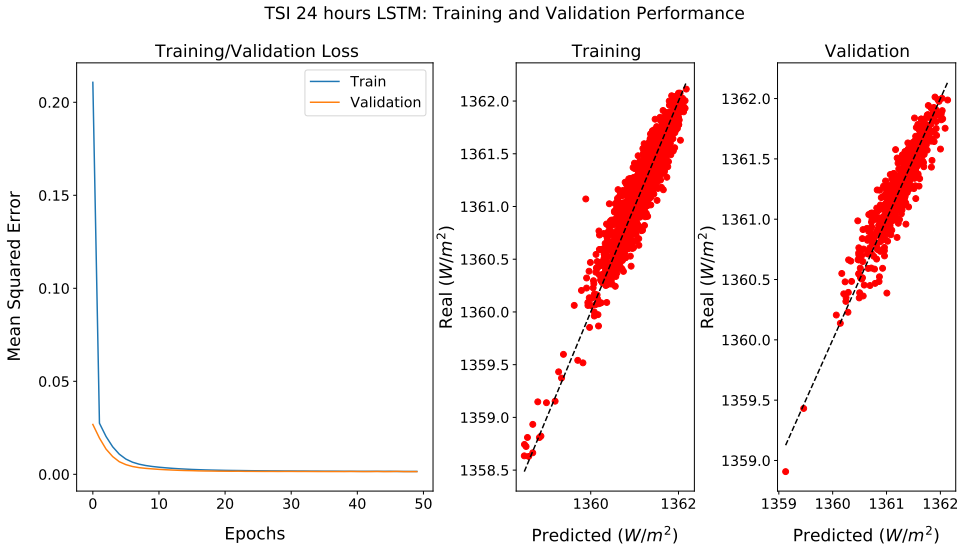
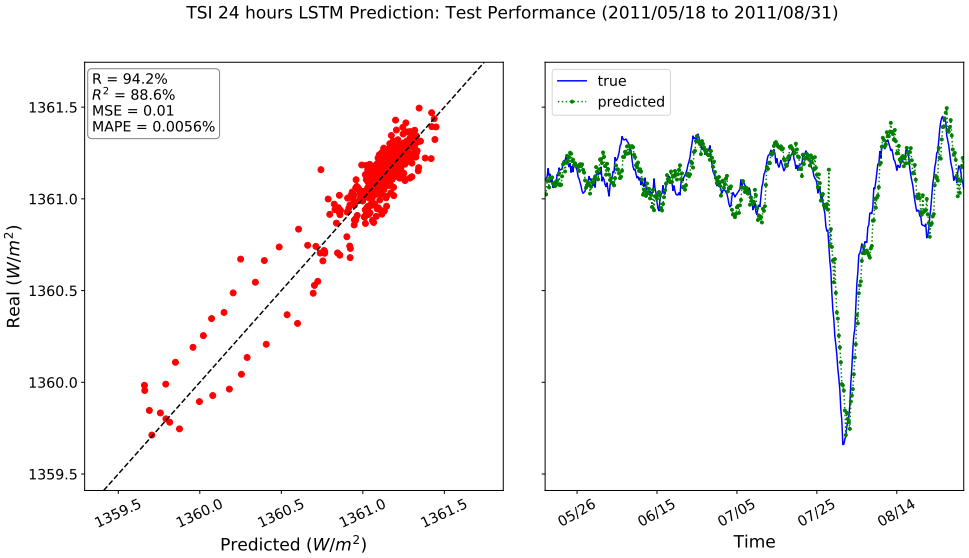


Figure 6.35 - TSI 24 hours forecast test performance with LSTM net.



Figures 6.36 and 6.38 show the LSTM training and validation performance, respectively, for two and three days prediction. And Figures 6.37 and 6.39 show the LSTM test performance for two and three days prediction, respectively.

Figure 6.36 - TSI 48 hours forecast training and validation performance with LSTM.

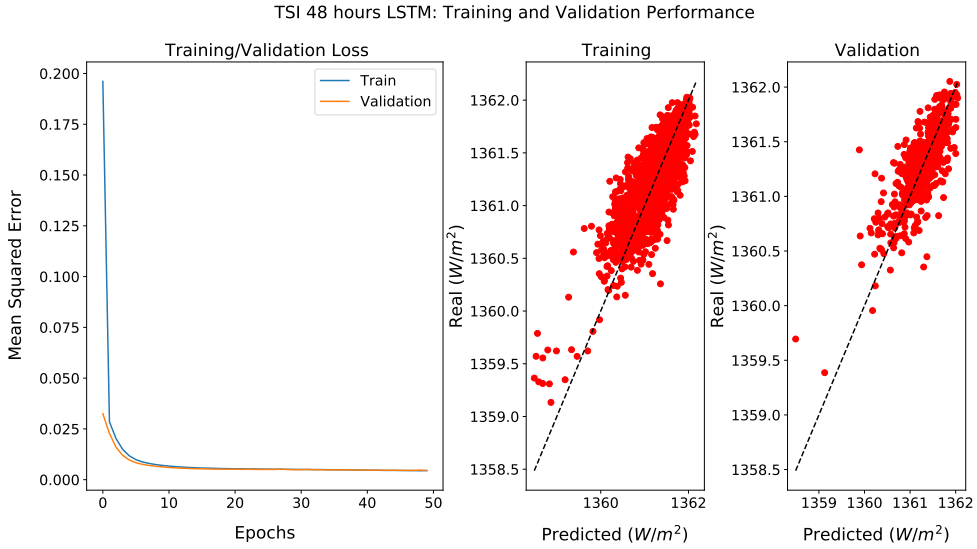


Figure 6.37 - TSI 48 hours forecast test performance with LSTM net.

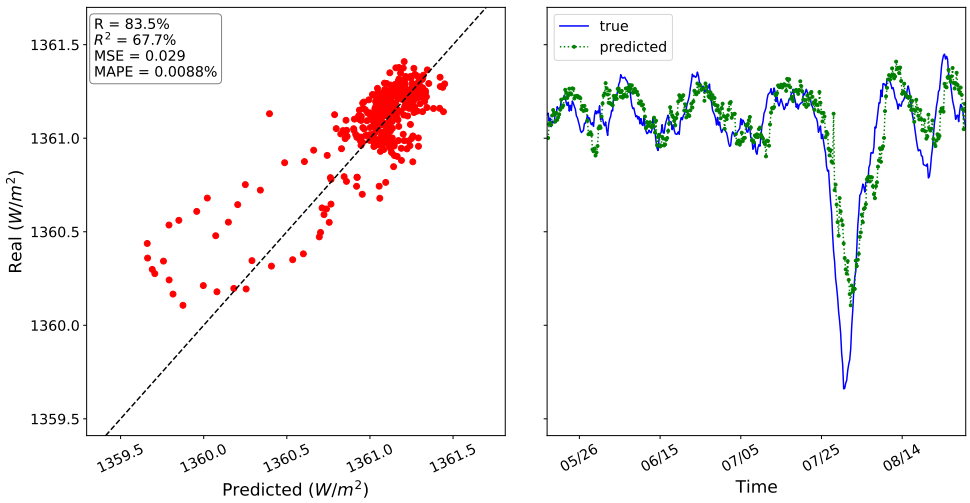


Figure 6.38 - TSI 72 hours forecast training and validation performance with LSTM.

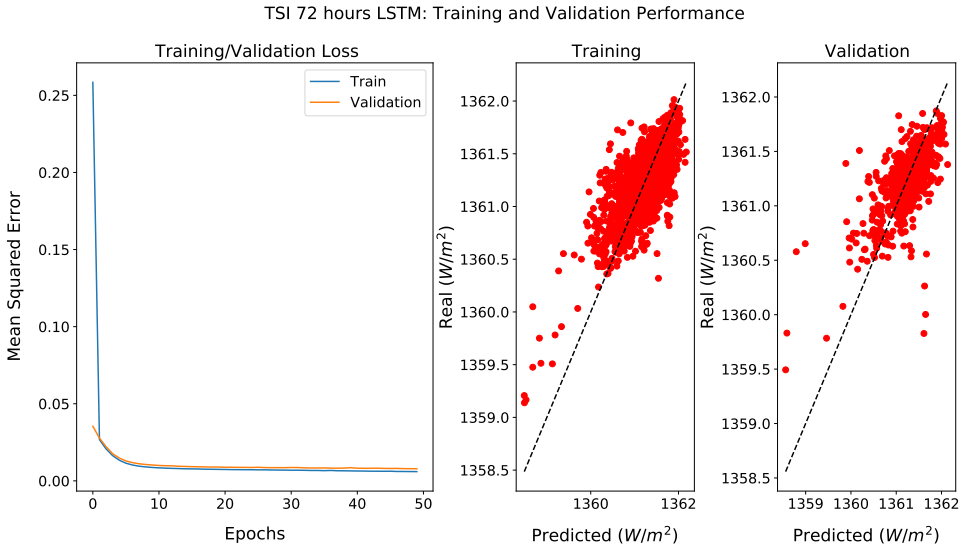
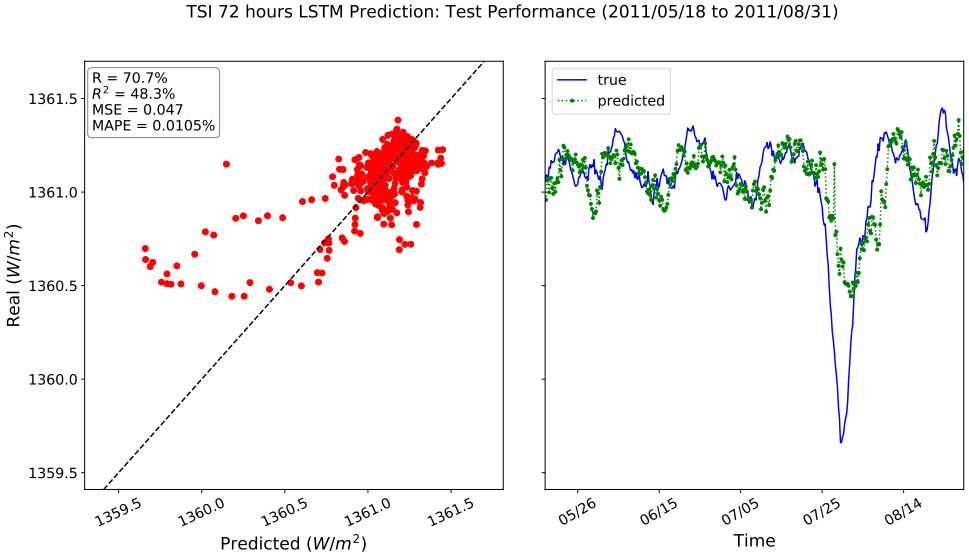


Figure 6.39 - TSI 72 hours forecast test performance with LSTM net.



For forecasts up to 24 hours, even if it can be seen that there is a slight decrease in the quality of the forecast, the results still seem satisfactory, considering the desired accuracy for a real-time forecast system, for example. However, for 48 hours and 72 hours predictions, there is a considerable drop in the quality of the forecast, with the predicted data curve not being able to follow the tendency in the case of higher TSI values, or with the real lower values not being reached by the predicted values.

This difference in response pattern between TSI forecasts of up to one day and those of two and three days leads us to believe that the input data is not as relevant or sufficient for forecasts with intervals of two days or more. Tests for shorter time intervals ranging from one to two days could be performed to verify at what point the response pattern starts to change more significantly.

For long term predictions, other input parameters that can complement the magnetograms and the continuous images could be considered to improve the quality of the prediction.

6.6 SSI predictions

For the SSI prediction tests, in addition to the systematic experiments performed in search of the best hyper-parameters for the networks, some non-systematic experiments looked for the best values for the other parameters. For example, for the best number of epochs, random trainings were performed observing the training curves, and it was observed that for the three lines, after 50 epochs, the error did not decrease any further. Another example is about the hidden units number: the best hidden units number offered by the $lyman - \alpha$ experiment was repeated initially for the other lines as well, and such value turned out to be approximately the best range for the other lines as well.

Table 6.3 shows the hyper-parameters that have brought the best training and testing results so far, and those that were adopted on the next results.

Table 6.3 - SSI experiments default hyper-parameters.

Hyper-Parameters	$30.5nm$	$48.5nm$	$121.5nm$
Hidden Units number	4	5	5
Epochs	50	100	50
Output Scaling	$[0, 1]$	$[0, 1]$	$[0, 1]$
Input Activation	$tanh$	$tanh$	$tanh$
Batch Size	1%	5%	1%

Like the TSI experiments, the SSI experiments also showed very similar results for LSTM and GRU. So we will list in this section only the results obtained by LSTM, and those with GRU will be placed in the APPENDIX A - A.3.

6.6.1 Emission line $30.5nm$ (helium)

Figures 6.40, 6.42 and 6.44 present the training/validation performance for emission line $30.5nm$ obtained with LSTM for, respectively, one, two and three days. Figures 6.41, 6.43 e 6.45 present the test performance for emission line $30.5nm$, obtained with LSTM, for, respectively, one, two and three days forecast.

Figure 6.40 - Emission line $30.5nm$ 1 day forecast training and validation performance with LSTM.

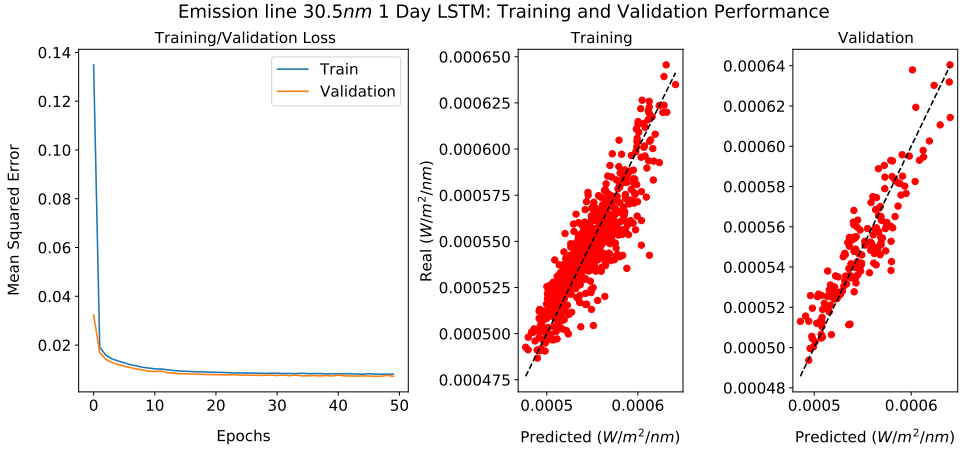


Figure 6.41 - Emission line $30.5nm$ 1 day forecast test performance with LSTM net.

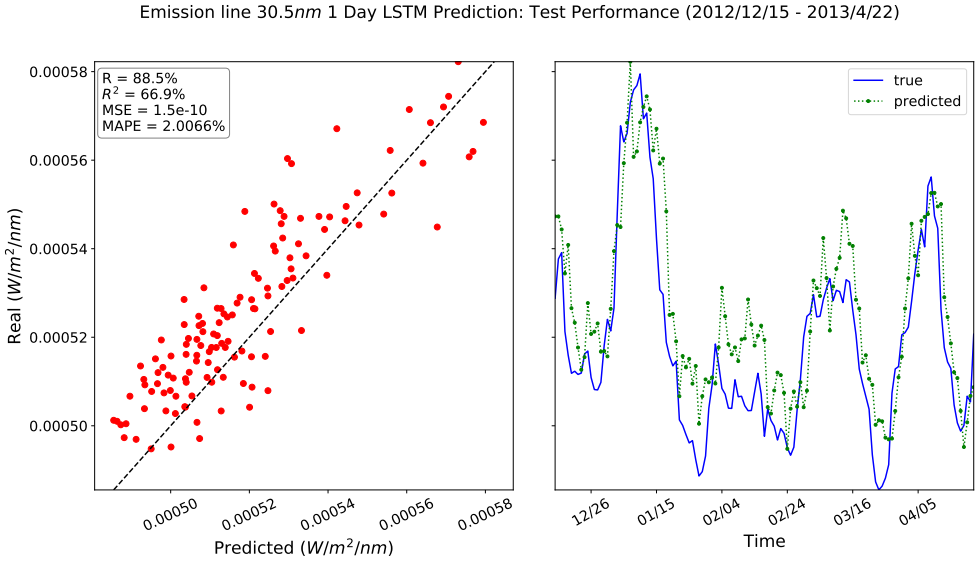


Figure 6.42 - Emission line 30.5nm 2 days forecast training and validation performance with LSTM.

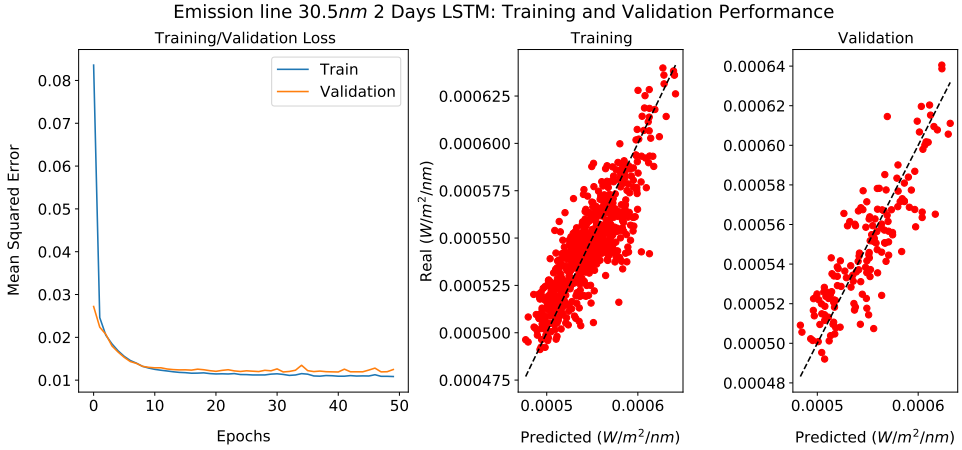


Figure 6.43 - Emission line 30.5nm 2 days forecast test performance with LSTM net.

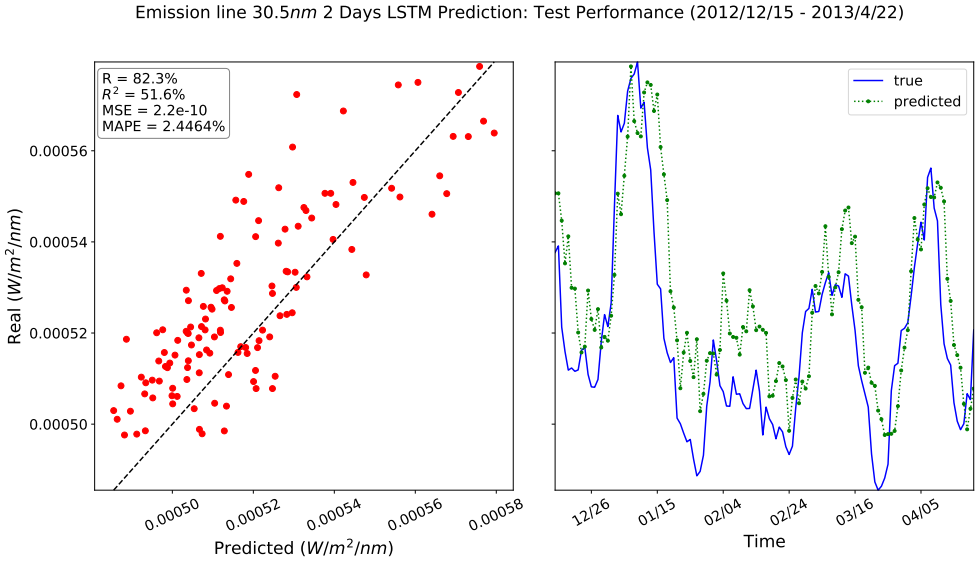


Figure 6.44 - Emission line 30.5nm 3 days forecast training and validation performance with LSTM.

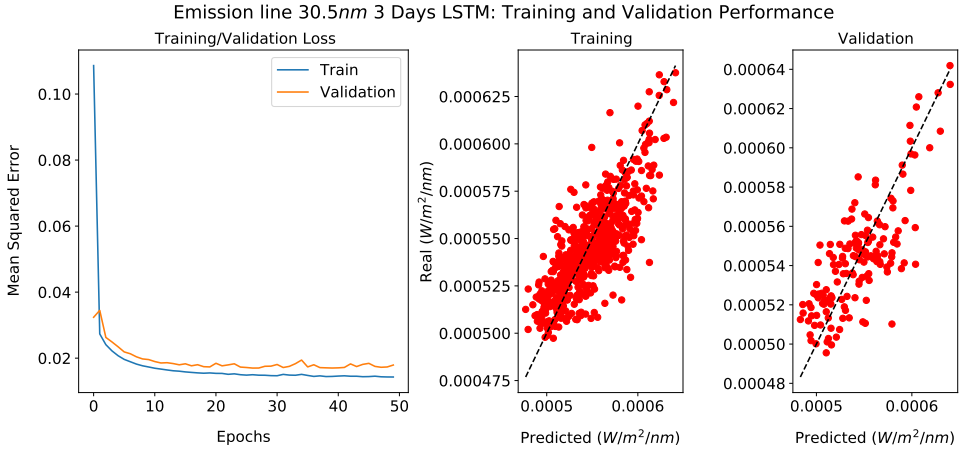
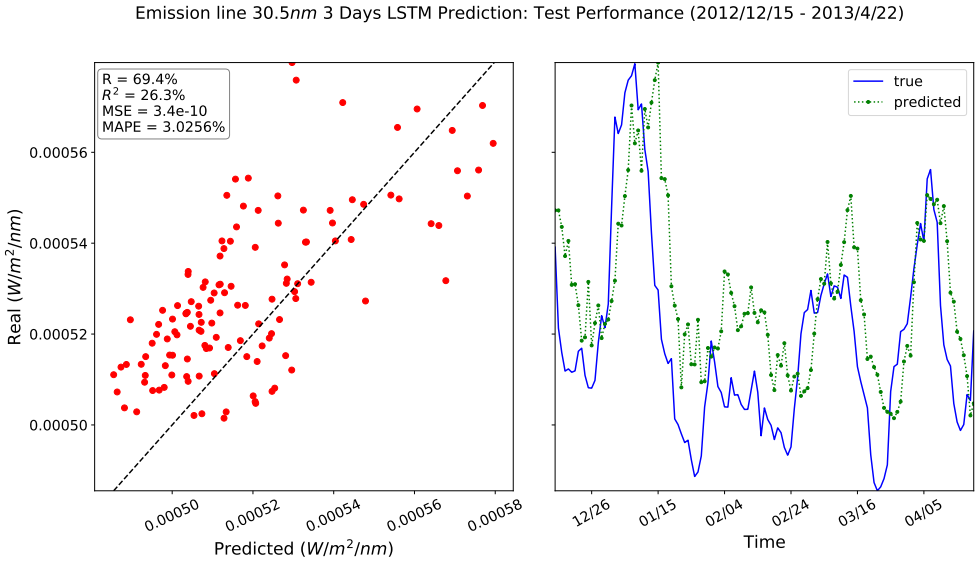


Figure 6.45 - Emission line 30.5nm 3 days forecast test performance with LSTM net.



The one day prediction for the 30.5nm helium emission line showed significantly lower quality than that obtained for the up to one-day TSI predictions. The convergence obtained in the training suggests that even if we try to tune the network, the results are not likely to improve considerably. It was already expected the worsening in the results when increasing the forecast interval, but in general, for the three time intervals, the same response pattern is observed in the forecast. This leads one to believe that the improvement in the one-hour forecast would bring, proportionally, the same level of improvement for forecasts of the other time instants.

6.6.2 Emission line $48.5nm$ (helium)

Figures 6.46, 6.48 and 6.50 present the training/validation performance for emission line $48.5nm$, obtained with LSTM, for, respectively, one, two and three days forecast. And Figures 6.47, 6.49 e 6.51 present the test performance for emission line $48.5nm$, obtained with LSTM, for, respectively, one, two and three days forecast.

Figure 6.46 - Emission line $48.5nm$ 1 day forecast training and validation performance with LSTM.

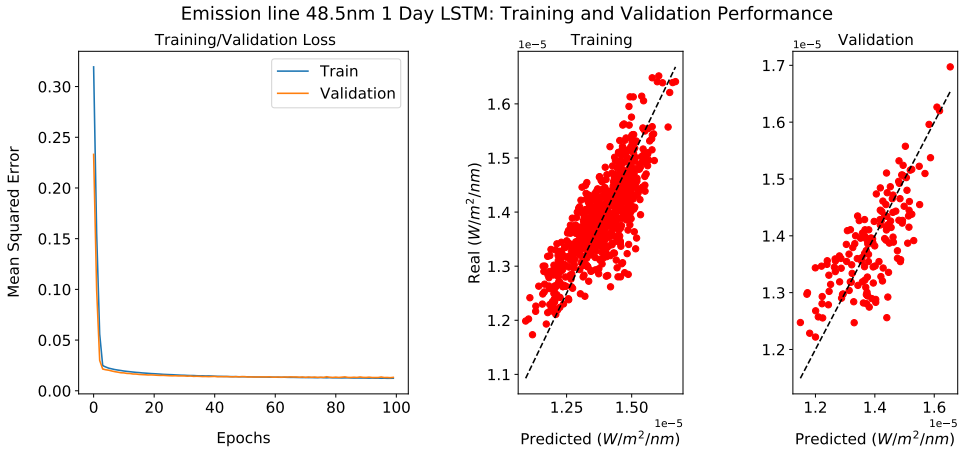


Figure 6.47 - Emission line $48.5nm$ 1 day forecast test performance with LSTM net.

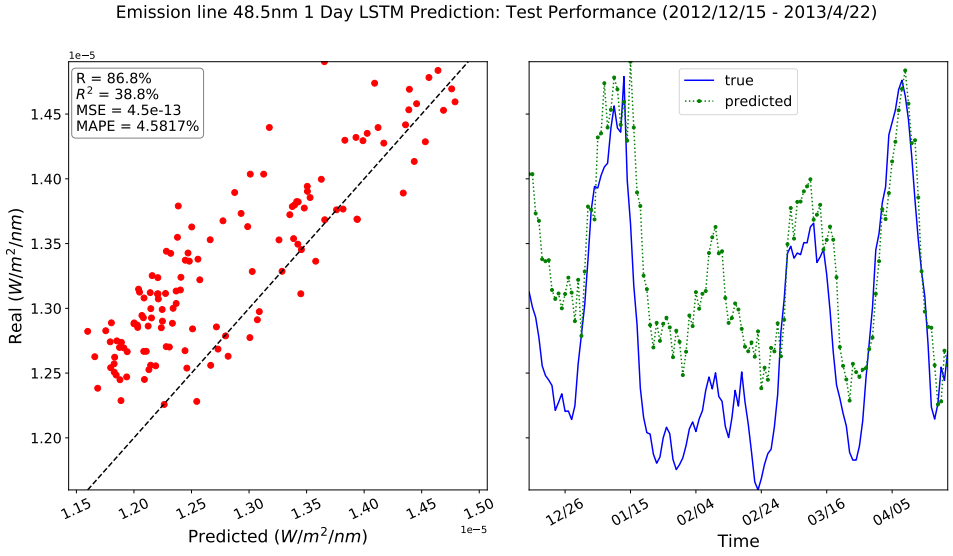


Figure 6.48 - Emission line 48.5nm 2 days forecast training and validation performance with LSTM.

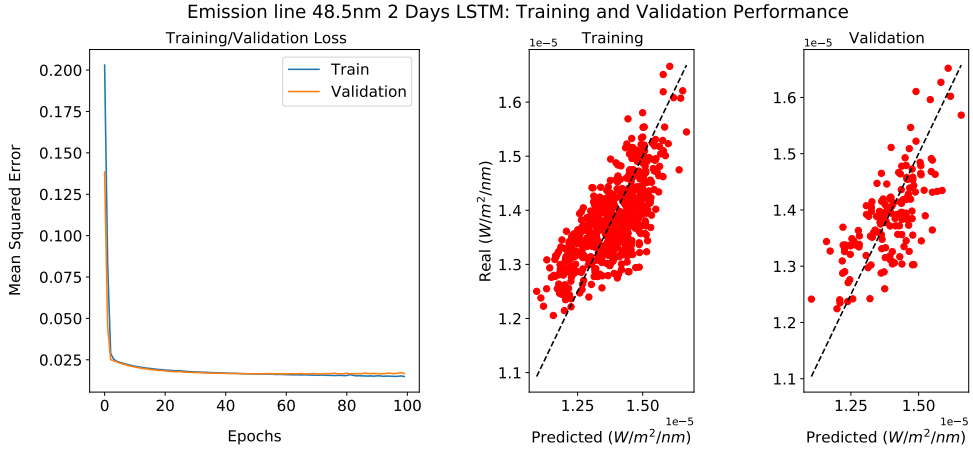


Figure 6.49 - Emission line 48.5nm 2 days forecast test performance with LSTM net.

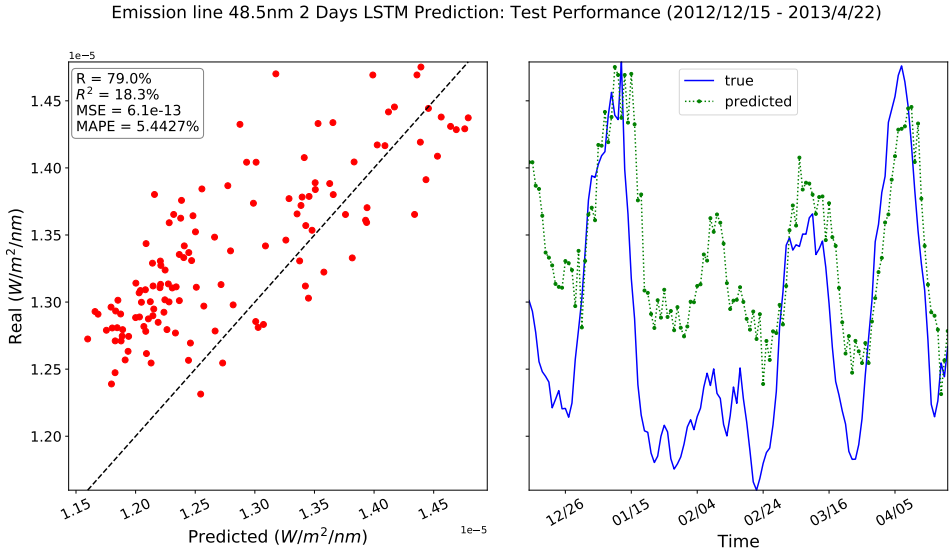


Figure 6.50 - Emission line 48.5nm 3 days forecast training and validation performance with LSTM.

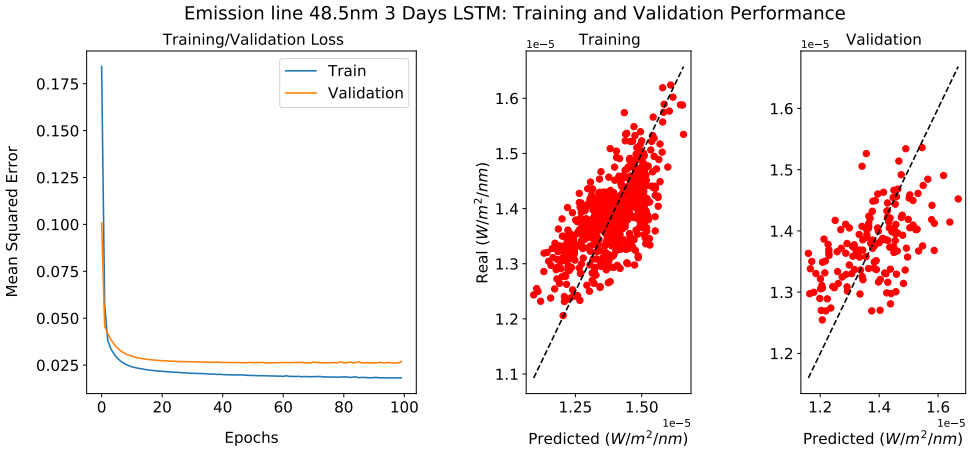
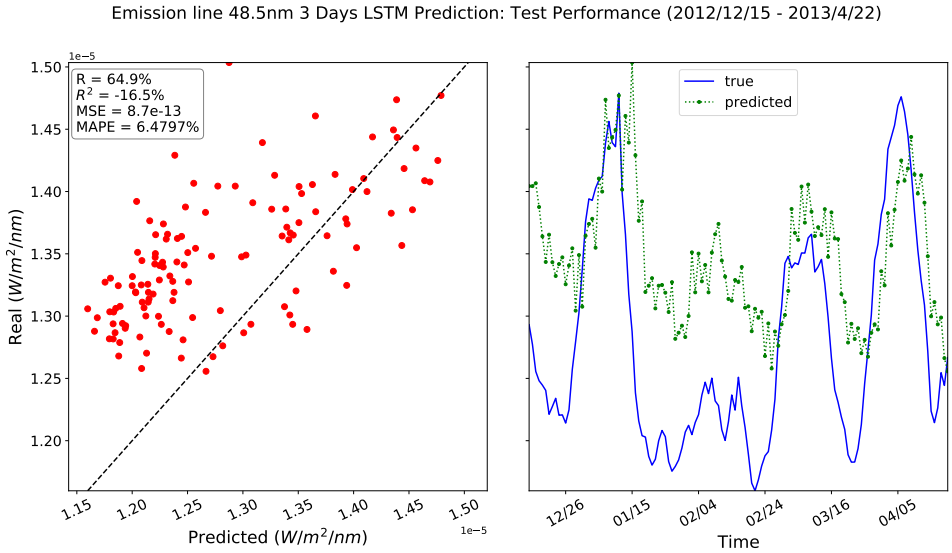


Figure 6.51 - Emission line 48.5nm 3 days forecast test performance with LSTM net.



6.6.3 Emission line 121.5nm or $lyman - \alpha$ (hydrogen)

Figures 6.52, 6.54 and 6.56 present the training/validation performance for emission line 121.5nm, obtained with LSTM, for, respectively, one, two and three days forecast. And Figures 6.53, 6.55 e 6.57 present the test performance for emission line 121.5nm, obtained with LSTM, for, respectively, one, two and three days forecast.

Figure 6.52 - Emission line 121.5nm 1 day forecast training and validation performance with LSTM.

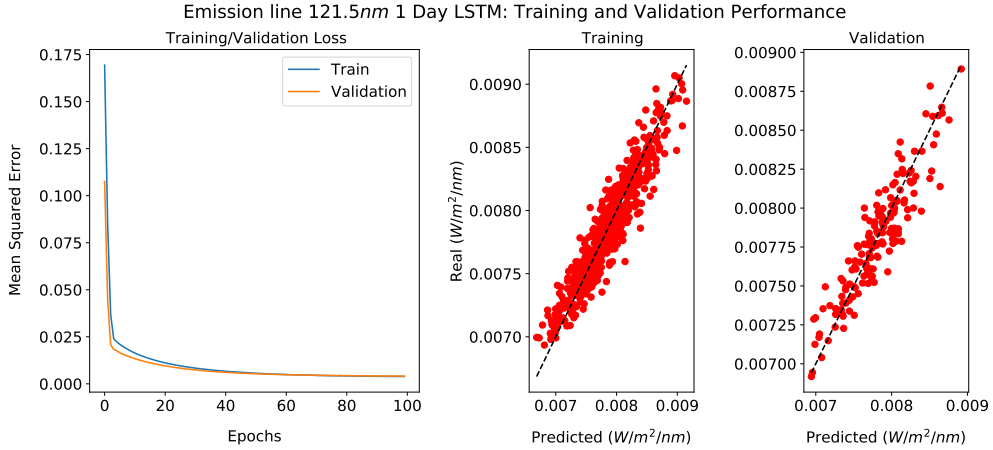


Figure 6.53 - Emission line 121.5nm 1 day forecast test performance with LSTM net.

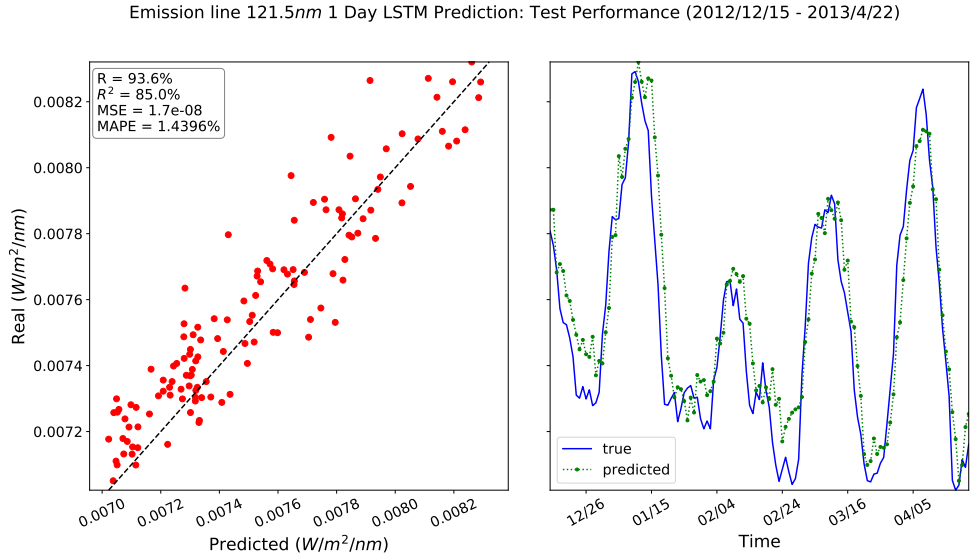


Figure 6.54 - Emission line 121.5nm 2 days forecast training and validation performance with LSTM.

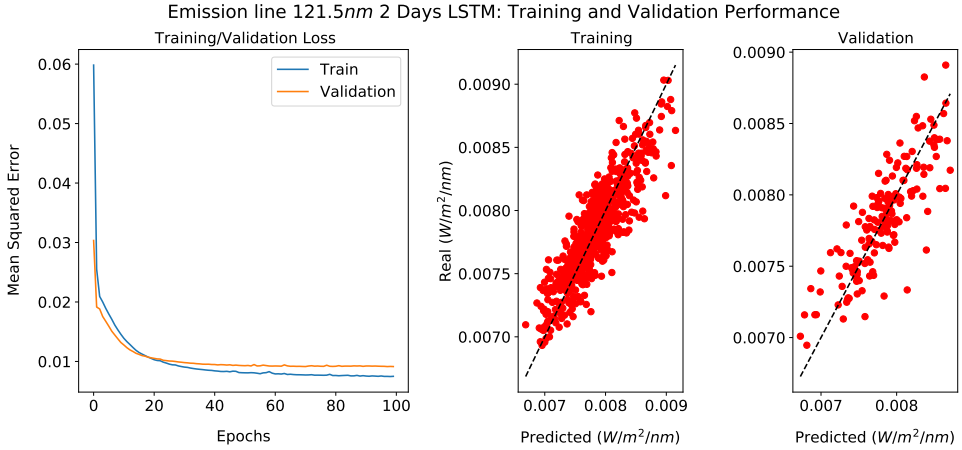


Figure 6.55 - Emission line 121.5nm 2 days forecast test performance with LSTM net.

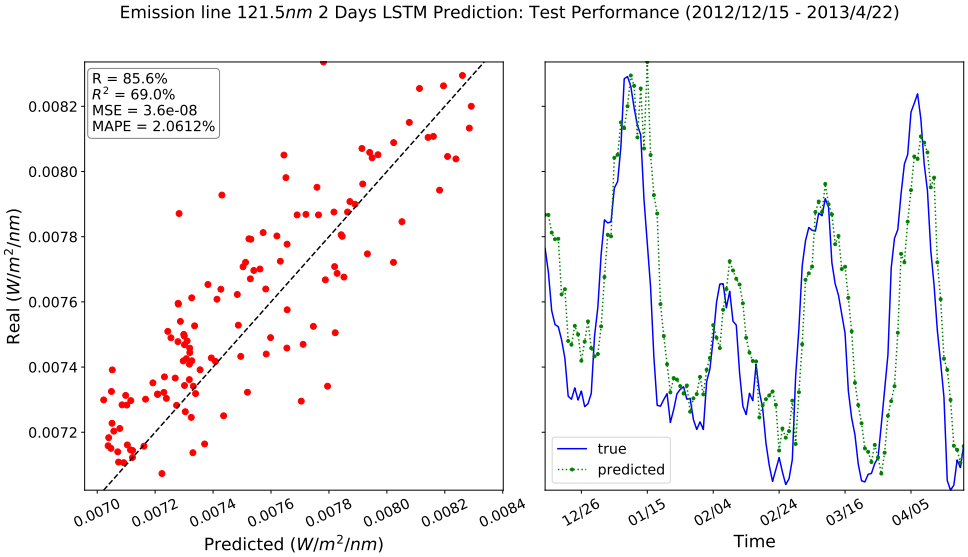


Figure 6.56 - Emission line 121.5nm 3 days forecast training and validation performance with LSTM.

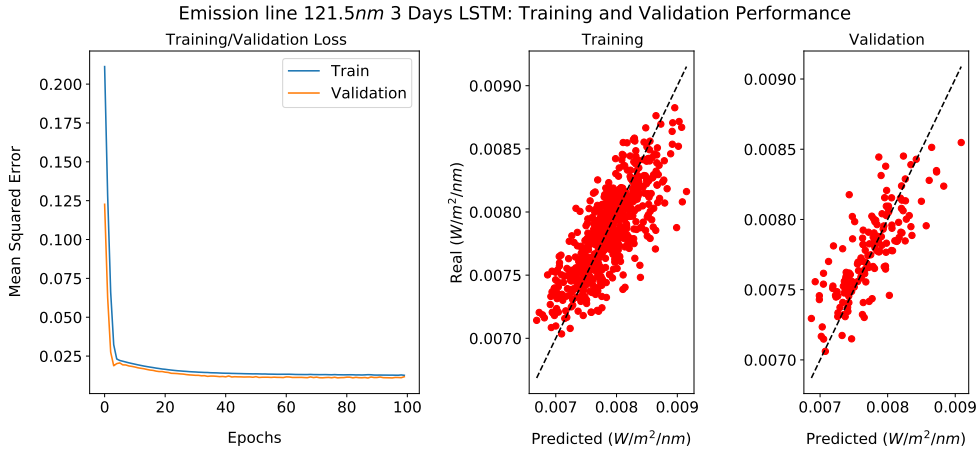
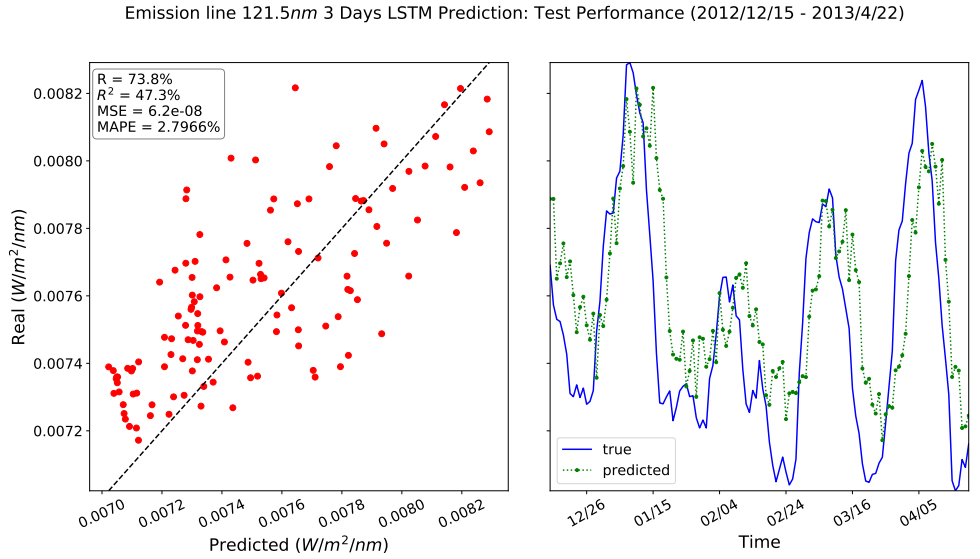


Figure 6.57 - Emission line 121.5nm 3 days forecast test performance with LSTM net.



The $lyman-\alpha$ was the emission line that showed the best prediction results, among the spectral lines tested. Looking at the one-day forecast, even if the forecast line does not follow the real measurements very closely, it can be seen that the forecast is able to pass, in general, a satisfactory trend of rising and falling values. The accuracy is close to the TSI one-day forecast, although the errors are higher.

$Lyman-\alpha$'s other time intervals (two and three days) predictions showed similar behavior to that obtained by the 30.5nm line, in which the forecast quality worsens

to longer time intervals, but seems to maintain the same pattern as the one-day forecast.

6.6.4 Additional emission lines

Some emission lines in a different range than those tested previously were used for one-day prediction. Two lines in the visible range - green ($549.41nm$) and red ($698.85nm$) - and one in the infrared range ($798.83nm$) were chosen. This extra experiment was performed to check the behavior of its prediction and analyze similarities or differences obtained by the initial lines.

After testing values for hyper-parameters, in a non-systematic way, the same hyper-parameters were used for all three lines. The values are reported in Table 6.4:

Table 6.4 - Visible light and infrared default hyper-parameters.

Hyper-Parameters	Values
Hidden Units number	30
Epochs	50
Output Scaling	Std
Input Activation	$tanh$
Dropout rate	0.3
Batch Size	1%

Figures 6.58, 6.59, and 6.60 show the training, validation and test for one day forecast, obtained by the LSTM network, for emission lines $549.41nm$ (green), $698.85nm$ (red) and $798.83nm$ (infrared), respectively.

The results obtained by the extra lines were much lower than the others observed in this work. For this reason, experiments with them were not carried out. But we observed a characteristic present in them, especially in the red and infrared spectra: in parts of the graph, the prediction curve seems to follow the real curve well, but in other parts the curves are far apart. The question for this behavior is whether there is something besides the information of active regions and sunspots that may be influencing the variation of these emission lines.

Figure 6.58 - Emission line 549.41nm 1 day forecast: training, validation, and test performance with LSTM net.

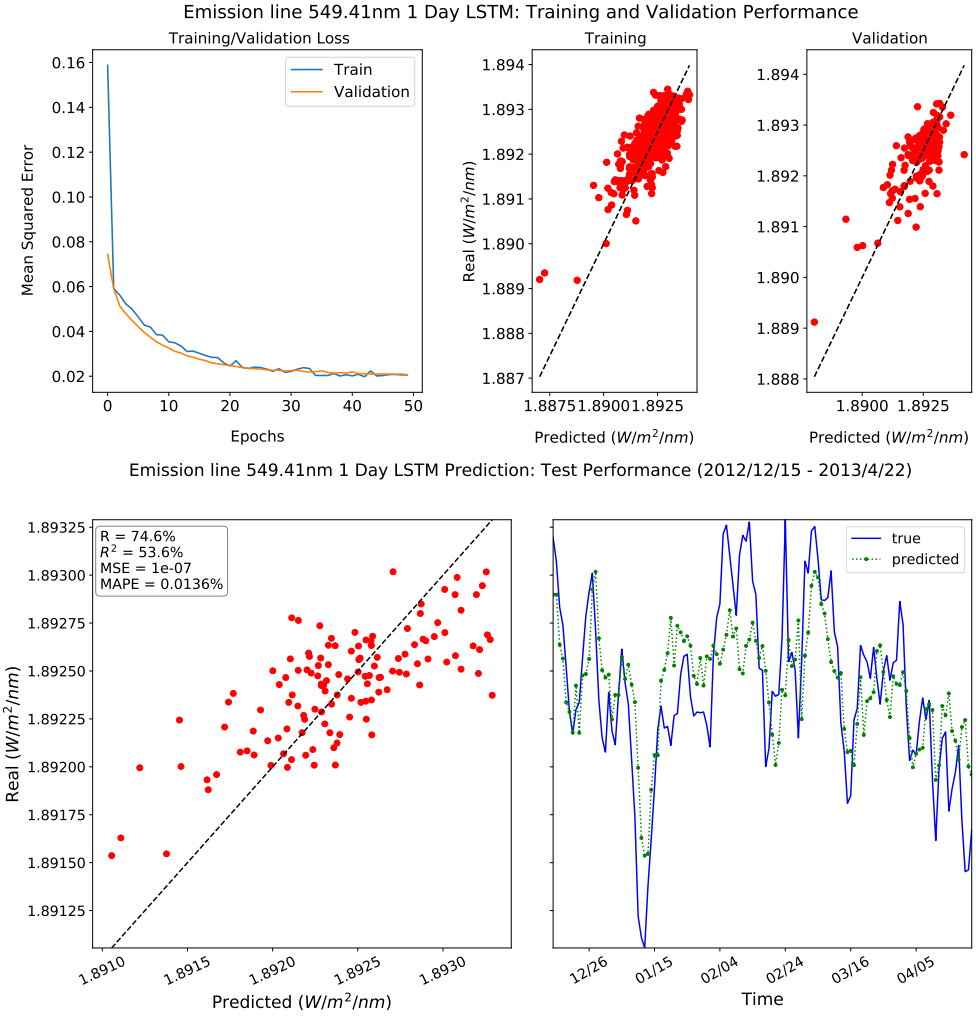


Figure 6.59 - Emission line 698.85nm 1 day forecast: training, validation, and test performance with LSTM net.

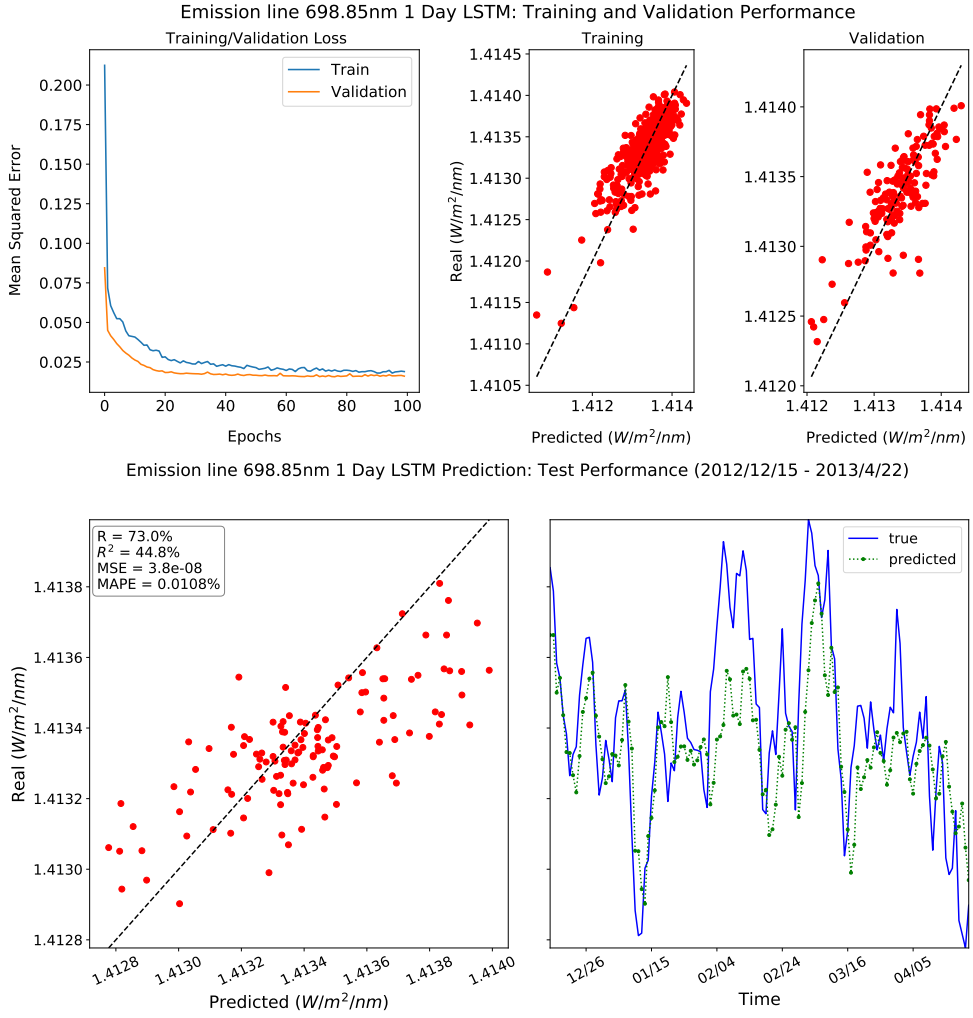
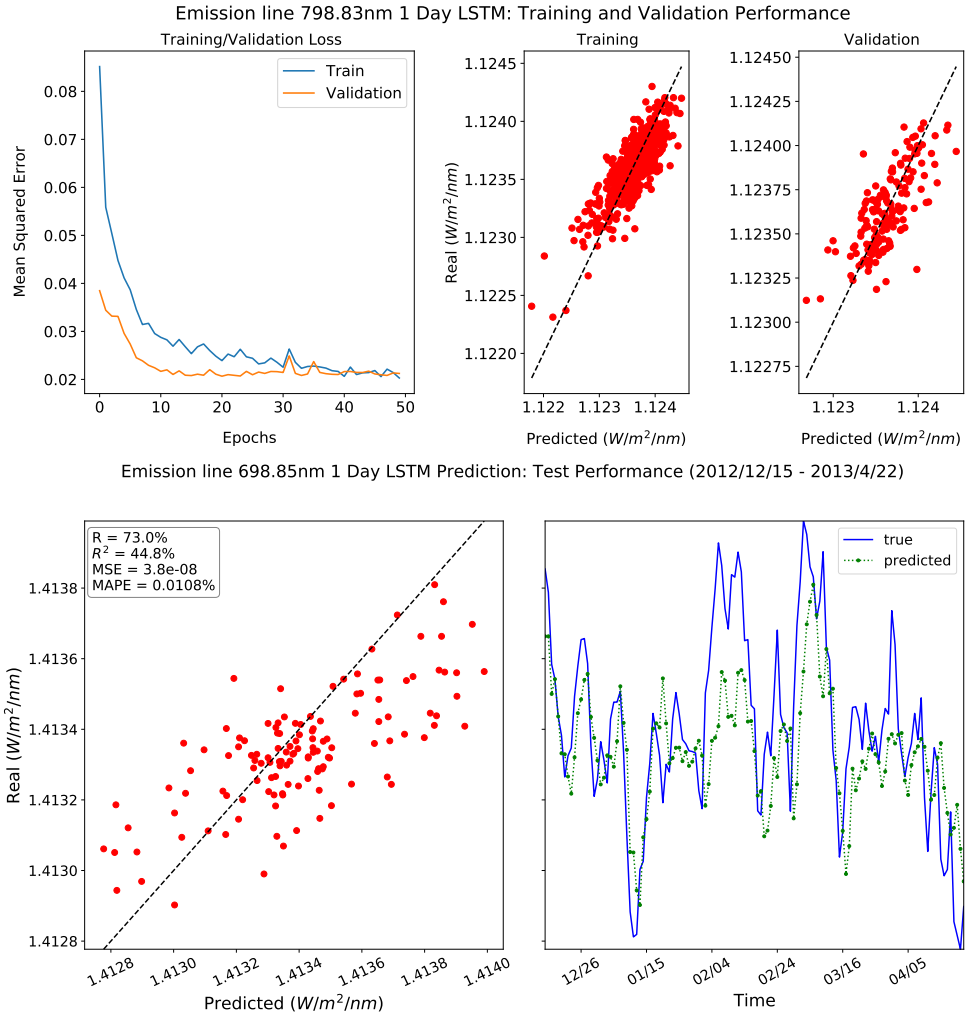


Figure 6.60 - Emission line 798.83nm 1 day forecast: training, validation, and test performance with LSTM net.



6.7 RNN vs physical models

As the last group of experiments, the RNNs predictions were compared to the SATIRE and EMPIRE physical models reconstructions. For SSI, only the emission line for $lyman-\alpha$ was found in the database from where SATIRE and EMPIRE data were downloaded: the Max Planck Institute for Solar System Research[2] webpage, so we chose to compare only this emission line results with the physical models, as well as the TSI.

Before evaluating the results quality, it is important to consider that the physical

[2]Max Plack Institute webpage from where the reconstructed data were downloaded: `http://www2.mps.mpg.de/projects/sun-climate/data.html`, accessed on April of 2021.

reconstruction models may have a different "calibration" for the TSI and SSI values. This is because each irradiance measurement instrument, as already mentioned in Chapter 2, has its own calibration; and the physical models, being independent, also have their own "calibration", or range of values, which may vary from the calibration of the measurement instruments that generated the data used in this work.

Tables 6.5 and 6.6 show loss and accuracy mean values over twenty training processes each obtained for TSI and $lyman - \alpha$ prediction tests, respectively, for the gated RNN architectures. The tables also show the same metrics applied to the physical models for the same data period.

Table 6.5 - RNNs performance for TSI prediction (mean values), and the same metrics applied to physical models' (SATIRE and EMPIRE) daily reconstructed values. This performance was obtained for the test period of 2012/11/20 to 2012/12/18.

	MSE	MAPE(%)	R^2(%)	R(%)
Six hours LSTM	0.003	0.003	96.4	98.3
Six hours GRU	0.003	0.003	96.3	98.3
SATIRE TSI	0.021	0.010	76.8	98.1
EMPIRE TSI	0.012	0.006	86.8	93.3

Table 6.6 - RNNs performance for SSI prediction (mean loss and mean accuracy), and the same metrics applied to physical models' (SATIRE and EMPIRE) daily reconstructed $lyman - \alpha$ values. This performance was obtained for the test period of 2012/12/15 to 2013/4/23.

	MSE	MAPE(%)	R^2(%)	R(%)
$Ly - \alpha$ 1 day LSTM	$1.7e-08$	1.3	85.6	93.3
$Ly - \alpha$ 1 day GRU	$2.0e-08$	1.5	83.1	93.2
SATIRE $(ly - \alpha)$	$4.2e-08$	2.4	63.5	97.6
EMPIRE $(ly - \alpha)$	$1.2e-07$	4.4	0.05	98.3

Looking at Table 6.5 and doing a joint analysis of the four performance metrics, we notice that the RNNs have smaller loss than the physical models. One assumption for this result is that the networks' predicted values are not so deviating from the real values even though the RNNs' correlation is often lower than the physical models'; while even though the physical model curves are well correlated with the real values, sometimes they shift up or down considerably from the real curve, probably

because of the "calibration" discrepancy already mentioned. This behavior is ratified by the R^2 values, which indicates the quality of the fitting between real values and the estimated ones, showed in Figure 6.61. R^2 values have shown themselves to be considerably higher on RNNs' predictions when compared to the physical models. Overall, it can be seen that the TSI prediction results offered by RNNs for time intervals of up to one day are near to the accuracy offered by reconstruction models. The predictions made by LSTM and GRU are almost overwriting each other, confirming once again the similarity of their performances.

In the sequence, Figures 6.62 and 6.61 present plots with those comparisons, providing a graphical visualization of the predictions beside the reconstructions.

Figure 6.61 - TSI six hours prediction using RNNs vs physical models one day reconstructions.

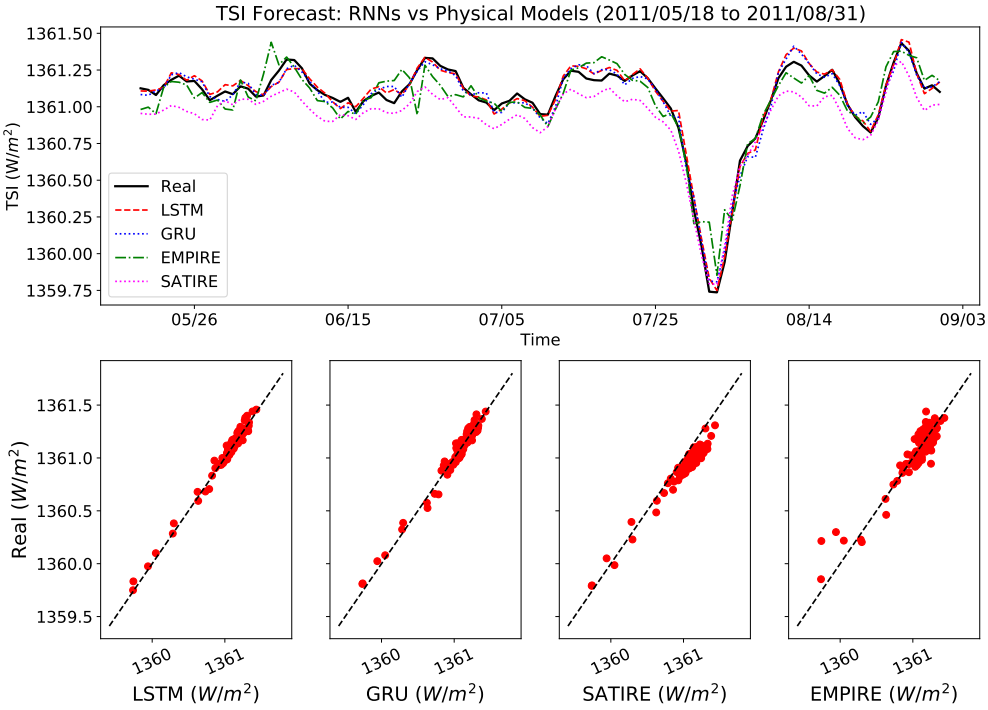
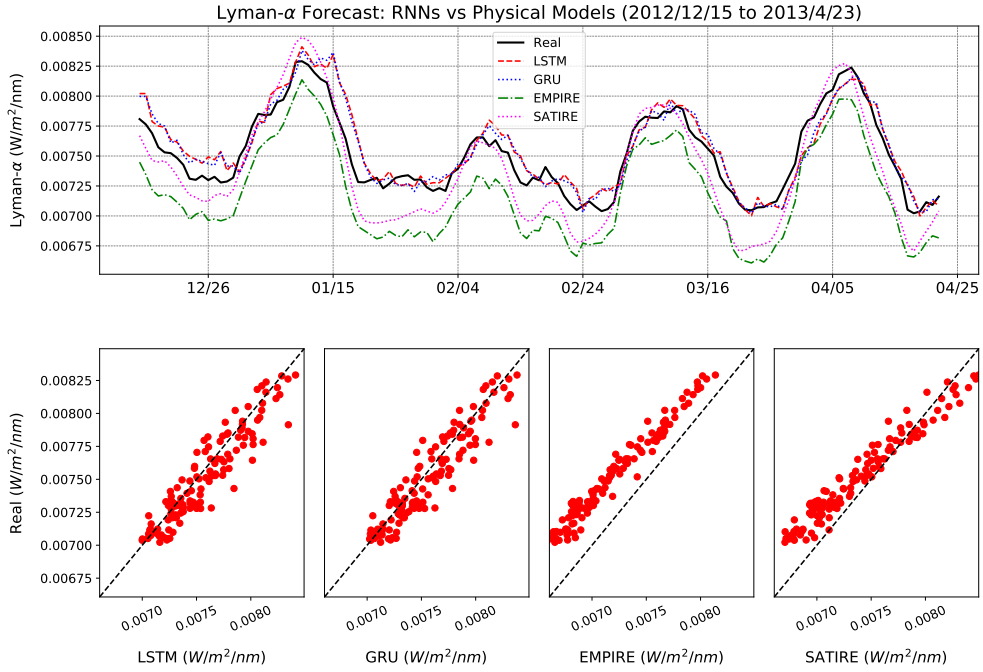


Figure 6.62 - $Lyman-\alpha$ 1 day prediction using LSTM vs physical models reconstructions.



The same considerations about "calibration" made for the comparison of the physical models with TSI can be applied for their comparison with $lyman-\alpha$. The considerations about the response similarity of the two gate architectures are also repeated for $lyman-\alpha$.

Despite presenting a slightly lower accuracy than TSI, $lyman-\alpha$'s prediction also presents quality close to that offered by reconstruction models. A shift is perceived in some moments of the forecast both in the $lyman-\alpha$ and in the physical models in relation to the actual curve. It is observed that the shift occurs, in this case, both to the left and to the right. One assumption is that this characteristic may represent the different response time that may exist between the small variations represented by the photosphere images and the variation in irradiance that is collected at the top of the Earth's atmosphere.

6.8 Conclusions

The experiments looking for the best set of hyper-parameters were very useful in increasing the quality of predictions. But the fact of dealing with many nets at the

same time, and considering processing time for several training runs for each net, may not lead to optimal values for the hyper-parameters. Surely the ideal would be to concentrate on one net at a time to try to tune it as much as possible, which for this work would be unfeasible.

The results obtained for the TSI prediction were promising and lead us to believe that the solar activity status presented by the photosphere images used in this work, with active regions and sunspots, describe well the total irradiance variation, which confirms this relationship.

Therefore, we believe that the same level of prediction quality can be obtained with training performed with data from other programs, such as TSIS (Total and Spectral Solar Irradiance Sensor), which also carries aboard TIM, and which was launched in 2017, collecting TSI and SSI measurements.

The results obtained for the spectral emission lines were overall significantly lower than those obtained for the TSI, and suggest that more effort should be focused on one of them (or a selected range) in order to discover some mechanism to improve their prediction quality. Other solar parameters or other images added as inputs or even better selection of the current parameters may be options to achieve this improvement.

The $lyman-\alpha$, especially, showed satisfactory prediction quality, which may suggest a more direct relationship of its variation to the variation observed in the solar photosphere images, as is clearly observed with TSI.

In many of the prediction plots, both for SSI and TSI, it has been observed that there are shifts forward in time for longer prediction intervals. This shift may be related to the response time between photosphere and measured irradiance, or some other reason that should be investigated taking into account the physical concepts involved in the process.

The results obtained by comparing the actual irradiance values with those reconstructed by physical models suggest that prediction models can be an option to physical models for certain utilities, given that the results were close. Prediction outperforms reconstruction, for example, when there is a need to fill in periods of data collected by a specific instrument, and certainly when it is desired to know values at time instants ahead.

7 CONCLUSIONS

The development of this work allowed us to draw some specific conclusions already exposed in the previous chapters. Some general considerations will be related in this chapter, and future works will also be suggested in order to give continuity to this work, or to reuse it.

7.1 Reproducibility

Regarding reproducibility, it could be concluded that the reproduction of scientific work brings benefits that meet those cited in the references mentioned in Chapter 3. We can relate some of them:

- Allows knowing and understanding the procedure adopted in a paper, and how the final results were achieved. This understanding can even lead to a productive discussion about the approach adopted and whether it is worth trying another approach or method;
- Permits the work in question to be validated so that possible errors or inconsistencies can be detected;
- Makes it possible to reanalyze the whole procedure, its partial and final results, and to draw new conclusions on points that may not have been observed by the original authors;
- Makes it possible to contribute and advance a work, without the need to redo it from the beginning;
- Allows the aggregation of discoveries in ongoing studies, through the sharing of codes and partial and final results, promoting collaborative work;

However, for a work to be reproduced, it must be reproducible, that is, it must follow practices that make it easy for it to be reused. Everyday tools that aid reproducibility emerge, each one focusing on a different need. Making a work reproducible requires a certain effort, but we believe it is a worthwhile effort that can easily become a habit.

7.2 Irradiance prediction with RNN

Recurrent networks have generally shown themselves to be a good option for predicting total and spectral solar irradiance. The results encourage more specific experi-

ments focusing, for example, specific spectra or specific short/long-term predictions, depending on the research being conducted.

The main conclusions made concerning the predictions with RNN were:

a) The gated RNNs, LSTM and GRU, showed superior performance to the simple RNN model.

b) Among the two gated recurrent networks used, there was not one that has highlighted by presenting better results, since the differences between their results were slight, in general;

c) As expected, short-term forecasts (up to one day) offered higher accuracy than long-term forecasts (from two days ahead), but this does not exclude the possibility of obtaining better results for long-term prediction;

d) Since TSI and $lyman - \alpha$ short-term predictions presented similar accuracy compared to physical model reconstructions, we conclude that both gated RNNs (LSTM and GRU) can bring benefits beyond the forecast advantages. They can be an option to provide irradiance values in cases that physical models were the only option. Training a network with a specific instrument's data can allow the estimation of missing values for that specific calibration.

e) For the emission lines that did not obtain enough quality in their predictions, we would bet to consider adding as input other solar parameters that could have relation to the variability of each line or spectral band.

7.3 Final considerations

Although there were numerous ideas to continue with the work, we can state that the proposed objectives were achieved:

a) The workflow was migrated to a free platform, using *Python*, a multiplatform programming language, which can run in free operating systems, like *Linux* distributions. Thus, it will not require a programming language, operating system, or any other proprietary software for it to be reproduced.

b) The effectiveness of the RNNs was proven through the good forecast quality achieved for TSI and $lyman - \alpha$ for predictions from six hours (for

TSI) up to one day (for both). Therefore, it is believed that RNN is a promising technique also for the other emission lines, whose forecast could be improved by seeking new data alternatives, by tuning the networks, or by choosing other options to compose the network input parameters.

c) The workflow was made available in the cloud with its computational elements - data, code, notebooks - as well as the software environment, with its dependencies, so that software versions or lack of code are not barriers to its validation, reproduction, and reuse.

d) The segmentation (modularization) applied in the codes, although there is still some improvement to be made, allows some parts of the workflow to be interchanged. For example, the prediction step itself is being performed on a notebook, where other Machine Learning techniques can easily replace the RNNs to assess and improve prediction performance. Other ML techniques can also be used as options for the identification and classification of the structures present in the solar photosphere images.

7.4 Future works

This work focused on reproducing the original workflow steps without considering modifications to the original algorithms and their strategies. The first suggestion for future work is an experimental work that focuses on process optimization, starting by analyzing the need of changing, editing, or adding steps to the original version of the workflow.

We also suggest taking more varied and larger data periods that cover, for example, different phases of the solar cycle and different cycles. However, for this, it is necessary to treat the existing lack of measured data adequately.

Future works can also consider new strategies to pre-process the input images. One option would be to use another model of solar disk split or another way of weighting sunspots and active regions in different regions of the solar disk by distinguishing, for example, the equatorial region from the polar ones. ML techniques could be considered for some of those tasks.

Simultaneously, it would be interesting to apply an exploratory data analysis preceding the forecast to analyze statistically the input matrices characteristics and the relationship of each one of them with the outputs. This could be significant to choose the more influential input features for the prediction.

Another suggestion to contribute to the level of reproducibility of the work is to make it modular. The new workflow codes already use the CSV standard for the format of their inputs and outputs. The next step would be to separate each sub-task of the workflow into modules. These modules would be independent of each other and could be reused in various ways. The *Dagster*[1] tool is one option for creating a modular workflow, where a script can be programmed to sequentially execute a set of modules. The ability for modules written in one programming language (e.g. *Python*) to run in conjunction with modules written in another programming language (e.g. R) is among the benefits of modularity. Reusing part of the workflow and editing the code of only one of the tasks to, for example, use a new technique instead of the one originally suggested, would be another benefit of modularity.

[1]Dagster's webpage: `https://dagster.io/`. Accessed in October 2021.

REFERENCES

ABRAHAM, A. Artificial neural networks. **Handbook of measuring system design**, Chichester, UK: John Wiley & Sons, 2005. 72

ACADEMY, D. S. **Deep learning book**. [s.n.], 2021. Available from: <https://www.deeplearningbook.com.br/>. Access in: 18 Aug. 2021. 87

AFGAN, E. et al. The galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. **Nucleic Acids Research**, v. 46, n. W1, p. W537–W544, 2018. 38

ALISSON, E. Erros em artigos científicos brasileiros são mais conceituais do que de expressão. **Revista Brasileira de Cirurgia Cardiovascular/Brazilian Journal of Cardiovascular Surgery**, v. 28, n. 1, p. 148–149, 2013. 33

ALLISON, D. B.; SHIFFRIN, R. M.; STODDEN, V. Reproducibility of research: issues and proposed remedies. **Proceedings of the National Academy of Sciences**, v. 115, n. 11, p. 2561–2562, 2018. Available from: <https://doi.org/10.1073/pnas.1802324115>. 32

ANDRADE, R. C. de; TIBA, C. Extreme global solar irradiance due to cloud enhancement in northeastern brazil. **Renewable Energy**, v. 86, p. 1433–1441, 2016. 23

ASOREY, H. et al. Data accessibility, reproducibility and trustworthiness with lago data repository. In: **INTERNATIONAL COSMIC RAY CONFERENCE**. [S.l.: s.n.], 2015. p. 1–8. 38

BADESCU, V. **Modeling solar radiation at the earth's surface**. [S.l.]: Springer, 2014. 2, 23

BAKER, M. Reproducibility crisis. **Nature**, v. 533, n. 26, p. 353–66, 2016. 32, 33, 37

_____. Why scientists must share their research code. **Nature News**, v. 1, p. 20504, 2016. 34, 35

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. **Journal of the Atmospheric Sciences**, v. 71, n. 11, p. 4086–4101, 2014. 23

BALL, W. T.; UNRUH, Y. C.; KRIVOVA, N. A.; SOLANKI, S.; HARDER, J. W. Solar irradiance variability: a six-year comparison between sorce observations and the satire model. **Astronomy & Astrophysics**, v. 530, p. A71, 2011. 23

BARBA, L. A. The hard road to reproducibility. **Science**, v. 354, n. 6308, p. 142–142, 2016. 37

BEAULIEU-JONES, B. K.; GREENE, C. S. Reproducibility of computational workflows is automated using continuous analysis. **Nature Biotechnology**, v. 35, n. 4, p. 342–346, 2017. 34, 38

BENUREAU, F. C.; ROUGIER, N. P. Re-run, repeat, reproduce, reuse, replicate: transforming code into scientific contributions. **Frontiers in Neuroinformatics**, v. 11, p. 69, 2018. 38

BERG, J. **Progress on reproducibility**. [S.l.]: American Association for the Advancement of Science, 2018. 34

BONACCORSO, G. **Machine learning algorithms**. [S.l.]: Packt Publishing, 2017. 69, 70

BOTTOU, L. Stochastic gradient descent tricks. In: **Neural networks: tricks of the trade**. [S.l.]: Springer, 2012. p. 421–436. 86

BOYLAN, J. E.; GOODWIN, P.; MOHAMMADIPOUR, M.; SYNTETOS, A. A. Reproducibility in forecasting research. **International Journal of Forecasting**, v. 31, n. 1, p. 79–90, 2015. 32

BRASSEUR, G. P.; SOLOMON, S. **Aeronomy of the middle atmosphere: chemistry and physics of the stratosphere and mesosphere**. [S.l.]: Springer Science & Business Media, 2006. 19

BURRELL, A.; YEOMAN, T.; STEPHEN, M.; LESTER, M. Influence of solar irradiance on polar ionospheric convection. In: AGU FALL MEETINGS, 2016. **Abstracts...** [S.l.]: AGU, 2016. 22

CACIOPPO, J. T.; KAPLAN, R. M.; KROSNICK, J. A.; OLDS, J. L.; DEAN, H. **Social, behavioral, and economic sciences perspectives on robust and reliable science**. [S.l.: s.n.], 2015. 33

CASSIDY, S.; ESTIVAL, D. Supporting accessibility and reproducibility in language research in the alveo virtual laboratory. **Computer Speech & Language**, v. 45, p. 375–391, 2017. 38

CECATTO, J. R. O sol. In: **Introdução à astronomia e astrofísica v II**. São José dos Campos: Instituto Nacional de Pesquisas Espaciais, 2003. p. 61. Available from: <http://urlib.net/rep/8JMKD3MGP3W34R/3SLL5LL>. Access in: 06 maio 2021. 7

CHIRIGATI, F.; RAMPIN, R.; SHASHA, D.; FREIRE, J. Reprozip: computational reproducibility with ease. In: INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, 2016. **Proceedings...** [S.l.]: ACM, 2016. p. 2085–2088. ISBN 978-1-4503-3531-7. 41

CHO, K.; MERRIËNBOER, B. V.; GULCEHRE, C.; BAHDANAU, D.; BOUGARES, F.; SCHWENK, H.; BENGIO, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. **arXiv preprint arXiv:1406.1078**, 2014. 79

CLAERBOUT, J. **Making scientific contributions reproducible**. 2011. Available from: <http://sepwww.stanford.edu/oldsep/matt/join/redoc/web/iris.html>. Access in: 16 July 2021. 37

CODDINGTON, O.; LEAN, J.; PILEWSKIE, P.; SNOW, M.; LINDHOLM, D. A solar irradiance climate data record. **Bulletin of the American Meteorological Society**, v. 97, n. 7, p. 1265–1282, 2016. 22

DA, K. A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014. 86

DAS, S.; DEY, A.; PAL, A.; ROY, N. Applications of artificial intelligence in machine learning: review and prospect. **International Journal of Computer Applications**, v. 115, n. 9, 2015. 69

D'AZAMBUJA, L. Le colloque de physique solaire organisé à rome par l'académie des lincei, en septembre 1952. **L'Astronomie**, v. 67, p. 129, 1953. 14

DIAGNE, M.; DAVID, M.; LAURET, P.; BOLAND, J.; SCHMUTZ, N. Review of solar irradiance forecasting methods and a proposition for small-scale insular grids. **Renewable and Sustainable Energy Reviews**, v. 27, p. 65–76, 2013. 22

DIRNBERGER, D.; BLACKBURN, G.; MÜLLER, B.; REISE, C. On the impact of solar spectral irradiance on the yield of different pv technologies. **Solar Energy Materials and Solar Cells**, v. 132, p. 431–442, 2015. 22

DOMINGO, V.; FLECK, B.; POLAND, A. Soho: the solar and heliospheric observatory. **Space Science Reviews**, v. 72, n. 1-2, p. 81–84, 1995. 9

DRIEL-GESZTELYI, L. V.; GREEN, L. M. Evolution of active regions. **Living Reviews in Solar Physics**, v. 12, n. 1, p. 1, 2015. 14, 15

ECHER, E.; RIGOZO, N. R.; NORDEMANN, D. J. R.; VIEIRA, L. E. A.; PRESTES, A.; FARIA, H. H. de. O número de manchas solares, índice da atividade do sol. **Revista Brasileira de Ensino de Física**, v. 25, n. 2, p. 157–163, 2003. 12

ECHER, E.; SOUZA, M. P.; SCHUCH, N. J. A lei de beer aplicada na atmosfera terrestre. **Revista Brasileira de Ensino de Física**, v. 23, p. 276 – 283, 09 2001. ISSN 1806-1117. Available from: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1806-11172001000300004&nrm=iso>. 19

EDDY, J. A. The maunder minimum. **Science**, v. 192, n. 4245, p. 1189–1202, 1976. 10

ELSEVIER. **The data availability statement**. 2021. Available from: <https://scientific-publishing.webshop.elsevier.com/manuscript-preparation/data-availability-statement/>. Access in: 15 July 2021. 36

ERMOLLI, I.; MATTHES, K.; WIT, T. Dudok de; KRIVOVA, N. A.; TOURPALI, K.; WEBER, M.; UNRUH, Y. C.; GRAY, L.; LANGEMATZ, U.; PILEWSKIE, P. et al. Recent variability of the solar spectral irradiance and its impact on climate modelling. **Atmospheric Chemistry and Physics**, v. 13, n. 8, p. 3945–3977, 2013. 19, 20, 22

FETZER, J. H. What is artificial intelligence? In: FETZER, J. H. (Ed.). **Artificial intelligence: its scope and limits**. [S.l.]: Springer, 1990. p. 3–27. 69

FLIGGE, M.; SOLANKI, S.; UNRUH, Y.; FRÖHLICH, C.; WEHRLI, C. A model of solar total and spectral irradiance variations. **Astronomy and Astrophysics**, v. 335, p. 709–718, 1998. 23

FONDRIEST. **FONDRIEST Environmental webpage**. 2021. Available from: <https://www.fondriest.com/environmental-measurements/parameters/weather/photosynthetically-active-radiation/>. Access in: 30 July 2021. 19

FONTENLA, J.; HARDER, J.; LIVINGSTON, W.; SNOW, M.; WOODS, T. High-resolution solar spectral irradiance from extreme ultraviolet to far infrared. **Journal of Geophysical Research: Atmospheres**, v. 116, n. D20, 2011. 23

FRÖHLICH, C. Total solar irradiance: what have we learned from the last three cycles and the recent minimum? **Space Science Reviews**, v. 176, n. 1, p. 237–252, 2013. 23

FUENTES, M. Reproducible research in jasa. **AMSTAT news: the membership magazine of the American Statistical Association**, n. 469, p. 17, 2016. Available from: <https://magazine.amstat.org/blog/2016/07/01/jasa-reproducible16/>. Access in: 16 July 2021. 36

GALILEI, G.; MONTINARI, M.; WELSER, M. **Istoria e dimostrazioni intorno alle macchie solari e loro accidenti**. [S.l.]: Roma: Theoria, 1982. 10

GARDNER, M. W.; DORLING, S. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. **Atmospheric Environment**, v. 32, n. 14-15, p. 2627–2636, 1998. 72, 73

GERS, F. A.; SCHMIDHUBER, J.; CUMMINS, F. Learning to forget: continual prediction with lstm. **Neural Computation**, v. 12, n. 10, p. 2451–2471, 2000. 77

GIBB, B. C. Reproducibility. **Nature Chemistry**, v. 6, n. 8, p. 653–654, 2014. 33

GÓMEZ, J. M. R. **Evolution of the electron density, temperature distribution in the solar corona during solar cycles 23 and 24**. Thesis (Doctoral in Geophysics) — Instituto Nacional de Pesquisas Espaciais (INPE), 2017. Available from: <http://urlib.net/sid.inpe.br/mtc-m21b/2017/02.16.21.15>. Access in: 21 fev. 2018. 2

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A.; BENGIO, Y. **Deep learning**. Cambridge: MIT Press, 2016. 74, 75, 87

GOODMAN, S. N.; FANELLI, D.; IOANNIDIS, J. P. What does research reproducibility mean? **Science Translational Medicine**, v. 8, n. 341, p. 341ps12–341ps12, 2016. 29, 30, 31

GORGOLEWSKI, K. J. et al. Bids apps: improving ease of use, accessibility, and reproducibility of neuroimaging data analysis methods. **PLoS Computational**

Biology, v. 13, n. 3, p. e1005209, 2017. Available from: <https://doi.org/10.1371/journal.pcbi.1005209>. 34

GRAY, L. J. et al. Solar influences on climate. **Reviews of Geophysics**, v. 48, n. 4, 2010. 16

GREFF, K.; SRIVASTAVA, R. K.; KOUTNÍK, J.; STEUNEBRINK, B. R.; SCHMIDHUBER, J. Lstm: A search space odyssey. **IEEE Transactions on Neural Networks and Learning Systems**, v. 28, n. 10, p. 2222–2232, 2016. 77

GRÜNING, B.; CHILTON, J.; KÖSTER, J.; DALE, R.; SORANZO, N.; BEEK, M. van den; GOECKS, J.; BACKOFEN, R.; NEKRUTENKO, A.; TAYLOR, J. Practical computational reproducibility in the life sciences. **Cell Systems**, v. 6, n. 6, p. 631–635, 2018. 34, 39

GULLI, A.; PAL, S. **Deep learning with Keras**. [S.l.]: Packt Publishing, 2017. 74, 77, 79, 80

GUO, J. Backpropagation through time. **Unpubl. ms., Harbin Institute of Technology**, v. 40, p. 1–6, 2013. 73, 76

HAIBE-KAINS, B. et al. Transparency and reproducibility in artificial intelligence. **Nature**, v. 586, n. 7829, p. E14–E16, 2020. Available from: <https://doi.org/10.1038/s41586-020-2766-y>. 30, 32

HAIGH, J. D. The impact of solar variability on climate. **Science**, v. 272, n. 5264, p. 981–984, 1996. 1

HAIGH, J. D.; WINNING, A. R.; TOUMI, R.; HARDER, J. W. An influence of solar spectral variations on radiative forcing of climate. **Nature**, v. 467, n. 7316, p. 696–699, 2010. 22

HALE, G. E.; ELLERMAN, F.; NICHOLSON, S. B.; JOY, A. H. The magnetic polarity of sun-spots. **The Astrophysical Journal**, v. 49, p. 153, 1919. 13

HANSON, B.; SUGDEN, A.; ALBERTS, B. **Making data maximally available**. [S.l.]: American Association for the Advancement of Science, 2011. 36

HARDER, J.; LAWRENCE, G.; FONTENLA, J.; ROTTMAN, G.; WOODS, T. The spectral irradiance monitor: scientific requirements, instrument design, and operation modes. In: ROTTMAN, G.; WOODS, T.; GEORGE, V. (Ed.). **The Solar Radiation and Climate Experiment (SORCE)**. [S.l.]: Springer, 2005. p. 141–167. 27

HATHAWAY, D. H. The solar cycle. **Living Reviews in Solar Physics**, v. 7, n. 1, p. 1, Mar 2010. ISSN 1614-4961. Available from: <https://doi.org/10.12942/lrsp-2010-1>. 13, 14

_____. _____. **Living Reviews in Solar Physics**, v. 12, n. 1, p. 4, 2015. 11, 17

HOCHREITER, S. Recurrent neural net learning and vanishing gradient. **International Journal Of Uncertainity, Fuzziness and Knowledge-Based Systems**, v. 6, n. 2, p. 107–116, 1998. 76

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural Computation**, v. 9, n. 8, p. 1735–1780, 1997. 76

HOYT, D. V.; SCHATTEN, K. H. **The role of the sun in climate change**. [S.l.]: Oxford University Press, 1997. 12

HUTSON, M. **Artificial intelligence faces reproducibility crisis**. [S.l.]: American Association for the Advancement of Science, 2018. 32

INPE - NATIONAL INTITUTE FOR SPACE RESEARCH. **EMBRACE's web page**. 2021. Available from: <http://www2.inpe.br/climaespacial/SpaceWeatherTSIViewer/pt/inicio>. Access in: 15 July 2021. 3

IVIE, P.; THAIN, D. Reproducibility in scientific computing. **ACM Computing Surveys (CSUR)**, v. 51, n. 3, p. 1–36, 2018. 36

JIN, J.; LI, M.; JIN, L. Data normalization to accelerate training for linear neural net to predict tropical cyclone tracks. **Mathematical Problems in Engineering**, v. 2015, 2015. 84

JOINT SCIENCE OPERATIONS CENTER (JSOC) / SCIENCE DATA PROCESSING (SDP). **JSOC-SDP**. 2021. Available from: <http://jsoc.stanford.edu/data/hmi/images>. Access in: 16 July 2021. 26

JOZEFOWICZ, R.; ZAREMBA, W.; SUTSKEVER, I. An empirical exploration of recurrent network architectures. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 2015. **Proceedings...** [S.l.], 2015. p. 2342–2350. 77, 80

KAELBLING, L. P.; LITTMAN, M. L.; MOORE, A. W. Reinforcement learning: a survey. **Journal of Artificial Intelligence Research**, v. 4, p. 237–285, 1996. 70

KALANTARY, S.; JAHANI, A.; POURBABAKI, R.; BEIGZADEH, Z. Application of ann modeling techniques in the prediction of the diameter of pcl/gelatin nanofibers in environmental and medical studies. **RSC Advances**, v. 9, n. 43, p. 24858–24874, 2019. 82

KAPPES, H. B. et al. The reproducibility project: a model of large-scale collaboration for empirical research on reproducibility. [S.l.]: CRC Press, Taylor & Francis Group, 2014. Available from: <http://eprints.lse.ac.uk/65169/>. 33

KERAS. **Keras documentation**. 2021. Available from: <https://keras.io/>. Access in: 16 Aug 2021. 80, 81

KIEPENHEUER, K. O. Preface and introduction. In: KIEPENHEUER, K. O. (Ed.). **Structure and development of solar active regions**. [S.l.]: Springer, 1968. p. 3–9. 14

KING, G. How not to lie with statistics: avoiding common mistakes in quantitative political science. **American Journal of Political Science**, p. 666–687, 1986. Available from: <https://doi.org/10.2307/2111095>. 32

KOPP, G. An assessment of the solar irradiance record for climate studies. **Journal of Space Weather and Space Climate**, v. 4, p. A14, 2014. 21

KOPP, G.; LAWRENCE, G. The total irradiance monitor (tim): instrument design. **Solar Physics**, v. 230, n. 1-2, p. 91–109, 2005. 27

KRIVOVA, N.; SOLANKI, S.; FLIGGE, M.; UNRUH, Y. Reconstruction of solar irradiance variations in cycle 23: Is solar surface magnetism the cause? **Astronomy & Astrophysics**, v. 399, n. 1, p. L1–L4, 2003. 2, 23

KRIVOVA, N. A.; VIEIRA, L. E. A.; SOLANKI, S. K. Reconstruction of solar spectral irradiance since the maunder minimum. **Journal of Geophysical Research: Space Physics**, v. 115, n. A12, 2010. ISSN 2156-2202. A12112. Available from: <http://dx.doi.org/10.1029/2010JA015431>. 23

KUHN, M. et al. **Applied predictive modeling**. [S.l.]: Springer, 2013. 82

KULKARNI, N.; ALESSANDRÌ, L.; PANERO, R.; ARIGONI, M.; OLIVERO, M.; FERRERO, G.; CORDERO, F.; BECCUTI, M.; CALOGERO, R. A. Reproducible bioinformatics project: a community for reproducible bioinformatics analysis pipelines. **BMC Bioinformatics**, v. 19, n. 10, p. 5–13, 2018. 34

LAZARIS, C.; KELLY, S.; NTZIACHRISTOS, P.; AIFANTIS, I.; TSIRIGOS, A. Hic-bench: comprehensive and reproducible hi-c data analysis designed for parameter exploration and benchmarking. **BMC Genomics**, v. 18, n. 1, p. 22, 2017. Available from: <https://doi.org/10.1186/s12864-016-3387-6>. 34

LEMEN, J. R. et al. The atmospheric imaging assembly (aia) on the solar dynamics observatory (sdo). In: CHAMBERLIN, P.; PESNELL, W. D.; THOMPSON, B. J. (Ed.). **The solar dynamics observatory**. [S.l.]: Springer, 2011. p. 17–40. 24

LIOU, K.-N. **An introduction to atmospheric radiation**. [S.l.: s.n.], 2002. 18

LIU, L.; WAN, W.; CHEN, Y.; LE, H. Solar activity effects of the ionosphere: a brief review. **Chinese Science Bulletin**, v. 56, n. 12, p. 1202–1211, 2011. 22

MAKIN, T. R.; XIVRY, J.-J. O. de. Science forum: ten common statistical mistakes to watch out for when writing or reviewing a manuscript. **Elife**, v. 8, p. e48175, 2019. 32

MASTERS, D.; LUSCHI, C. Revisiting small batch training for deep neural networks. **arXiv preprint arXiv:1804.07612**, 2018. 87

MAX PLANCK INSTITUTE FOR SOLAR SYSTEM RESEARCH. **Max Planck webpage**. 2021. Available from: <https://www2.mps.mpg.de/projects/sun-climate/se_body.html>. Access in: 30 May 2021. 10

MCCARTHY, J. What is artificial intelligence? 2007. Available from: <http://www-formal.stanford.edu/jmc/whatisai/>. 69

MCCLINTOCK, W. E.; ROTTMAN, G. J.; WOODS, T. N. Solar-stellar irradiance comparison experiment ii (solstice ii): instrument concept and design. In: ROTTMAN, G.; WOODS, T.; GEORGE, V. (Ed.). **The Solar Radiation and Climate Experiment (SORCE)**. [S.l.]: Springer, 2005. p. 225–258. 27

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The Bulletin of Mathematical Biophysics**, v. 5, n. 4, p. 115–133, 1943. 71

MCKINNEY, S. M.; KARTHIKESALINGAM, A.; TSE, D.; KELLY, C. J.; LIU, Y.; CORRADO, G. S.; SHETTY, S. Reply to: Transparency and reproducibility in artificial intelligence. **Nature**, v. 586, n. 7829, p. E17–E18, 2020. Available from: <https://doi.org/10.1038/s41586-020-2767-x>. 32

MCKINNEY, S. M. et al. International evaluation of an ai system for breast cancer screening. **Nature**, v. 577, n. 7788, p. 89–94, 2020. Available from: <https://doi.org/10.1038/s41586-019-1799-6>. 32

MCNUTT, M. **Reproducibility**. [S.l.]: American Association for the Advancement of Science, 2014. 33

MIYAKAWA, T. **No raw data, no science: another possible source of the reproducibility crisis**. [S.l.]: Springer, 2020. 32

MULLANE, K.; WILLIAMS, M. Enhancing reproducibility: failures from reproducibility initiatives underline core challenges. **Biochemical Pharmacology**, v. 138, p. 7–18, 2017. 34

MUNAFÒ, M. R.; NOSEK, B. A.; BISHOP, D. V.; BUTTON, K. S.; CHAMBERS, C. D.; SERT, N. P. D.; SIMONSOHN, U.; WAGENMAKERS, E.-J.; WARE, J. J.; IOANNIDIS, J. P. A manifesto for reproducible science. **Nature Human Behaviour**, v. 1, n. 1, p. 1–9, 2017. 34

MURALIKRISHNA, A.; SANTOS, R.; VIEIRA, L. E. A reproducible solar irradiance estimation process using recurrent neural network. In: EGU GENERAL ASSEMBLY CONFERENCE. **Proceedings...** [S.l.], 2020. p. 1130. 89

MURALIKRISHNA, A.; VIEIRA, L. E.; SANTOS, R. D. dos; ALMEIDA, A. P. A solar spectral irradiance prediction workflow using a recurrent neural network in a reproducible and replicable approach. In: ASTRONOMICAL DATA ANALYSIS SOFTWARE AND SYSTEMS XXX. **Proceedings...** To be published by ASP as ADASS Conference Proceedings, 2020. Available from: <https://adass2020.es/posters/>. 88, 89

MURALIKRISHNA, A.; VIEIRA, L. E. A.; SANTOS, R. D. C. dos; ALMEIDA, A. P. Total solar irradiance forecasting with keras recurrent neural networks. In: GERVASI, O.; MURGANTE, B.; MISRA, S.; GARAU, C.; BLEČIĆ, I.; TANIAR, D.; APDUHAN, B. O.; ROCHA, A. M. A. C.; TARANTINO, E.; TORRE, C. M.; KARACA, Y. (Ed.). **Proceedings of Computational Science and Its Applications – ICCSA 2020**. Cham: Springer International Publishing, 2020. p. 255–269. ISBN 978-3-030-58814-4. 88, 89, 93

NASA. **NASA's Image Gallery**. 2021. Available from: <https://svs.gsfc.nasa.gov/hyperwall/index/download/events/2013-agu/lika/EarthSunSystem_HW.jpg>. Access in: 30 May 2021. 9

_____. **NASA's Imagine the Universe**. 2021. Available from: <https://imagine.gsfc.nasa.gov/science/toolbox/emspectrum1.html>. Access in: 30 May 2021. 18

_____. **NASA's webpage**. 2021. Available from: <https://www.nasa.gov/mission_pages/sunearth/science/Sun-Wavelength-Chart.html>. Access in: 30 July 2021. 25

NASEM. **Reproducibility and replicability in science**. [S.l.]: National Academies Press, 2019. ISBN 978-0-309-48616-3. 30

NATURE. Announcement: Reducing our irreproducibility. **Nature**, v. 496, n. 7446, p. 398, 2013. Available from: <https://doi.org/10.1038/496398a>. 33

_____. Reality check on reproducibility. **Nature**, v. 533, p. 437, 2016. Available from: <https://doi.org/10.1038/533437a>. 29, 32, 34

_____. Reproducibility: let's get it right from the start. **Nature Communications**, v. 9, 2018. Available from: <https://doi.org/10.1038/s41467-018-06012-8>. 34

_____. **Reporting standards and availability of data, materials, code and protocols**. 2021. Available from: <https://www.nature.com/nature-portfolio/editorial-policies/reporting-standards>. Access in: 15 July 2021. 36

NETO, J.; ALMEIDA, L.; HOCHBERG, M.; MARTINS, C.; NUNES, L.; RENALS, S.; ROBINSON, T. **Speaker-adaptation for hybrid HMM-ANN continuous speech recognition system**. [S.l.]: International Speech Communication Association, 1995. 82

NICHOLSON, S. B. The solar cycle. **Leaflet of the Astronomical Society of the Pacific**, v. 7, p. 385, 1958. 10, 11

NINA, A.; ČADEŽ, V.; SREĆKOVIĆ, V.; ŠULIĆ, D. The influence of solar spectral lines on electron concentration in terrestrial ionosphere. **Open Astronomy**, v. 20, n. 4, p. 609–612, 2011. 22

NOGUEIRA, P. et al. Modeling the equatorial and low-latitude ionospheric response to an intense x-class solar flare. **Journal of Geophysical Research: Space Physics**, v. 120, n. 4, p. 3021–3032, 2015. 2, 23

NOSEK, B. A. et al. Promoting an open research culture. **Science**, v. 348, n. 6242, p. 1422–1425, 2015. 35, 36

PARR, A. C.; DATLA, R.; GARDNER, J. **Optical radiometry**. [S.l.: s.n.], 2005. 8, 21

PATIL, P.; PENG, R. D.; LEEK, J. T. A statistical definition for reproducibility and replicability. **BioRxiv**, p. 066803, 2016. Available from: <https://doi.org/10.1101/066803>. 30

PAWLIK, M.; HÜTTER, T.; KOCHER, D.; MANN, W.; AUGSTEN, N. A link is not enough–reproducibility of data. **Datenbank-Spektrum**, v. 19, n. 2, p. 107–115, 2019. 34

PENG, R. The reproducibility crisis in science: a statistical counterattack. **Significance**, v. 12, n. 3, p. 30–32, 2015. 32

PESNELL, W. D.; THOMPSON, B. J.; CHAMBERLIN, P. The solar dynamics observatory (sdo). In: CHAMBERLIN, P.; PESNELL, W. D.; THOMPSON, B. J. (Ed.). **The solar dynamics observatory**. [S.l.]: Springer, 2011. p. 3–15. 10, 24

PIEDEHIERRO, A.; ANTÓN, M.; CAZORLA, A.; ALADOS-ARBOLEDAS, L.; OLMO, F. Evaluation of enhancement events of total solar irradiance during cloudy conditions at granada (southeastern spain). **Atmospheric Research**, v. 135, p. 1–7, 2014. 23

PILKINGTON, S. M.; ENCKE, B.; KROHN, N.; HOEHNE, M.; STITT, M.; PYL, E.-T. Relationship between starch degradation and carbon demand for maintenance and growth in arabidopsis thaliana in different irradiance and temperature regimes. **Plant, Cell & Environment**, v. 38, n. 1, p. 157–171, 2015. 22

PINHEIRO, C. A. O.; SENNA, V. d. Multivariate analysis and neural networks application to price forecasting in the brazilian agricultural market. **Ciência Rural**, v. 47, n. 1, 2017. 74

PLESSER, H. E. Reproducibility vs. replicability: a brief history of a confused terminology. **Frontiers in Neuroinformatics**, Frontiers, v. 11, p. 76, 2018. 29

POPPER, K. **The logic of scientific discovery**. Reprint of 1959 edition. [S.l.]: Martino Fine Books, 2005. ISBN-10 1614277435. 30, 33, 37

PRADHAN, B.; LEE, S. Landslide susceptibility assessment and factor effect analysis: backpropagation artificial neural networks and their comparison with frequency ratio and bivariate logistic regression modelling. **Environmental Modelling & Software**, v. 25, n. 6, p. 747–759, 2010. 82

PUDOVKIN, M. Influence of solar activity on the lower atmosphere state. **International Journal of Geomagnetism and Aeronomy**, v. 5, n. 2, p. I2007, 2004. 22

REDISH, A. D.; KUMMERFELD, E.; MORRIS, R. L.; LOVE, A. C. Opinion: reproducibility failures are essential to scientific inquiry. **Proceedings of the National Academy of Sciences**, v. 115, n. 20, p. 5042–5046, 2018. 33

RIPLEY, B. D. **Pattern recognition and neural networks**. [S.l.]: Cambridge University Press, 2007. 82

Rodríguez Gómez, J.; VIEIRA, L.; LAGO, A. D.; PALACIOS, J. Coronal electron density temperature and solar spectral irradiance during solar cycles 23 and 24. **The Astrophysical Journal**, v. 852, n. 2, p. 137, 2018. 2

Rodríguez Gómez, J.; VIEIRA, L. A.; LAGO, A. D.; PALACIOS, J.; BALMACEDA, L.; STEKEL, T. Modelling short-term solar spectral irradiance (ssi) using coronal electron density and temperature profiles based on solar magnetic field observations. **Proceedings of the International Astronomical Union**, v. 12, n. S327, p. 82–85, 2016. 2, 23

ROJAS, R. The backpropagation algorithm. In: ROJAS, R. (Ed.). **Neural networks**. [S.l.]: Springer, 1996. p. 149–182. 72, 73

ROMERO, F. Philosophy of science and the replicability crisis. **Philosophy Compass**, v. 14, n. 11, p. e12633, 2019. 30

ROTTMAN, G.; WOODS, T.; GEORGE, V. **Solar Radiation and Climate Experiment (SORCE).** [S.l.]: Springer, 2014. 27

ROUGIER, N. P. et al. Sustainable computational science: the rescience initiative. **PeerJ Computer Science**, v. 3, p. e142, 2017. 29

RUSSELL, S.; NORVIG, P. **Artificial intelligence: a modern approach**. [S.l.]: Prentice Hall, 2002. 82

SAKAR, C. O.; POLAT, S. O.; KATIRCIOGLU, M.; KASTRO, Y. Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and

lstm recurrent neural networks. **Neural Computing and Applications**, v. 31, n. 10, p. 6893–6908, 2019. 77, 80

SCHLICH, T. Making mistakes in science: Eduard pflüger, his scientific and professional concept of physiology, and his unsuccessful theory of diabetes (1903–1910). **Studies in History and Philosophy of Science Part A**, v. 24, n. 3, p. 411–441, 1993. Available from: <https://doi.org/10.1016/0039-3681(93)90036-J>. 33

SCHMIDHUBER, J.; WIERSTRA, D.; GAGLIOLO, M.; GOMEZ, F. Training recurrent networks by evolino. **Neural Computation**, v. 19, n. 3, p. 757–779, 2007. 77

SCHOU, J. et al. Design and ground calibration of the helioseismic and magnetic imager (hmi) instrument on the solar dynamics observatory (sdo). In: CHAMBERLIN, P.; PESNELL, W. D.; THOMPSON, B. (Ed.). **The solar dynamics Observatory**. [S.l.]: Springer, 2011. p. 229–259. 12, 24

SCHOVE, D. J. **Sunspot cycles**. [S.l.]: Benchmark Papers in Geology, 1983. 410 p. 10

SCHRÖTER, E. The solar differential rotation: present status of observations. **Solar Physics**, v. 100, n. 1, p. 141–169, 1985. 7

SCIENCE. **American Association for the Advancement of Science**. 2011. Available from: <https://www.sciencemag.org/authors/science-journals-editorial-policies>. Access in: 15 July 2021. 36

SHARMA, S.; SHARMA, S. Activation functions in neural networks. **Towards Data Science**, v. 6, n. 12, p. 310–316, 2017. 83

SOLA, J.; SEVILLA, J. Importance of input data normalization for the application of neural networks to complex industrial problems. **IEEE Transactions on Nuclear Science**, v. 44, n. 3, p. 1464–1468, 1997. 84

SOLANKI, S. K.; INHESTER, B.; SCHÜSSLER, M. The solar magnetic field. **Reports on Progress in Physics**, v. 69, n. 3, p. 563, 2006. 7

SOLAR AND HELIOSPHERIC OBSERVATORY. **SOHO's webpage**. 2021. Available from: <https://sohowww.nascom.nasa.gov/gallery/images/spotcollage.html>. Access in: 30 May 2021. 13

SOLAR INFLUENCES DATA ANALYSIS CENTER. **SIDC/Royal Observatory of Belgium webpage**. 2021. Available from: <http://sidc.oma.be/html/wolfmms.html>. Access in: 30 May 2021. 11

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. **The Journal of Machine Learning Research**, v. 15, n. 1, p. 1929–1958, 2014. 87

STODDEN, V.; MCNUTT, M.; BAILEY, D. H.; DEELMAN, E.; GIL, Y.; HANSON, B.; HEROUX, M. A.; IOANNIDIS, J. P.; TAUFER, M. Enhancing reproducibility for computational methods. **Science**, v. 354, n. 6317, p. 1240–1241, 2016. 34, 35, 36

STODDEN, V.; SEILER, J.; MA, Z. An empirical analysis of journal policy effectiveness for computational reproducibility. **Proceedings of the National Academy of Sciences**, v. 115, n. 11, p. 2584–2589, 2018. Available from: <https://doi.org/10.1073/pnas.1708290115>. 34, 36

STRONG, K. T.; SCHMELZ, J. T.; SABA, J. L.; KUCERA, T. A. Understanding space weather: Part ii: the violent sun. **Bulletin of the American Meteorological Society**, v. 98, n. 11, p. 2387–2396, 2017. 15

STUIVER, M.; QUAY, P. D. Changes in atmospheric carbon-14 attributed to a variable sun. **Science**, v. 207, n. 4426, p. 11–19, 1980. ISSN 0036-8075. Available from: <http://science.sciencemag.org/content/207/4426/11>. 11, 22

SUKHODOLOV, T.; ROZANOV, E.; BALL, W. T.; PETER, T.; SCHMUTZ, W. Modeling of the middle atmosphere response to 27-day solar irradiance variability. **Journal of Atmospheric and Solar-Terrestrial Physics**, v. 152, p. 50–61, 2017. 23

SUTTON, R. S.; BARTO, A. G. **Reinforcement learning: an introduction**. [S.l.]: MIT Press, 2018. 70

TIERNEY, N. J.; RAM, K. A realistic guide to making data available alongside code to improve reproducibility. **arXiv preprint arXiv:2002.11626**, 2020. 34

UHRIN, M.; HUBER, S. P.; YU, J.; MARZARI, N.; PIZZI, G. Workflows in aiida: engineering a high-throughput, event-based engine for robust and modular computational workflows. **Computational Materials Science**, v. 187, p. 110086, 2021. 38

VAQUERO, J. M. A forgotten naked-eye sunspot recorded by galileo. **Solar Physics**, v. 223, n. 1, p. 283–286, 2004. Available from: <https://doi.org/10.1007/s11207-004-1041-y>. 10

VIEIRA, L. E. A.; SOLANKI, S. K.; KRIVOVA, N. A.; USOSKIN, I. Evolution of the solar irradiance during the holocene. **Astronomy & Astrophysics**, v. 531, p. A6, 2011. 2, 23

VIEIRA, L. E. A.; WIT, T. D. de; KRETZSCHMAR, M. Short-term forecast of the total and spectral solar irradiance. **arXiv preprint arXiv:1111.5308**, 2011. 2, 17, 23, 43, 89

VINCENZI, M. D.; FASANO, G. **Monitoring coastal areas: a brief history of measuring instruments for solar radiation**. Firenze University Press Best Practice in Scholarly Publishing, 2020. 676–687 p. Available from: <https://doi.org/10.36253/fup_best_practice>. 10

VOYANT, C.; NOTTON, G.; KALOGIROU, S.; NIVET, M.-L.; PAOLI, C.; MOTTE, F.; FOUILLOY, A. Machine learning methods for solar radiation forecasting: a review. **Renewable Energy**, v. 105, p. 569 – 582, 2017. ISSN 0960-1481. Available from: <http://www.sciencedirect.com/science/article/pii/S0960148116311648>. 22

WALCZAK, S. Artificial neural networks. In: KHOSROW-POUR, M. (Ed.). **Advanced methodologies and technologies in artificial intelligence, computer simulation, and human-computer interaction**. [S.l.]: IGI Global, 2019. p. 40–53. 72

WALT, S. Van der; SCHÖNBERGER, J. L.; NUNEZ-IGLESIAS, J.; BOULOGNE, F.; WARNER, J. D.; YAGER, N.; GOUILLART, E.; YU, T. Scikit-image: image processing in python. **PeerJ**, v. 2, p. e453, 2014. 44, 86

WANG, F.; MI, Z.; SU, S.; ZHAO, H. Short-term solar irradiance forecasting model based on artificial neural network using statistical feature parameters. **Energies**, v. 5, n. 5, p. 1355–1370, 2012. 22

WANG, F.; ZHEN, Z.; MI, Z.; SUN, H.; SU, S.; YANG, G. Solar irradiance feature extraction and support vector machines based weather status pattern recognition model for short-term photovoltaic power forecasting. **Energy and Buildings**, v. 86, p. 427–438, 2015. 22

WERBOS, P. J. Backpropagation through time: what it does and how to do it. **Proceedings of the IEEE**, v. 78, n. 10, p. 1550–1560, 1990. 76

WILLIAMSON, C. E. et al. Solar ultraviolet radiation in a changing climate. **Nature Climate Change**, v. 4, n. 6, p. 434, 2014. 22

WITTMANN, A. D.; XU, Z. A catalogue of sunspot observations from 165 bc to ad 1684. **Astronomy and Astrophysics Supplement Series**, v. 70, p. 83–94, 1987. 10

WOOD, J. M.; NEZWORSKI, M. T. Science as a history of corrected mistakes: comment. **American Psychologist**, American Psychological Association, 2005. Available from: <https://psycnet.apa.org/doi/10.1037/0003-066X.60.6.657>. 33

WOODS, T. et al. Extreme ultraviolet variability experiment (eve) on the solar dynamics observatory (sdo): overview of science objectives, instrument design, data products, and model developments. In: CHAMBERLIN, P.; PESNELL, W. D.; THOMPSON, B. (Ed.). **The solar dynamics observatory**. [S.l.]: Springer, 2010. p. 115–143. 24

WOODS, T. N.; KOPP, G.; CHAMBERLIN, P. C. Contributions of the solar ultraviolet irradiance to the total solar irradiance during large flares. **Journal of Geophysical Research: Space Physics**, v. 111, n. A10, 2006. 23

WOODS, T. N.; ROTTMAN, G.; VEST, R. Xuv photometer system (xps): overview and calibrations. In: ROTTMAN, G.; WOODS, T.; GEORGE, V. (Ed.). **The Solar Radiation and Climate Experiment (SORCE)**. [S.l.]: Springer, 2005. p. 345–374. 27

YANG, H.; KURTZ, B.; NGUYEN, D.; URQUHART, B.; CHOW, C. W.; GHONIMA, M.; KLEISSL, J. Solar irradiance forecasting using a ground-based sky imager developed at uc san diego. **Solar Energy**, v. 103, p. 502 – 524, 2014. ISSN 0038-092X. Available from: <http://www.sciencedirect.com/science/article/pii/S0038092X14001327>. 22

YANG, P.-C.; PURAWAT, S.; IEONG, P. U.; JENG, M.-T.; DEMARCO, K. R.; VOROBYOV, I.; MCCULLOCH, A. D.; ALTINTAS, I.; AMARO, R. E.; CLANCY, C. E. A demonstration of modularity, reuse, reproducibility, portability and scalability for modeling and simulation of cardiac electrophysiology using kepler workflows. **PLoS Computational Biology**, v. 15, n. 3, p. e1006856, 2019. 34, 38, 39

YAU, K.; STEPHENSON, F. A revised catalogue of far eastern observations of sunspots (165 bc to ad 1918). **Quarterly Journal of the Royal Astronomical Society**, v. 29, p. 175–197, 1988. Available from: <http://adsabs.harvard.edu/abs/1988QJRAS..29..175Y>. 10

YEO, K.; BALL, W.; KRIVOVA, N.; SOLANKI, S.; UNRUH, Y.; MORRILL, J. Uv solar irradiance in observations and the nrlssi and satire-s models. **Journal of Geophysical Research: Space Physics**, v. 120, n. 8, p. 6055–6070, 2015. 2, 23

YEO, K.; KRIVOVA, N.; SOLANKI, S. Empire: a robust empirical reconstruction of solar irradiance variability. **Journal of Geophysical Research: Space Physics**, v. 122, n. 4, p. 3888–3914, 2017. 23, 24

YEO, K.; KRIVOVA, N.; SOLANKI, S.; GLASSMEIER, K. Reconstruction of total and spectral solar irradiance from 1974 to 2013 based on kpvt, soho/mdi, and sdo/hmi observations. **Astronomy & Astrophysics**, v. 570, p. A85, 2014. 23, 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. **Journal of Atmospheric and Solar-Terrestrial Physics**, v. 141, p. 1–12, 2016. 22

YU, Y.; SI, X.; HU, C.; ZHANG, J. A review of recurrent neural networks: Lstm cells and network architectures. **Neural Computation**, v. 31, n. 7, p. 1235–1270, 2019. 77, 79

ZAREMBA, W.; SUTSKEVER, I.; VINYALS, O. Recurrent neural network regularization. **arXiv preprint arXiv:1409.2329**, 2014. 87

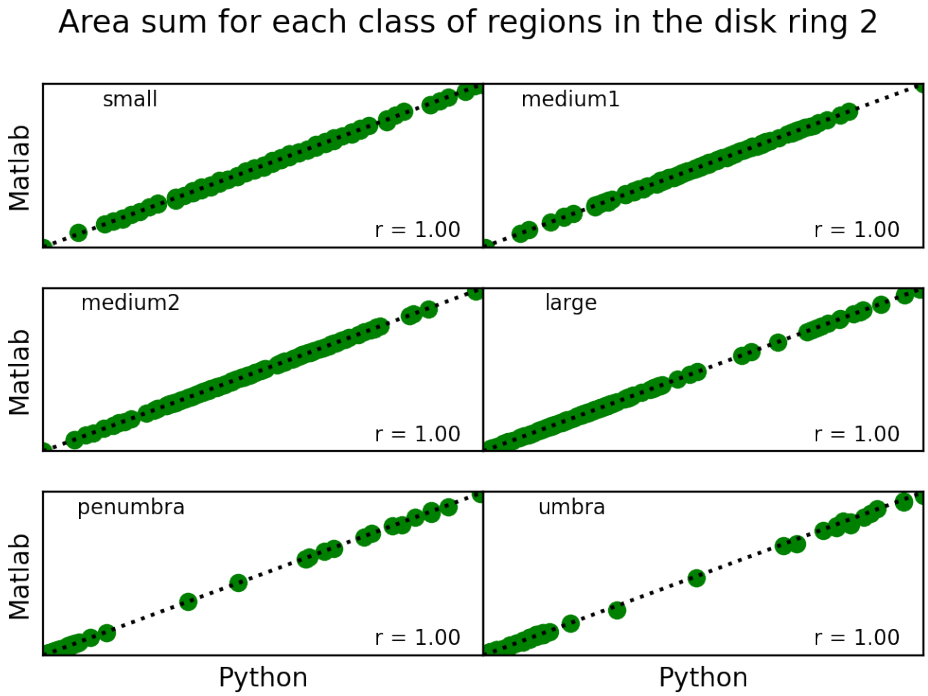
APPENDIX A

This appendix concentrates some of the results that were not showed in the regular chapters.

A.1 Areas sum on solar disk rings

This section presents the results of the areas sum for the solar disk rings not showed in Chapter 4.

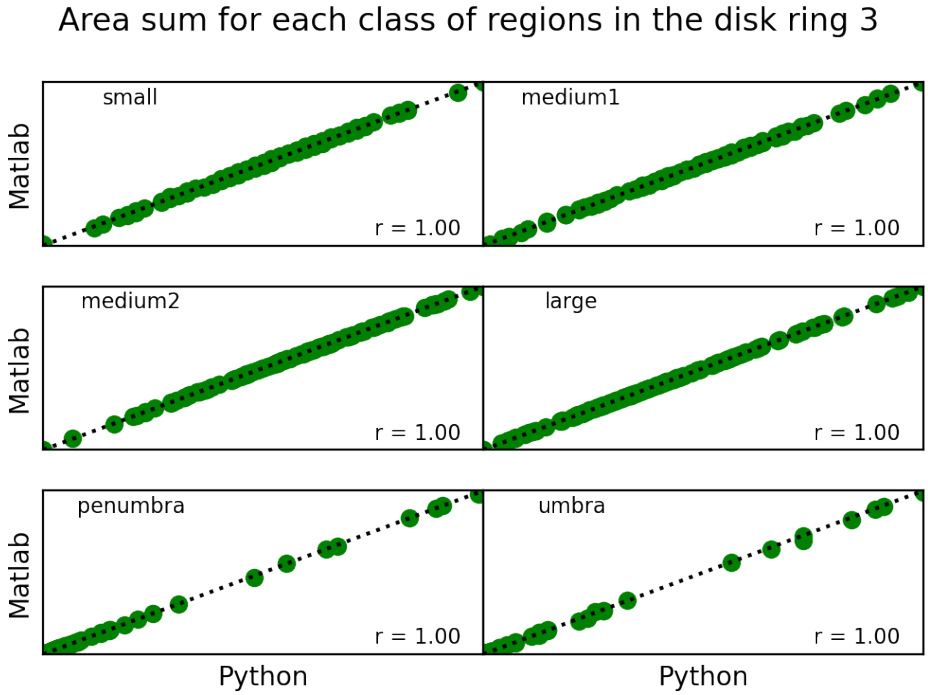
Figure A.1 - *Matlab* and *Python* output matrices comparison: areas sum for each class of regions on solar disk 2^{nd} ring. Time period of October 2014.



A.2 TSI prediction

This section presents the TSI Predictions for 12 to 72 hours, performed by LSTM and GRU, for the test set of 11/20/2012 to 12/18/2012.

Figure A.2 - *Matlab* and *Python* output matrices comparison: areas sum for each class of regions on solar disk 3^{rd} ring. Time period of October 2014.



A.3 SSI prediction

This section presents the training, validation, and test performance for three emission lines: $30.5nm$, $48.5nm$, and $121.5nm$. The LSTM results are presented on Chapter 6, and here the GRU results will be presented, to predictions of one to three days ahead.

Figure A.3 - *Matlab* and *Python* output matrices comparison: areas sum for each class of regions on solar disk 4^{th} ring. Time period of October 2014.

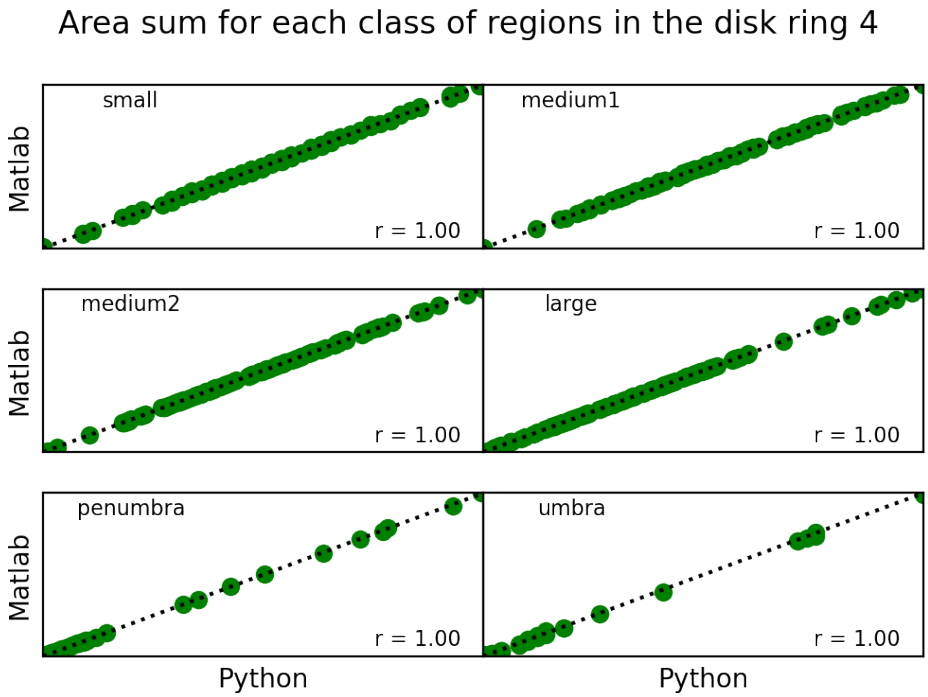


Figure A.4 - *Matlab* and *Python* output matrices comparison: areas sum for each class of regions on solar disk 5^{th} ring. Time period of October 2014.

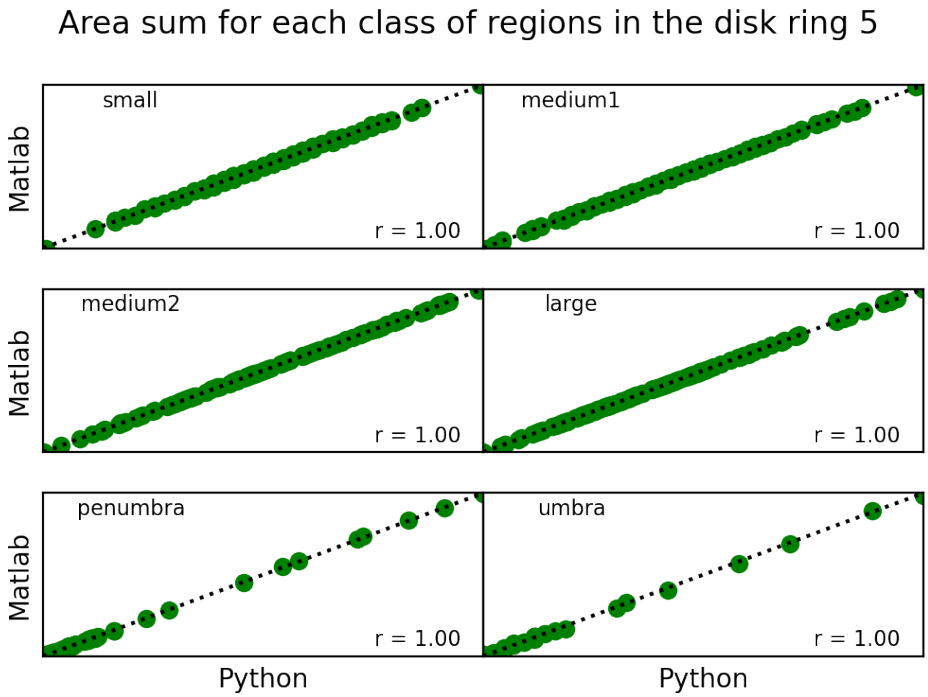


Figure A.5 - *Matlab* and *Python* output matrices comparison: areas sum for each class of regions on solar disk 6^{th} ring. Time period of October 2014.

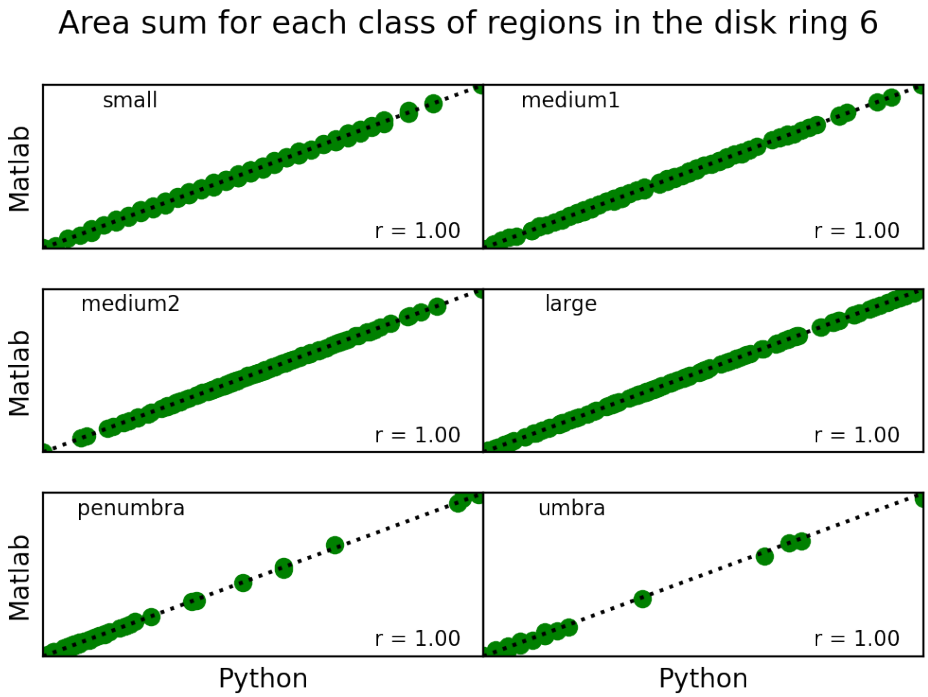


Figure A.6 - *Matlab* and *Python* output matrices comparison: areas sum for each class of regions on solar disk 7^{th} ring. Time period of October 2014.

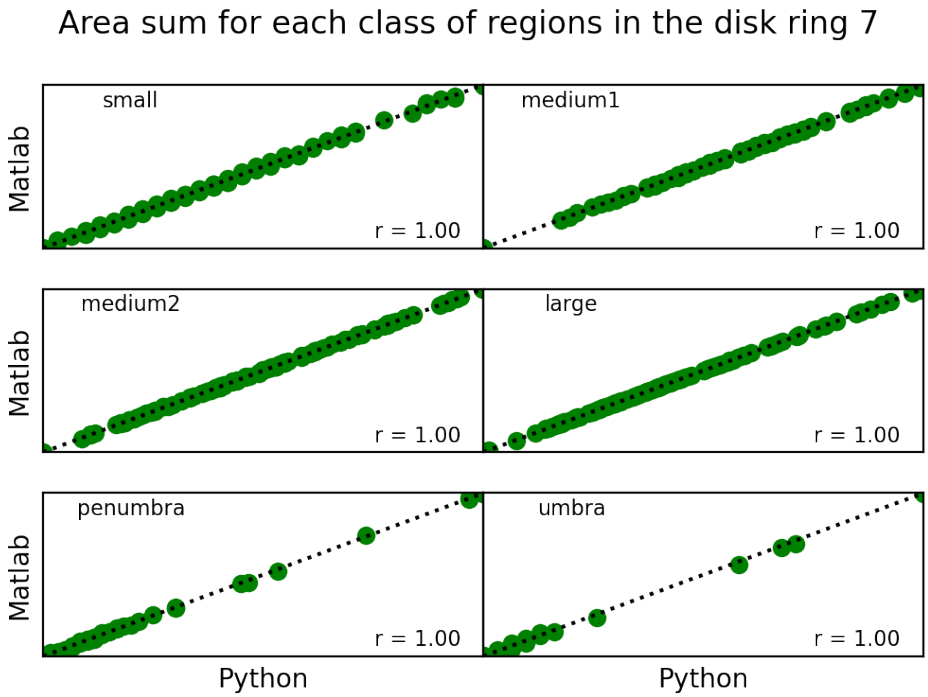


Figure A.7 - *Matlab* and *Python* output matrices comparison: areas sum for each class of regions on solar disk 8^{th} ring. Time period of October 2014.

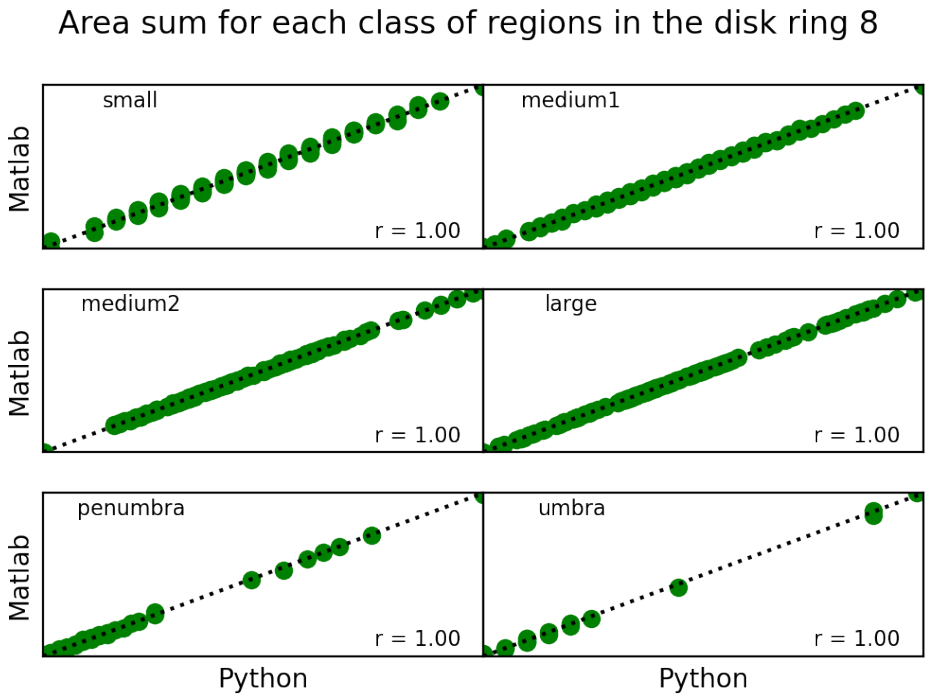


Figure A.8 - *Matlab* and *Python* output matrices comparison: areas sum for each class of regions on solar disk 9^{th} ring. Time period of October 2014.

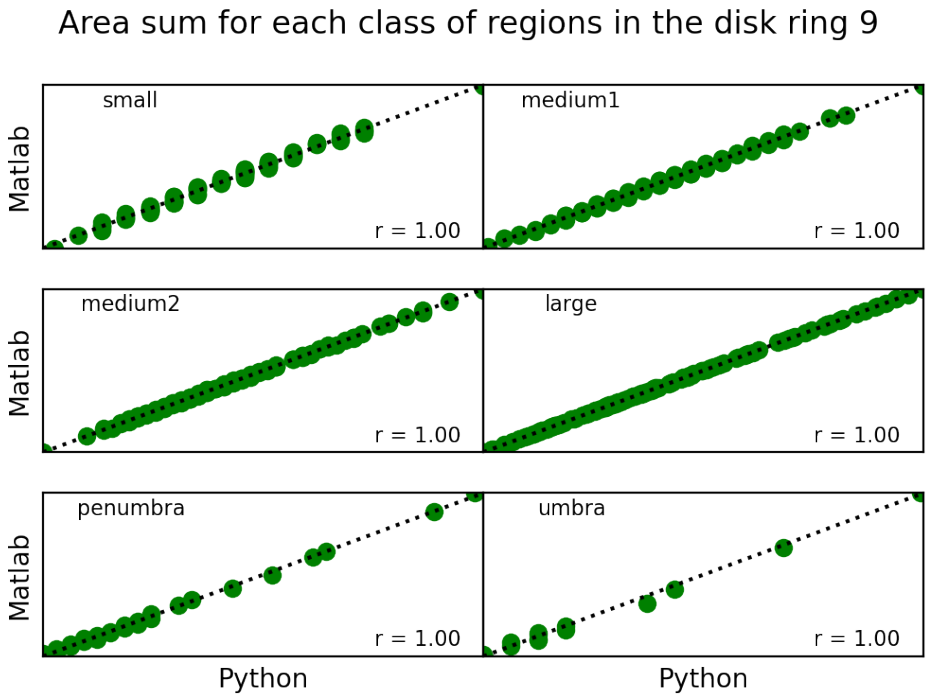


Figure A.9 - *Matlab* and *Python* output matrices comparison: areas sum for each class of regions on solar disk 10^{th} ring. Time period of October 2014.

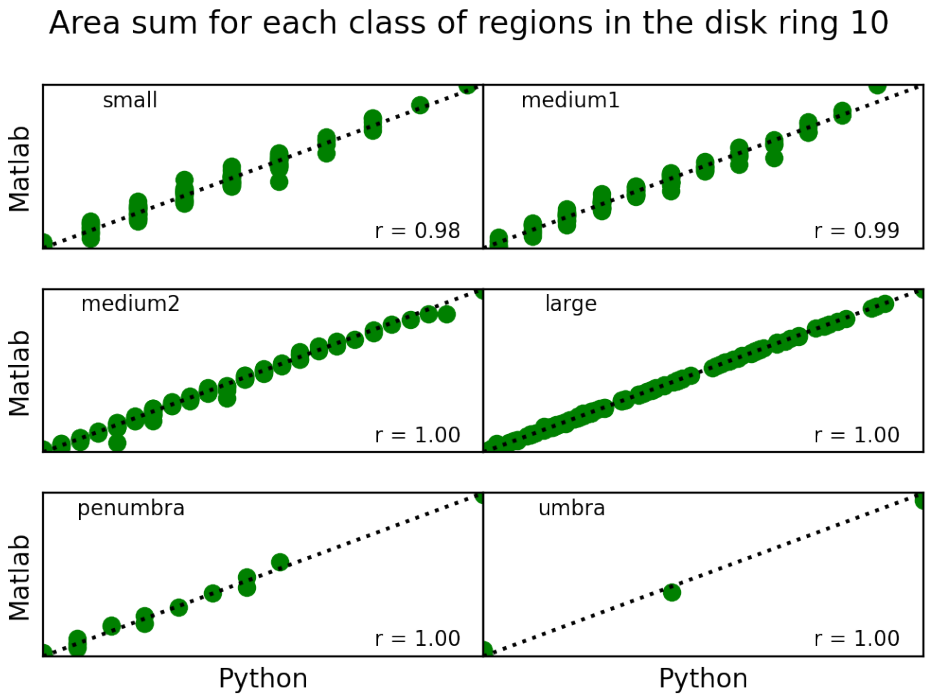


Figure A.10 - TSI 12 hours forecast test performance with LSTM net.

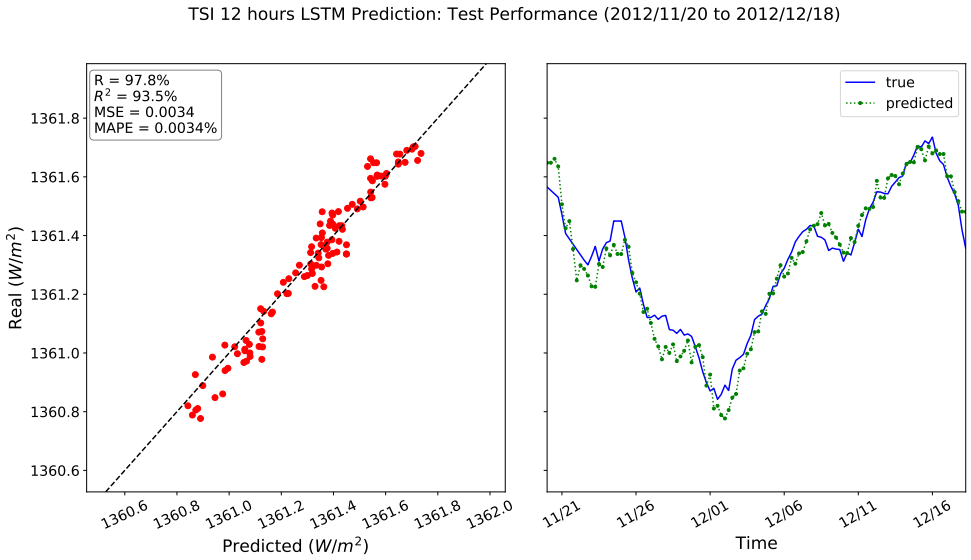


Figure A.11 - TSI 12 hours forecast test performance with GRU net.

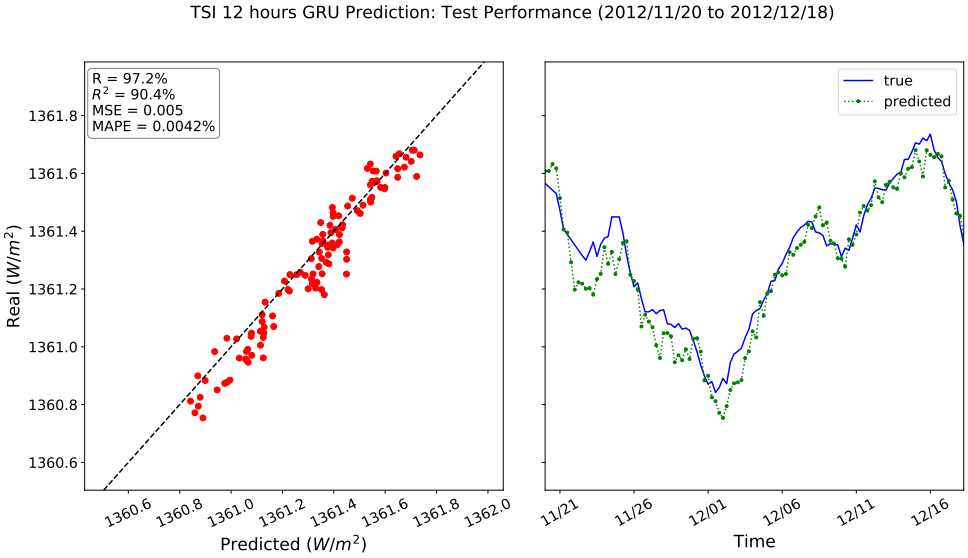


Figure A.12 - TSI 18 hours forecast test performance with LSTM net.

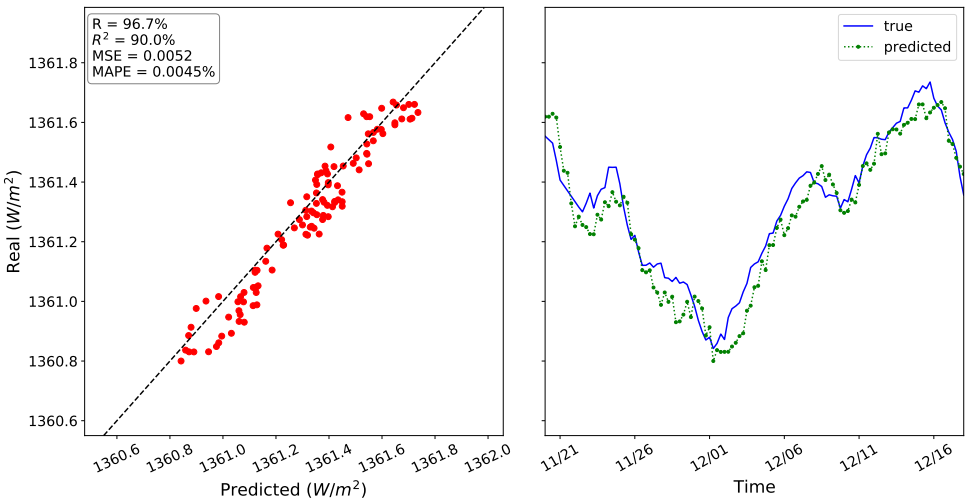


Figure A.13 - TSI 18 hours forecast test performance with GRU net.

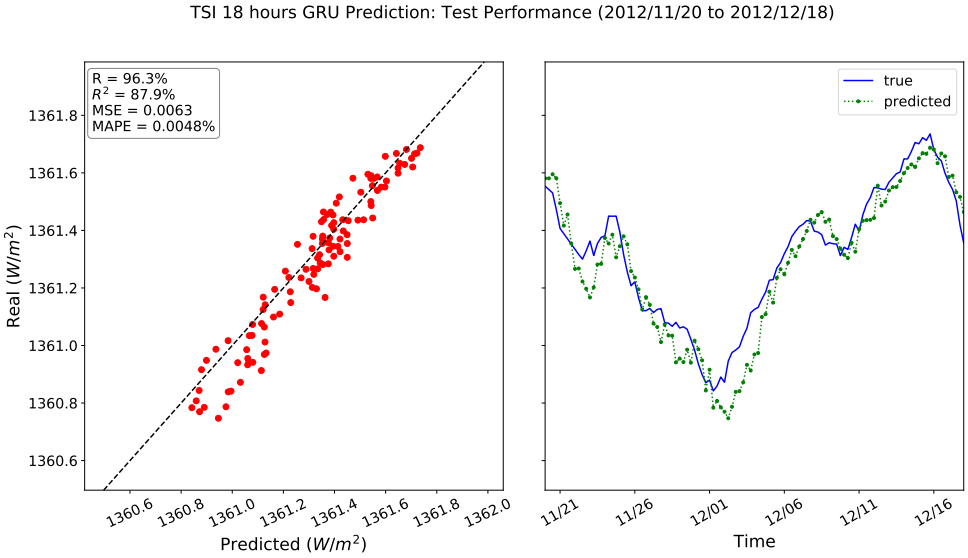


Figure A.14 - TSI 24 hours forecast test performance with LSTM net.

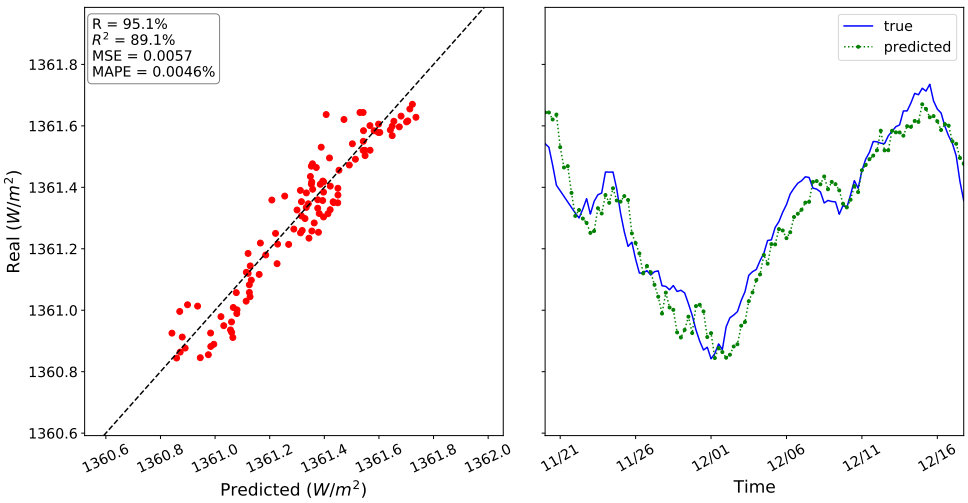


Figure A.15 - TSI 24 hours forecast test performance with GRU net.

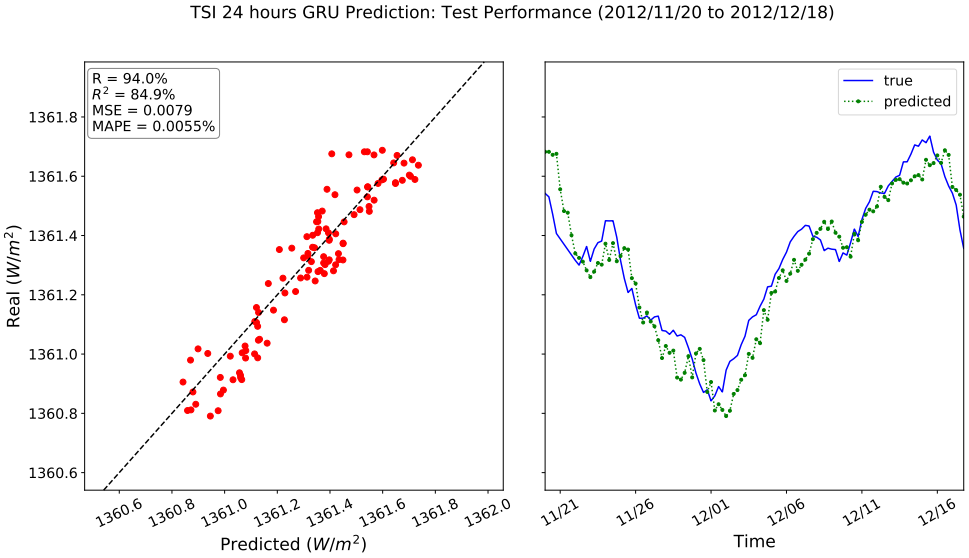


Figure A.16 - TSI 48 hours forecast test performance with LSTM net.

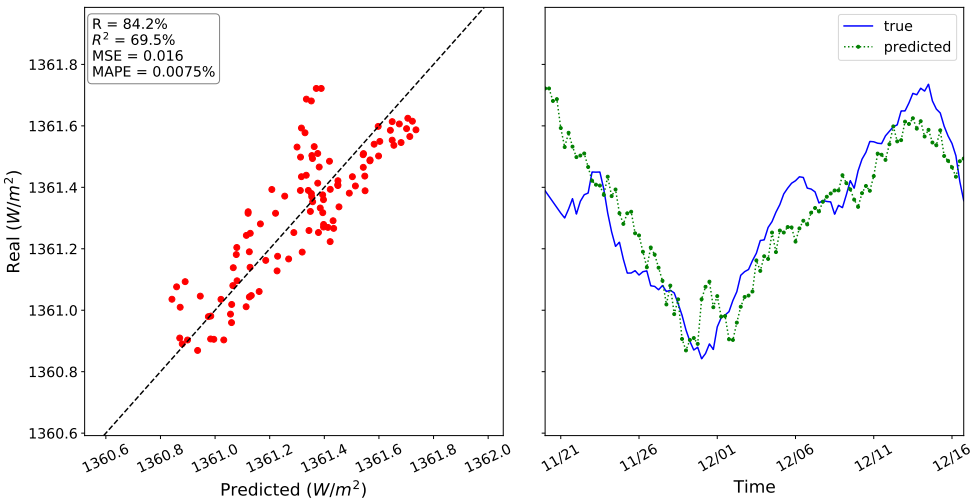


Figure A.17 - TSI 48 hours forecast test performance with GRU net.

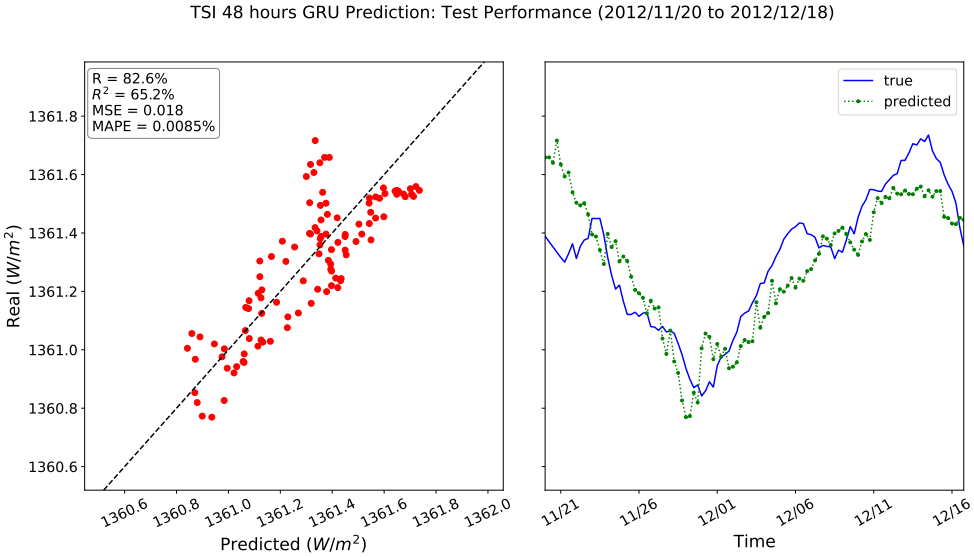


Figure A.18 - TSI 72 hours forecast test performance with LSTM net.

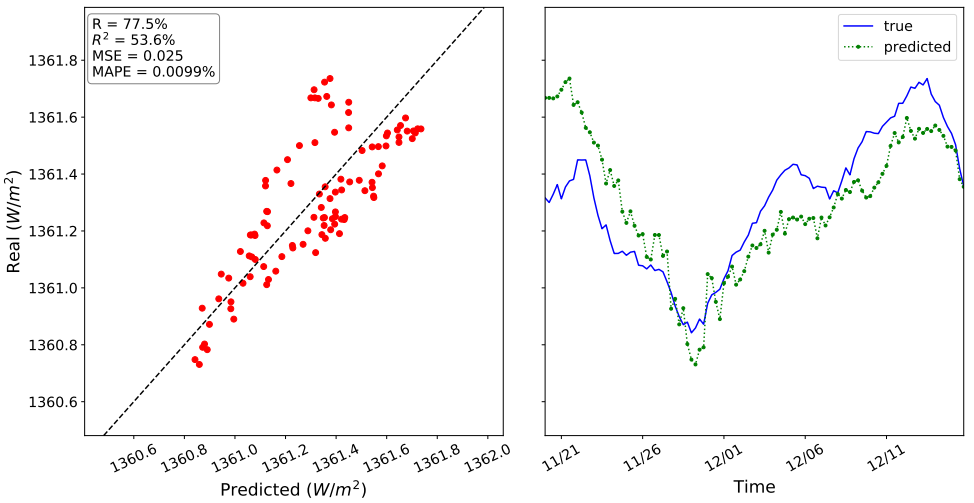


Figure A.19 - TSI 72 hours forecast test performance with GRU net.

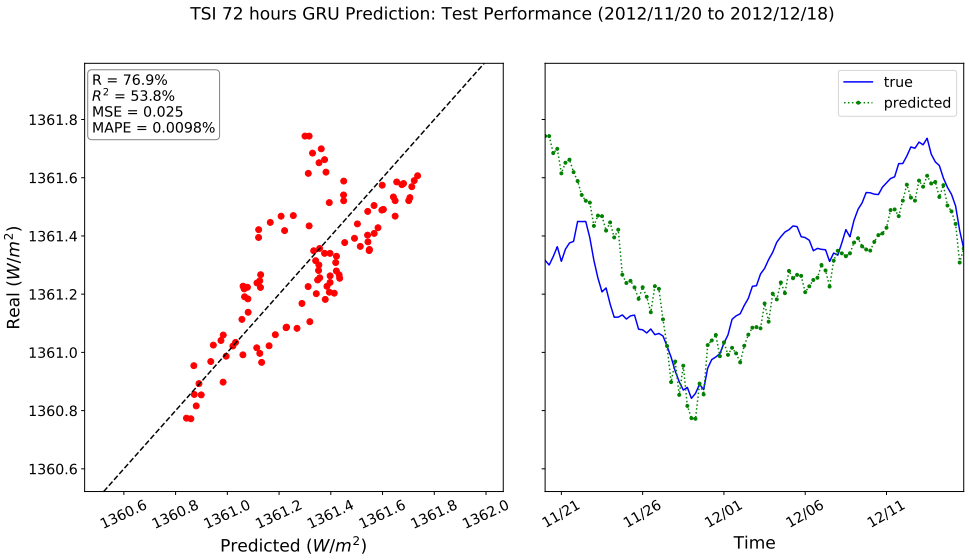


Figure A.20 - 30.5nm one day forecast training and validation performance with GRU.

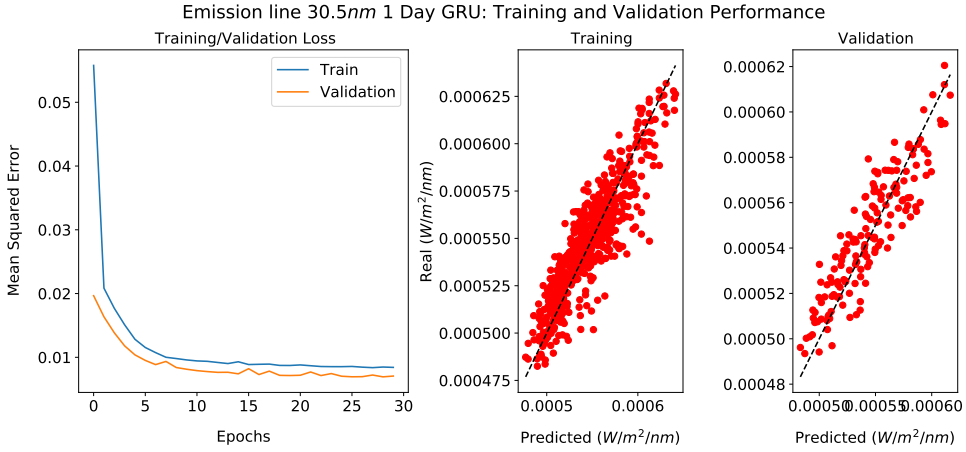


Figure A.21 - 30.5nm one day forecast test performance with GRU.

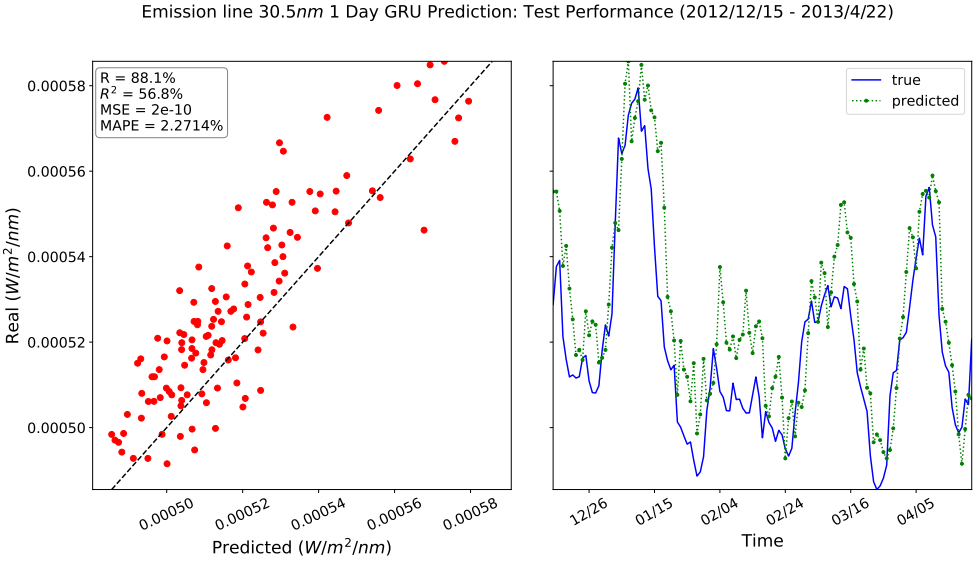


Figure A.22 - 30.5nm two days forecast training and validation performance with GRU.

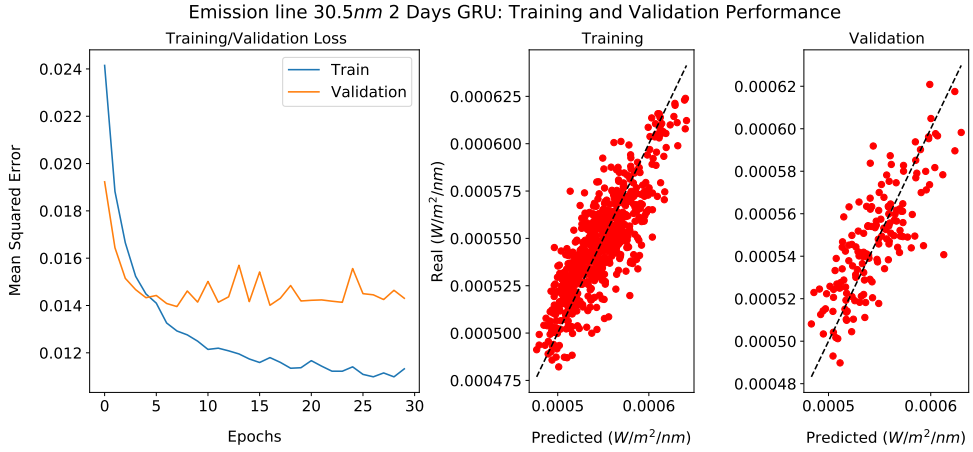


Figure A.23 - 30.5nm two days forecast test performance with GRU.

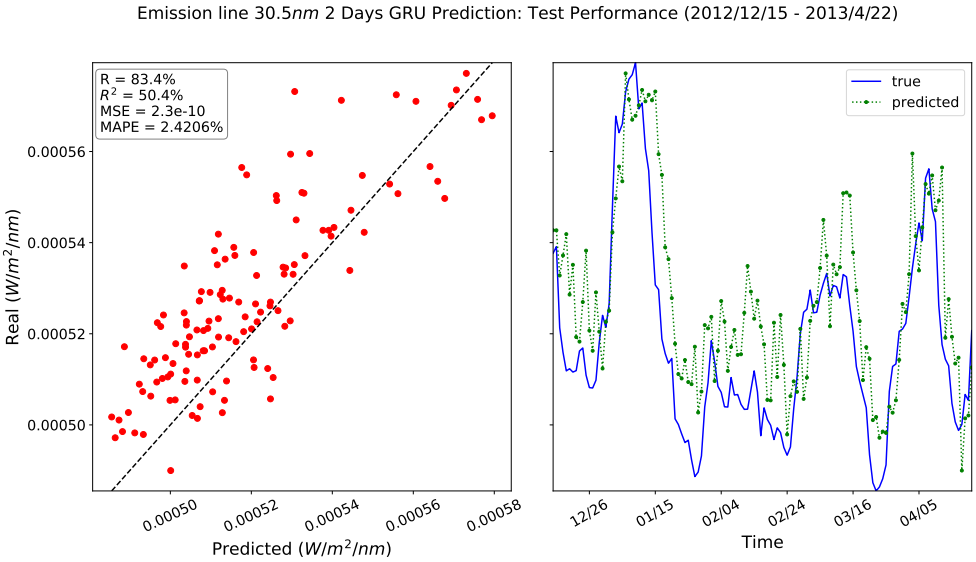


Figure A.24 - 30.5nm three days forecast training and validation performance with GRU.

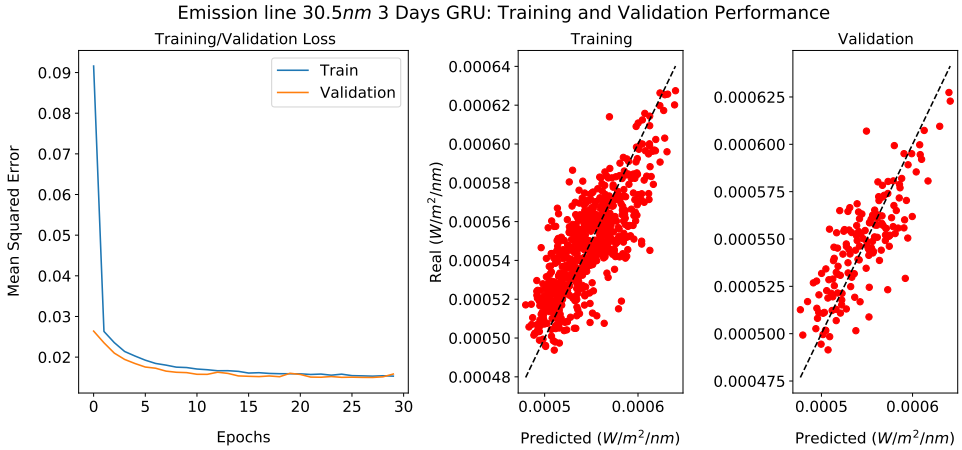


Figure A.25 - 30.5nm three days forecast test performance with GRU.

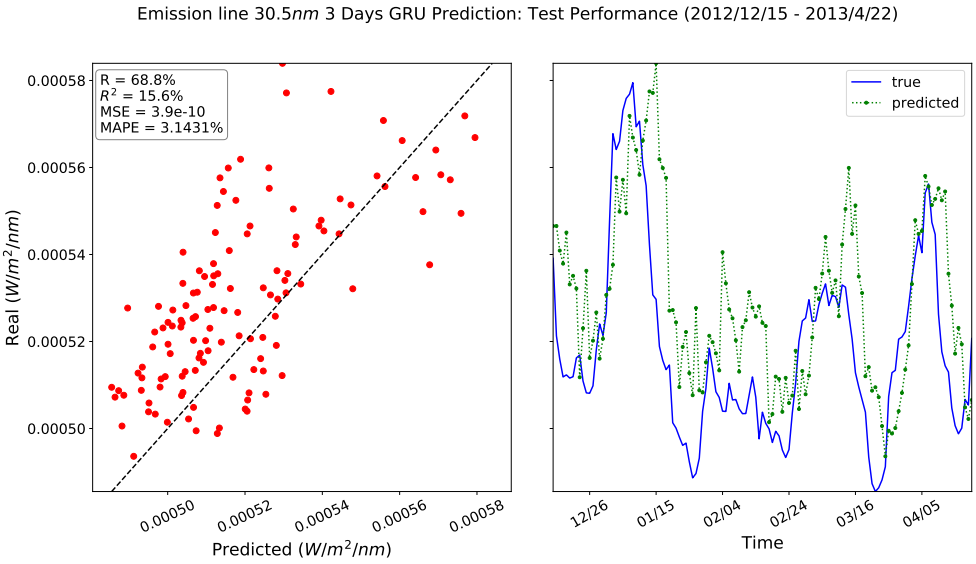


Figure A.26 - 48.5nm one day forecast training and validation performance with GRU.

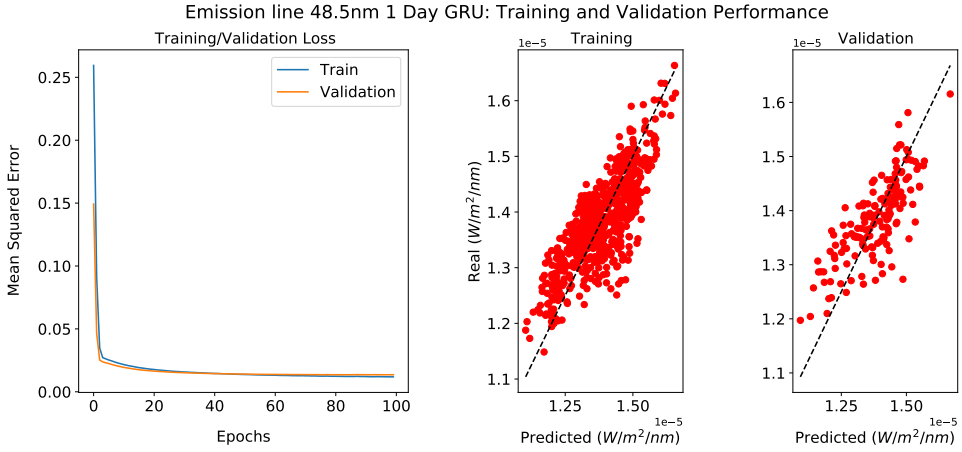


Figure A.27 - 48.5nm one day forecast test performance with GRU.

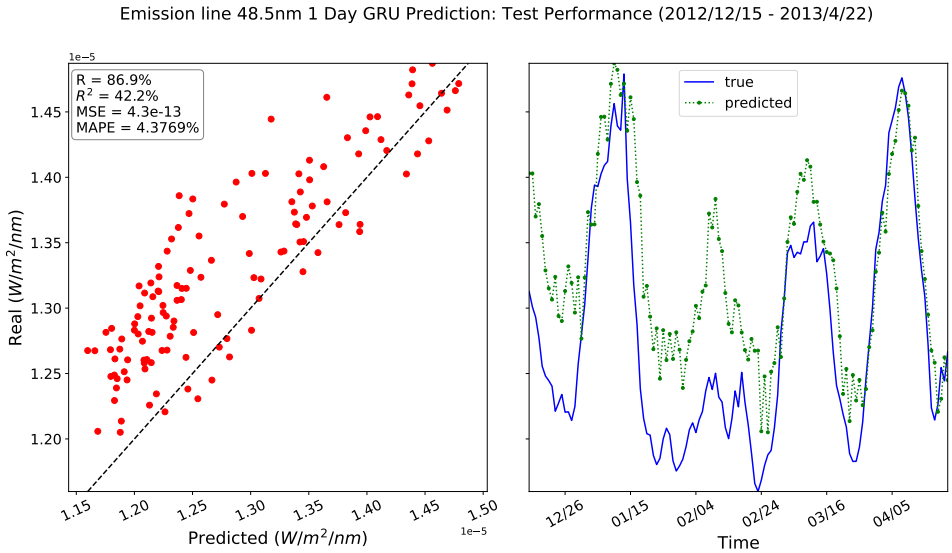


Figure A.28 - 48.5nm two days forecast training and validation performance with GRU.

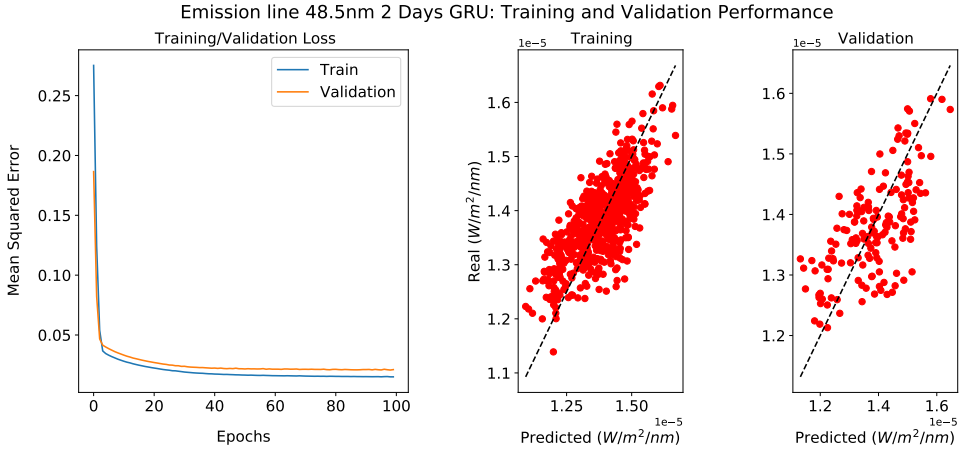


Figure A.29 - 48.5nm two days forecast test performance with GRU.

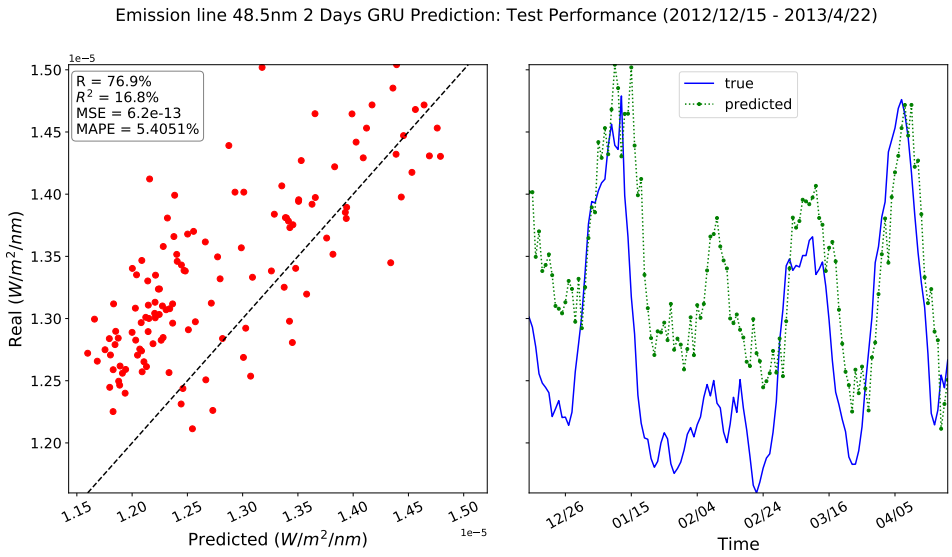


Figure A.30 - 48.5nm three days forecast training and validation performance with GRU.

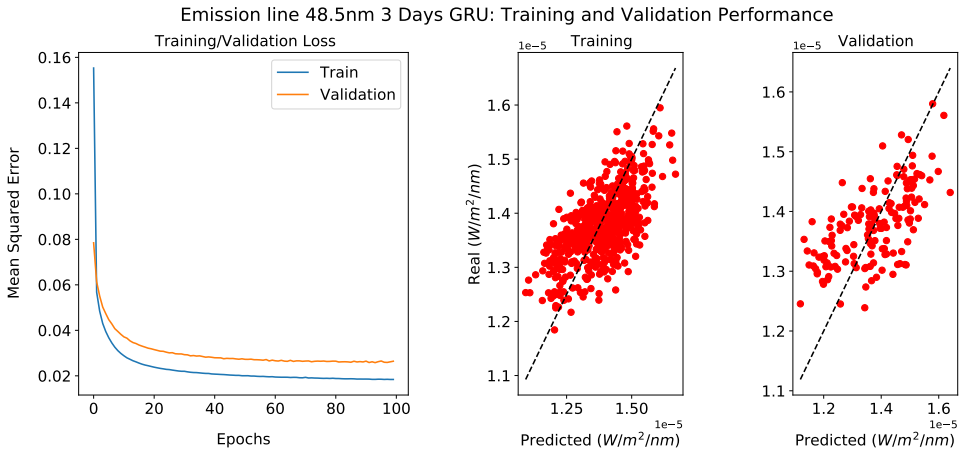


Figure A.31 - 48.5nm three days forecast test performance with GRU.

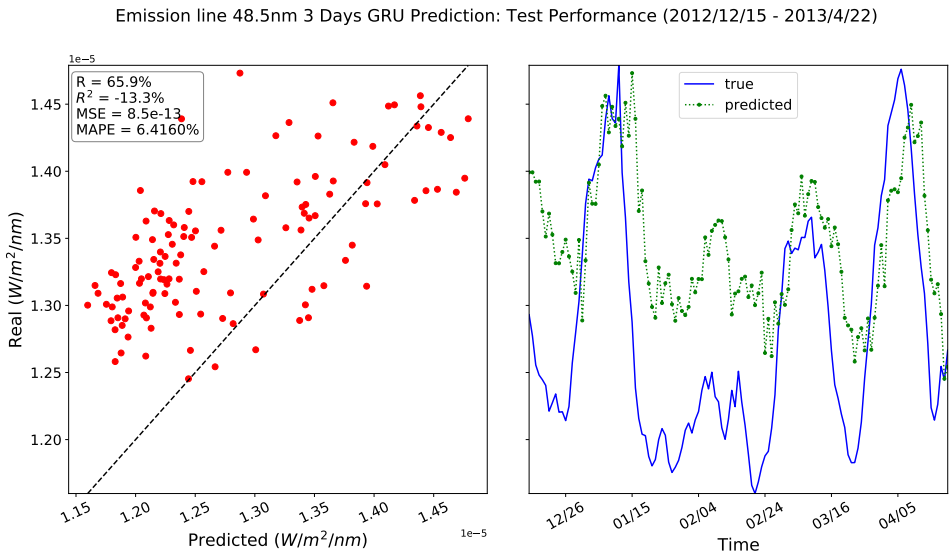


Figure A.32 - 121.5nm one day forecast training and validation performance with GRU.

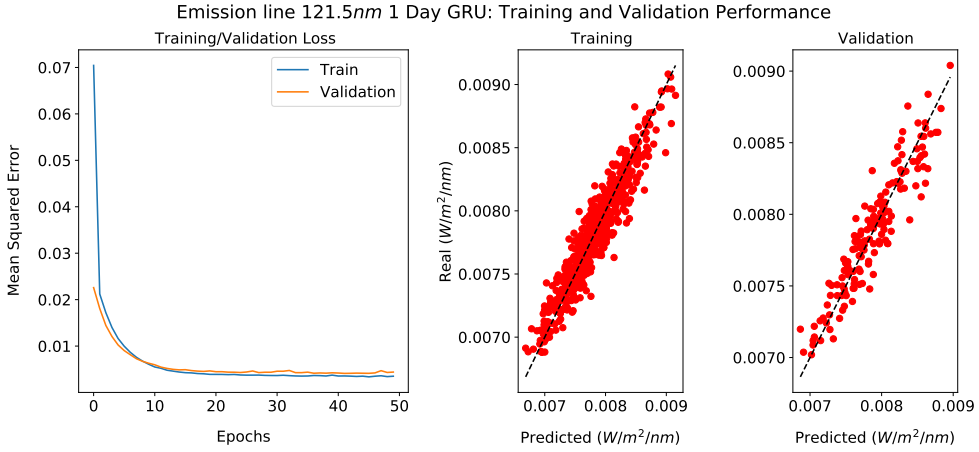


Figure A.33 - 121.5nm one day forecast test performance with GRU.

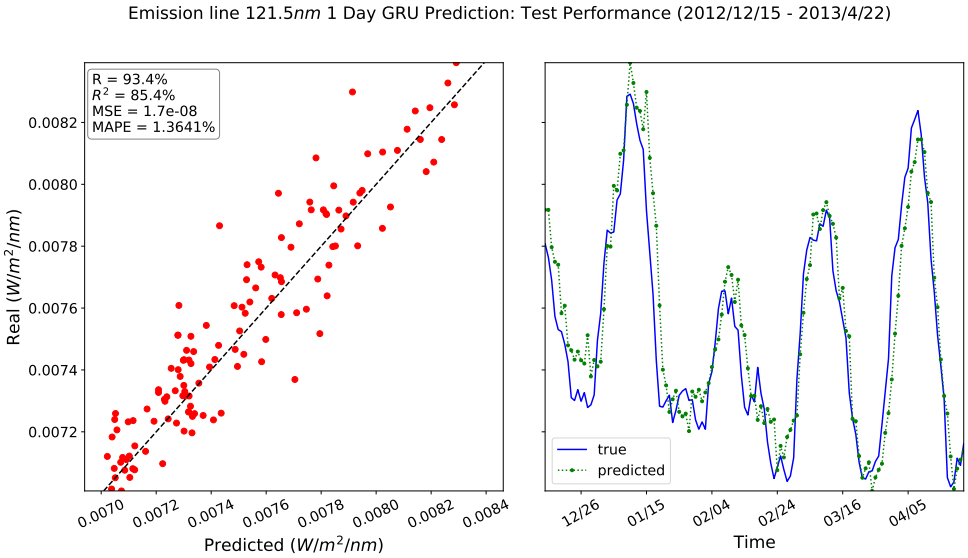


Figure A.34 - 121.5nm two days forecast training and validation performance with GRU.

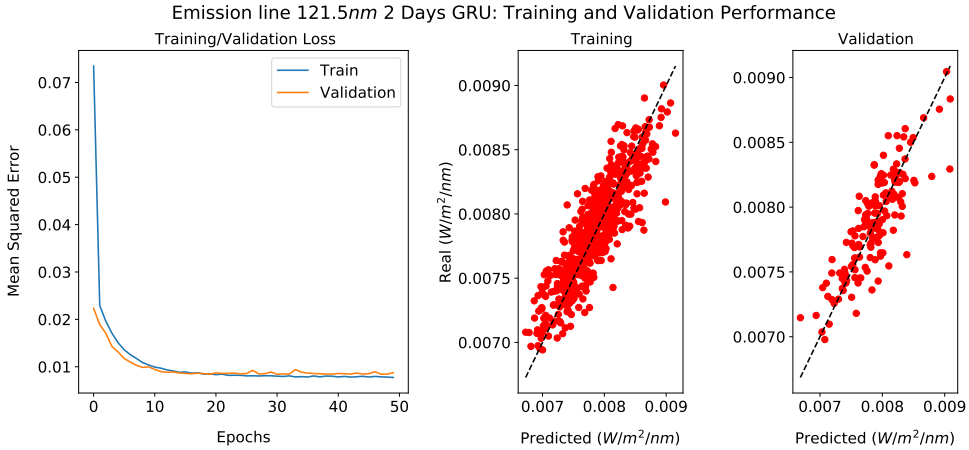


Figure A.35 - 121.5nm two days forecast test performance with GRU.

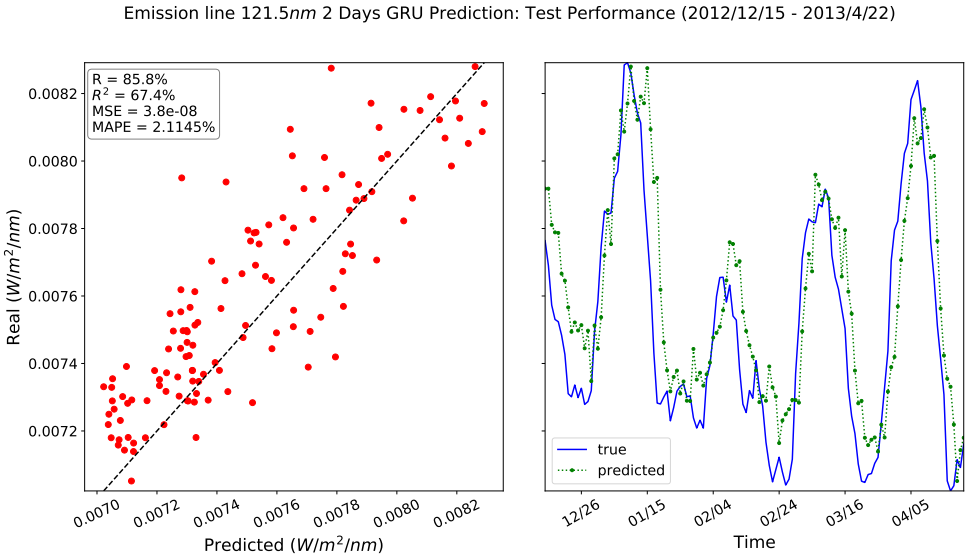


Figure A.36 - 121.5nm three days forecast training and validation performance with GRU.

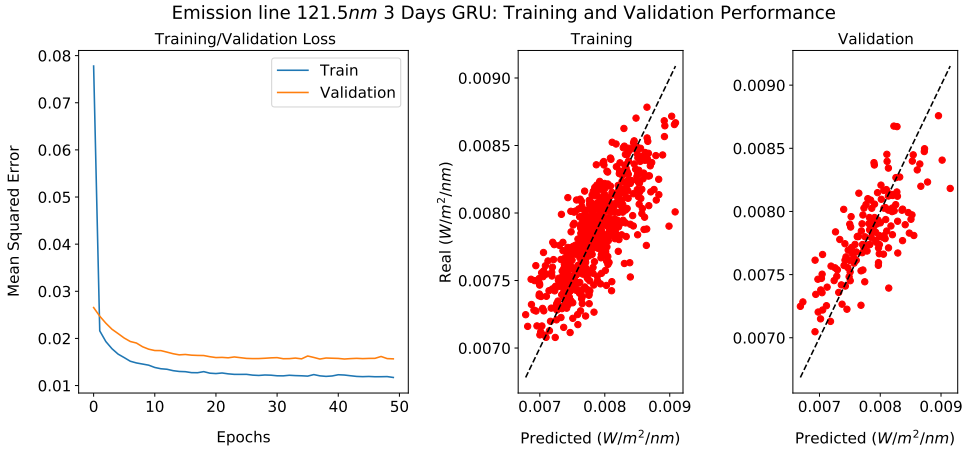
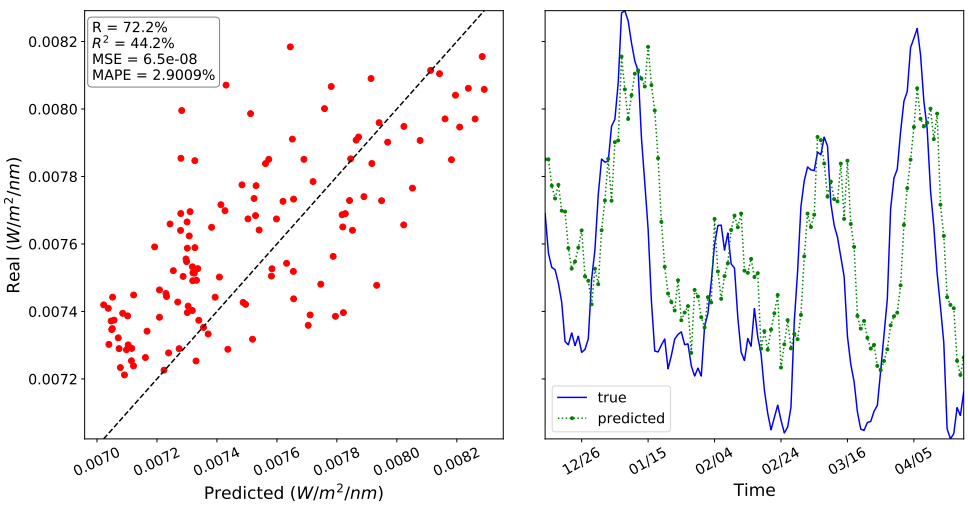


Figure A.37 - 121.5nm three days forecast test performance with GRU.



PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE

Teses e Dissertações (TDI)

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

Notas Técnico-Científicas (NTC)

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programas de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

Propostas e Relatórios de Projetos (PRP)

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

Publicações Seriadas

São os seriados técnicocientíficos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Constam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

Pré-publicações (PRE)

Todos os artigos publicados em periódicos, anais e como capítulos de livros.

Manuais Técnicos (MAN)

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

Relatórios de Pesquisa(RPQ)

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

Publicações Didáticas (PUD)

Incluem apostilas, notas de aula e manuais didáticos.

Programas de Computador (PDC)

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. Aceitam-se tanto programas fonte quanto os executáveis.