ABSTRACT

MACHINE LEARNING-BASED DATA ANALYTICS FOR UNDERSTANDING SPACE WEATHER AND CLIMATE

by
Yasser Abduallah

This dissertation addresses multiple crucial problems in space weather and climate, presenting new machine learning-based data analytics algorithms and models for tackling the problems.

First, the dissertation presents two new approaches to predicting solar flares. One approach, called DeepSun, predicts solar flares by utilizing a machine-learning-as-a-service (MLaaS) platform. The DeepSun system provides a friendly interface for Web users and an application programming interface (API) for remote programming users. It adopts an ensemble learning method that employs several machine learning algorithms to perform multiclass flare prediction. The other approach, named SolarFlareNet, forecasts the occurrence of solar flares within the next 24 to 72 hours by using a deep learning-based transformer model. This model is implemented into a fully operational near real-time flare forecasting system accessible on the Web.

Second, the dissertation presents a deep learning method, specifically a bidirectional long short-term memory (biLSTM) network, to predict if a solar active region (AR) would produce a solar energetic particle (SEP) event given that (i) the AR will produce an M- or X-class flare and a coronal mass ejection (CME) associated with the flare, or (ii) the AR will produce an M- or X-class flare regardless of whether or not the flare is associated with a CME. Experimental results demonstrate the superiority of the biLSTM network over related machine learning algorithms and its feasibility for SEP prediction.

Third, the dissertation presents multiple algorithms and models to forecast geomagnetic indices, which are used by geospace scientists to measure space storms

and their activities. The algorithms and models include a graph neural network combined with bidirectional long short-term memory for predicting the SYM-H index, a transformer-based model for predicting the Kp index, and a hybrid model combining multi-head attention layers and long short-term memory with a convolutional neural network for predicting the disturbance storm time (Dst) index. These algorithms and models incorporate Bayesian inference into their learning frameworks, capable of quantifying both aleatoric (data) uncertainty and epistemic (model) uncertainty when predicting future indices.

Finally, the dissertation presents a method, named TSInet, to reconstruct total solar irradiance (TSI). A minor change in solar irradiance can have a significant impact on the Earth's climate and atmosphere. As a result, studying and measuring solar irradiance is crucial in understanding climate change and solar variability. TSInet reconstructs total solar irradiance by leveraging deep learning for short and long periods of time that span beyond the current physical models' data availability. It can be used to reconstruct TSI for more than 9,000 years.

All the algorithms and models presented in this dissertation are implemented into open-source software tools using Jupyter notebooks with GitHub, which are publicly available on the Web. These tools are Binder enabled and have Zenodo archive for download. The tools are integrated into a machine learning (ML) enhanced cyberinfrastructure that contains ML software and databases for advancing space weather research and education.

MACHINE LEARNING-BASED DATA ANALYTICS FOR UNDERSTANDING SPACE WEATHER AND CLIMATE

by
Yasser Abduallah

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Computer Science

Department of Computer Science

December 2022

APPROVAL PAGE

MACHINE LEARNING-BASED DATA ANALYTICS FOR UNDERSTANDING SPACE WEATHER AND CLIMATE

Yasser Abduallah

Dr. Jason T. L. Wang, Dissertation Advisor — Date
Professor of Computer Science, NJIT

Dr. Ali Mili, Committee Member — Date
Professor of Computer Science, NJIT

Dr. Ioannis Koutis, Committee Member — Date
Associate Professor of Computer Science, NJIT

Dr. Chase Wu, Committee Member — Date
Professor of Data Science, NJIT

Dr. Haimin Wang, Committee Member — Date
Distinguished Professor of Physics, NJIT

Dr. Wynne Hsu, Committee Member — Date
Provost's Chair Professor of Computer Science, National University of Singapore, Singapore

BIOGRAPHICAL SKETCH

Author: Yasser Abduallah

Degree: Doctor of Philosophy

Date: December 2022

Undergraduate and Graduate Education:

- Doctor of Philosophy in Computer Science,
  New Jersey Institute of Technology, Newark, NJ, 2022
- Master of Science in Computer Science,
  New Jersey Institute of Technology, Newark, NJ, 2014
- Bachelor of Engineering in Computer Engineering,
  City College of New York, New York, NY, 2011

Major: Computer Science

Presentations and Publications:

Y. Abduallah, J. T. L. Wang, C. Xu, and H. Wang. "A Transformer-Based Framework for Geomagnetic Activity Prediction". In *Proceedings of the Foundations of Intelligent Systems: 26th International Symposium on Methodologies for Intelligent Systems, Cosenza, Italy*, 325–335, Heidelberg, Berlin: Springer-Verlag, Oct. 2022.

K. A. Alobaid, Y. Abduallah, J. T. L. Wang, H. Wang, H. Jiang, Y. Xu, V. Yurchyshyn, H. Zhang, H. Cavus, and J. Jing. "Predicting CME Arrival Time Through Data Integration and Ensemble Learning". *Frontiers in Astronomy and Space Sciences*, 9:1013345, Oct. 2022.

Y. Abduallah, V. K. Jordanova, H. Liu, Q. Li, J. T. L. Wang, and H. Wang. "Predicting Solar Energetic Particles Using SDO/HMI Vector Magnetic Data Products and a Bidirectional LSTM Network". *The Astrophysical Journal Supplement Series*, 260(1):16, May 2022.

Y. Abduallah, J. T. L. Wang, P. Bose, G. Zhang, F. Gerges, and H. Wang. "Forecasting the Disturbance Storm Time Index with Bayesian Deep Learning". In *Proceedings of the 35th International Florida Artificial Intelligence Research Society Conference*, vol. 35, Hutchinson Island, Jensen Beach, Florida, May 2022.

Y. Abduallah, J. T. L. Wang, Y. Nie, C. Liu, and H. Wang. "DeepSun: Machine-Learning-as-a-Service for Solar Flare Prediction". *Research in Astronomy and Astrophysics*, 21(7):160, 2021.

Y. Abduallah, J.T.L. Wang, Y. Shen, K. Alobaid, S. Criscuoli and H. Wang, "Reconstruction of Total Solar Irradiance by Deep Learning," In *Proceedings of the 34th International Florida Artificial Intelligence Research Society Conference*, vol. 34, North Miami Beach, Florida, May 2021.

Y. Abduallah and J.T.L. Wang, "New Algorithms for Inferring Gene Regulatory Networks from Time-Series Expression Data on Apache Spark," *International Journal of Big Data Intelligence*, vol. 6(3&4), pp 153-162, 2019.

Y. Abduallah, T. Turki, K. Byron, Z. Du, M. Cervantes-Cervantes and J.T.L. Wang, "MapReduce Algorithms for Inferring Gene Regulatory Networks from Time-Series Microarray Data Using an Information-Theoretic Approach," *BioMed Research International*, vol. 2017:6261802, 2017.

Y. Abduallah and J.T.L. Wang, "A Time-Delayed Information-Theoretic Approach to the Reverse Engineering of Gene Regulatory Networks Using Apache Spark," In *Proceedings of the 15th IEEE International Conference on Dependable, Autonomic and Secure Computing, the 15th International Conference on Pervasive Intelligence and Computing, and the 3rd International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress*, pp 1106-1113, Orlando, FL, Nov. 2017.

H. Wang, H. Jiang, Y. Abduallah, and J.T.L Wang, "Create an Advanced Solar Active Region Database and Predict Solar Flares Using Machine Learning Tools," *American Geophysical Union Fall Meeting*, Chicago, Illinois, May 2021, Poster Presentation.

Y. Abduallah, J.T.L. Wang, H. Wang, G. Nita, A. Kosovichev, V. Oria and H. Jiang, "A Cyberinfrastructure for Advancing Space Weather Research and Education," *The 2022 EarthCube Annual Meeting*, La Jolla, California, June 2022, Poster Presentation.

Y. Abduallah, J.T.L. Wang and H. Wang, "Predicting Solar Flares with Machine Learning," *The 2022 EarthCube Annual Meeting*, La Jolla, California, June 2022, Oral Presentation.

Y. Abduallah, J.T.L. Wang and H. Wang, "SolarDB: A Cyberinfrastructure for Advancing Space Weather Research," *The 2022 EarthCube Annual Meeting*, La Jolla, California, June 2022, Oral and Demo Presentation.

Y. Abduallah, J.T.L. Wang, K. Alobaid, S. Criscuoli and H. Wang, "Deep Learning Based Reconstruction of Total Solar Irradiance," *Applications of Statistical Methods and Machine Learning in the Space Sciences (SSI)*, Boulder, Colorado, May 2021, Poster Presentation.

I would like to dedicate this work to my family. My wife, Wafaa, who has been a constant extraordinary source of support and encouragement throughout the challenges of graduate and undergraduate studies as well as our life together. I am very gratefully and appreciative for having you in my life. I would also like to dedicate this work to and thank my sons, Abdulrahman, Abdulaziz, and Abdulmalek for their complete support and patience throughout my journey. You made me better than I ever could imagine.

ACKNOWLEDGMENT

First and foremost, I would also like to extend my appreciation and warmest thanks to my advisor, Dr. Jason T. L. Wang, for allowing me to conduct and perform research under his supervision. I am very grateful for him allowing me to pursue my dream by putting confidence in me, giving me all the support I needed, and the freedom to accomplish my tasks. As a part-time student and full-time professional, it seemed impossible to come to this moment, but with Dr. Wang's support and encouragement, it became a reality. Thank you, Dr. Wang, for being an excellent mentor, role model, teacher, and researcher.

Second, I would like to express my deepest gratitude to my dissertation committee: Dr. Ali Mili, Dr. Ioannis Koutis, Dr. Chase Wu, Dr. Haimin Wang, and Dr. Wynne Hsu for offering their invaluable time, guidance, and support throughout the process of reviewing and completing my dissertation. Your feedback, discussion, and ideas have been invaluable.

I would also like to express my deepest gratitude to the Department of Computer Science for providing a tremendous collaborative PhD program for professional students like me. It would be almost impossible for me to complete the PhD degree without the full support of the CS department faculty and staff. The research projects in my dissertation were supported by the U.S. National Science Foundation under Grant Nos. AGS-1927578, AGS-2149748, AGS-2228996 and CNS-2206886.

Furthermore, I would like to thank my lab-mates Dr. Hao Liu, Dr. Haodi Jiang, PhD candidate Khalid A. Alobaid, and the Department of Physics collaborators: Dr. Qin Li, Dr. Jiasheng Wang and PhD candidate Nian Liu, for their extraordinary contributions and collaborations in my dissertation work.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF TABLES
(Continued)

Table **Page**

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

Solar physics phenomenons such as solar eruption, energetic particles, geomagnetic storms, solar irradiance have significant effects on Earth and human's life, spacecraft, electrical power grids, observers of the aurora, and navigation systems and more. Therefore, predicting and forecasting the activities of these phenomenons draw a lot of attention to researchers and scientists. Early and accurate prediction of these activities are essential for preparedness and disaster risk management.

Machine learning (ML) is an analytical method that gives computer programs the ability to learn from data and progressively improve performance. It uses input data, also called training data, and learns hidden insights in the training data to build a predictive model that will be used later to make predictions on unseen test data. Machine learning has been popular in predictive analytics for many years.

Deep learning (DL) is a branch of machine learning methods and algorithms focusing on the use of deep neural networks, to enhance the learning outcome. It is inspired by the human brain functions and improves the artificial intelligence (AI) process to learn new patterns and things for decisions making. DL has drawn significant interest in recent years. It has been used extensively in many applications and domains such as biomedical, finance, weather forecasting and more.

This dissertation leverages deep learning and presents novels methods and algorithms to tackle multiple crucial problems in solar physics domain focusing in solar weather and solar climate. Specifically, in solar weather, it addresses flares prediction, solar energetic particles and geomagnetic activities: disturbance storm time (Dst) index and the planetary index (Kp). In solar climate, it addresses the solar irradiance construction. In addition, the dissertation provides software development

tools that can be used in space weather and space climate research and operational systems.

First, the dissertation presents tow algorithms for solar flare predictions. Solar flare prediction plays an important role in understanding and forecasting space weather. The main goal of the Helioseismic and Magnetic Imager (HMI), one of the instruments on NASA's Solar Dynamics Observatory, is to study the origin of solar variability and characterize the Sun's magnetic activity. HMI provides continuous full-disk observations of the solar vector magnetic field with high cadence data that lead to reliable predictive capability; yet, solar flare prediction effort utilizing these data is still limited. Specifically, we construct training data by utilizing the physical parameters provided by the Space-weather HMI Active Region Patches (SHARP) and categorize solar flares into four classes, namely B, C, M, X, according to the X-ray flare catalogs available at the National Centers for Environmental Information (NCEI). The first algorithm is a framework called DeepSun. It is the first web based machine-learning-as-a-service (MLaaS) framework that is capable of predicting solar flares through the internet. The DeepSun system employs several machine learning algorithms to tackle this multi-class prediction problem and provides an application programming interface (API) for remote programming users. The second algorithm is a hybrid-transformer that combines transform encoder, called SolarFlareNet, convolutional neural network, and long short-term memory components to predict if the given SHARP parameters will produce a flare within the next 24 hours.

Second, the dissertation tackles the solar energetic particles (SEPs). SEPS are an essential source of space radiation, which are hazards for humans in space, spacecraft, and technology in general. The dissertation presents a deep learning method, specifically a bidirectional long short-term memory (biLSTM) network, to predict if an active region (AR) would produce an SEP event given that (i) the AR will produce an M- or X-class flare and a coronal mass ejection (CME) associated with

the flare, or (ii) the AR will produce an M- or X-class flare regardless of whether or not the flare is associated with a CME. The data samples used here are collected from the *Geostationary Operational Environmental Satellite*'s X-ray flare catalogs provided by the National Centers for Environmental Information. We select M- and X-class flares with identified ARs in the catalogs for the period between 2010 and 2021, and find the associations of flares, CMEs and SEPs in the Space Weather Database of Notifications, Knowledge, Information during the same period. Each data sample contains physical parameters collected from the Helioseismic and Magnetic Imager on board the *Solar Dynamics Observatory*. The dissertation also discusses extensions of the approach for probabilistic forecasting and calibration with empirical evaluation.

Third, the dissertation presents multiple algorithms to forecast geomagnetic indices. Geomagnetic indices are used by Geospace scientists to measure space storms and their activities. The first index is one of the most important geomagnetic indices, the SYM-H index. At midlatitude, the SYM-H index indicates the longitudinally symmetric geomagnetic disturbance of the horizontal component of the magnetic field. Early and accurate estimation of the SYM-H index is crucial for disaster risk management and preparedness. We propose a novel deep learning framework, named SYMHNet, that employs a graph neural network and a bidirectional long short-term memory network to cooperatively learn patterns from solar wind data for short-term forecasts of the SYM-H index. SYMHNet takes as input a time series of solar wind parameters and predicts as output SYM-H index values for the next t hours where t ranges from 1 to 6. NASA's Space Science Data Coordinated Archive provides the solar wind parameters used here. Similarly, the dissertation presents another two novel algorithms to forecast two more indexes: the interplanetary index called the Kp Index and the disturbance storm time (Dst) index. The three models have an additional add-on capability that integrates the Bayesian inference into the learning

framework, that is able to quantify both aleatoric (data) uncertainty and epistemic (model) uncertainty when predicting future indices.

Fourth, the dissertation presents the first deep learning method in space climate, called TSInet, to reconstruct total solar irradiance (TSI). The Earth's primary source of energy is the radiant energy generated by the Sun, which is referred to as solar irradiance, or total solar irradiance (TSI) when all of the radiation is measured. A minor change in the solar irradiance can have a significant impact on the Earth's climate and atmosphere. As a result, studying and measuring solar irradiance is crucial in understanding climate changes and solar variability. Several methods have been developed to reconstruct total solar irradiance for long and short periods of time; however, they are physics-based and rely on the availability of data, which does not go beyond 9,000 years. The TSInet reconstructs total solar irradiance by deep learning for short and long periods of time that span beyond the physical models' data availability. On the data that are available, our method agrees well with the state-of-the-art physics-based reconstruction models. Therefore, it can be used as an add-on to physical models to reconstruct TSI. This is the first time that deep learning has been used to reconstruct total solar irradiance for more than 9,000 years.

Lastly, the dissertation presents the implementation of the deep learning tools developed for the algorithms presented in the dissertation. The tools are implemented using Jupyter notebooks with Github and publicly available to download and use. The tools include the TSInet for reconstructing total solar irradiance, DeepSun MLaaS for predicting solar flares, SEP biLSTM for predicting solar energetic particles, DSTT for predicting and forecasting disturbance storm time, and KpNet for forecasting the Kp index. The tools are Bender enabled and also have Zendo archive to download.

The dissertation is organized as follows. Chapter 2 presents the solar flare prediction algorithms: the machine learning-as-a-service (MLaaS), DeepSun, framework and the solare flare prediction SolarFlareNet. Chapter 3 details the

bidirection long short-term memory (biLSTM) to predict the solar energetic particles (SEPs). Chapter 4 presents the detailed implementation geomagnetic indices algorithms: SYMHNet, KpNet, and DSTT. Chapter 5 describes solar climate model, TSInet, for reconstructing the total solar irradiance. Finally chapter 6 present the software development tools that are useful for Space Weather and Space Climate researches and operational systems.

CHAPTER 2

PREDICTION OF SOLAR FLARES

2.1 Machine-Learning-as-a-Service for Flare Prediction

2.1.1 Background and Related Work

Solar flares and the often-associated coronal mass ejections (CMEs) highly impact the near-Earth space environment [117, 121]. They have the potential to cause catastrophic damage to technology infrastructure [53]. According to the U.S. National Space Weather Strategy, released by the Space Weather Prediction Center, it is a challenging task to correctly predict solar flares and CMEs. Recent efforts led by the United States and its partners resulted in substantial progress toward monitoring, prediction, and mitigation plans, but much more effort is still needed.

Researches have indicated that the magnetic free energy stored in the corona, quickly discharged by magnetic reconnection, powers solar flares and CMEs [137] . The process of building the coronal free energy is controlled by the structural evolution of the magnetic field on the photosphere where plasma dominates the process. Observing and measuring the structure and evolution of the photospheric magnetic field can provide valuable information and clues to the triggering mechanisms of flares and CMEs. There are many physical properties or parameters, as we will discuss later in this research, that characterize the static photospheric magnetic field, such as integrated Lorentz force, magnetic helicity injection, unsigned magnetic flux, vertical electric currents, magnetic shear and gradient, and magnetic energy dissipation.

Researchers spent significant efforts attempting to understand the physical relationship between flare productivity and non-potentiality of active regions (ARs) as specified by the physical parameters. This led researchers to use different methods to predict flares that are not based on physical models, but rather based on statistical

modeling and machine learning [18]. Machine learning gives computer programs the ability to learn from data and progressively improve performance. It uses input data, also called training data, and learns hidden insights in the training data to build a predictive model that will be used later to make predictions on unseen test data.

In our previous work [117], we reported the results of solar flare prediction using the random forests (RF) algorithm [33]. We constructed a database of solar flare events using the physical parameters provided by the Space-weather HMI Active Region Patches (SHARP), and categorized solar flares into four different classes, namely B, C, M, X, based on the X-ray flare catalogs available at the National Centers for Environmental Information (NCEI). Flares in the B class have the smallest magnitude while flares in the X class have the largest magnitude. We used the RF algorithm and the physical parameters or features to perform multi-class classification of solar flares, predicting the occurrence of a certain class of flares in a given active region (AR) within 24 hours. Our experimental results demonstrated the good performance of the RF algorithm.

In this research, we extend our previous work in [117] by considering two additional multi-class classification algorithms: multilayer perceptrons (MLP) and extreme learning machines (ELM). We implement these algorithms into a machine-learning-as-a-service (MLaaS) framework, called DeepSun, which allows scientists to perform multi-class flare prediction on the Internet. Our work here makes the following two contributions.

1. We develop an ensemble method for multi-class flare prediction that performs better than the existing machine learning algorithms including RF, MLP and ELM according to our experimental study.

2. We design and implement DeepSun, which is the first MLaaS system of its kind for solar flare prediction.

The rest of this work is organized as follows. Subsection 2.1.2 describes the data and the SHARP predictive parameters used in this study. Subsection 2.1.3 describes the

machine learning algorithms employed by DeepSun. Subsection 2.1.4 evaluates the performance of these machine learning algorithms. Subsection 2.1.5 details the design and implementation of the DeepSun framework. Subsection 2.1.7 surveys related work and compares DeepSun with existing services computing systems. Subsection 2.1.8 concludes the work and points out some directions for future research.

2.1.2 Data and Physical Parameters

In 2012, SHARP data were released. The main goal of the SHARP data was to facilitate AR (active region) event forecasting [31]. These data are available in the Joint Science Operations Center (JSOC)[1] as hmi.sharp series which include magnetic measures and parameters for many ARs. In 2014, another data series, cgem.Lorentz, were produced based on the SHARP data. This series include the Lorentz force estimations. The main goal of this series was to help diagnose the dynamic process of ARs. Bobra *et al.* [31] considered 25 physical parameters in the SHARP datasets that characterize the AR magnetic field properties. The authors used a univariate feature selection method to score the 25 parameters, and suggested that the top 13 out of the 25 parameters be used as predictors for flare activity. Table 2.1 summarizes these 13 important parameters and their descriptions. More details about the 13 physical parameters can be found in [117].

We constructed a database based on the SHARP parameters extracted from the solar images that are available at the Joint Science Operations Center (JSOC) and the X-ray flare catalogs provided by the National Centers for Environmental Information (NCEI) [117]. We considered the period between May 2010 and December 2016. There are 845 flares in this period, among which 128 flares are of class B, 552 flares are of class C, 142 flares are of class M, and 23 flares are of class X. These 845 flares come from 472 active regions (ARs). The duration of a flare ranges from several

[1]Retrieved on 05/01/2021 from `http://jsoc.stanford.edu/`

Table 2.1 13 Important SHARP Physical Parameters

Parameter	Description
ABSNJZH	Absolute value of the net current helicity
AREA_ACR	Area of strong field pixels in the active region
EPSZ	Sum of z-component of normalized Lorentz force
MEANPOT	Mean photospheric magnetic free energy
R_VALUE	Sum of flux near polarity inversion line
SAVNCPP	Sum of the modulus of the net current per polarity
SHRGT45	Fraction of area with shear $> 45°$
TOTBSQ	Total magnitude of Lorentz force
TOTFZ	Sum of z-component of Lorentz force
TOTPOT	Total photospheric magnetic free energy density
TOTUSJH	Total unsigned current helicity
TOTUSJZ	Total unsigned vertical current
USFLUX	Total unsigned flux

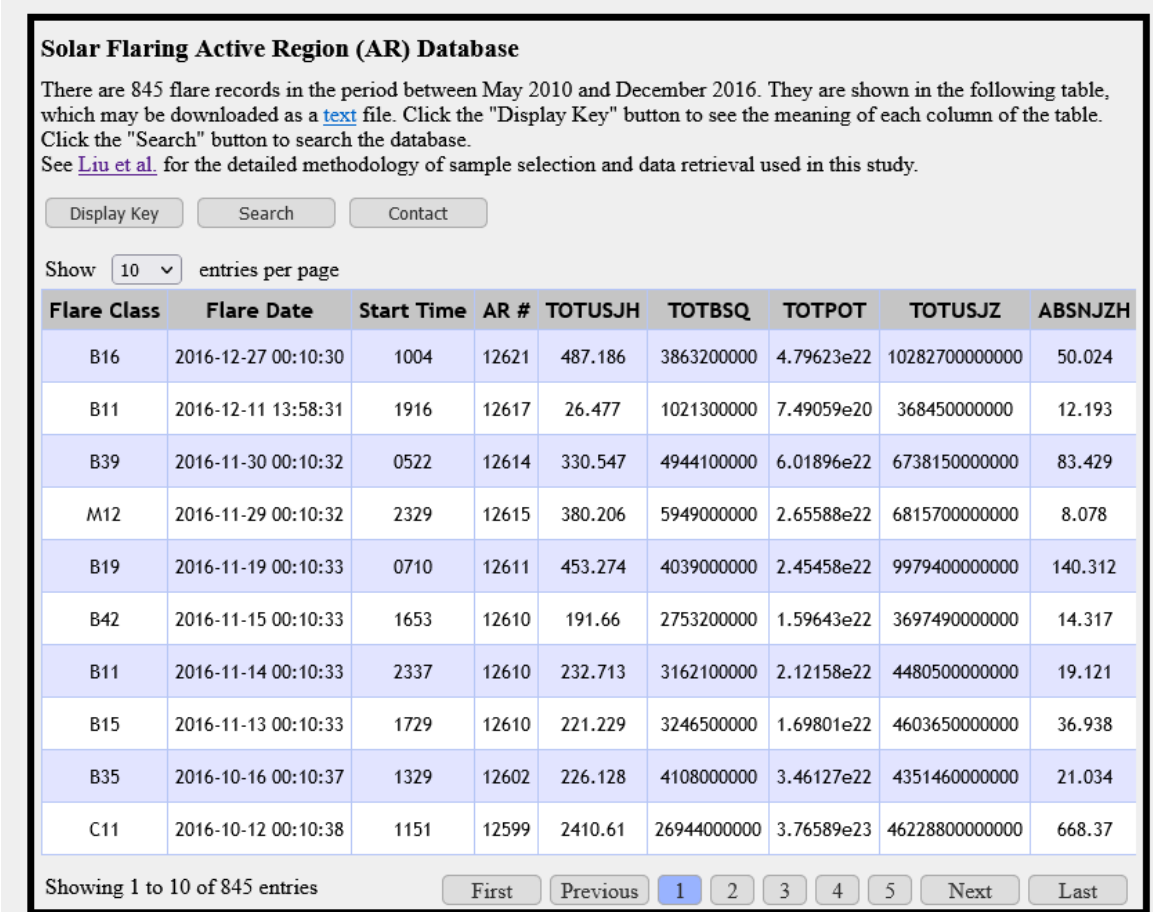
minutes to hours. The duration of an AR ranges from several minutes to days. Table 2.2 summarizes the flare information.

We created and stored 845 corresponding data samples in our database, shown in Figure 2.1[2], where each data sample contains values of the 13 physical parameters or features listed in Table 2.1. The two digits following a class label (B, C, M, X) are ignored in performing flare prediction. The time point of a data sample is the beginning time (00:00:01 early morning) of the start date of a flare and the label of the data sample is the class which the flare belongs to. These labeled data samples are used to train the DeepSun system.

2.1.3 Machine Learning Algorithms

DeepSun employs three machine learning algorithms for flare prediction: random forests (RF) [33], multilayer perceptrons (MLP) [145, 32] and extreme learning

[2]Retrieved on 05/15/2021 from `https://nature.njit.edu/spacesoft/Flare-Predict`



Solar Flaring Active Region (AR) Database

There are 845 flare records in the period between May 2010 and December 2016. They are shown in the following table, which may be downloaded as a text file. Click the "Display Key" button to see the meaning of each column of the table. Click the "Search" button to search the database.
See Liu et al. for the detailed methodology of sample selection and data retrieval used in this study.

Display Key | Search | Contact

Show 10 entries per page

Flare Class	Flare Date	Start Time	AR #	TOTUSJH	TOTBSQ	TOTPOT	TOTUSJZ	ABSNJZH
B16	2016-12-27 00:10:30	1004	12621	487.186	3863200000	4.79623e22	10282700000000	50.024
B11	2016-12-11 13:58:31	1916	12617	26.477	1021300000	7.49059e20	368450000000	12.193
B39	2016-11-30 00:10:32	0522	12614	330.547	4944100000	6.01896e22	6738150000000	83.429
M12	2016-11-29 00:10:32	2329	12615	380.206	5949000000	2.65588e22	6815700000000	8.078
B19	2016-11-19 00:10:33	0710	12611	453.274	4039000000	2.45458e22	9979400000000	140.312
B42	2016-11-15 00:10:33	1653	12610	191.66	2753200000	1.59643e22	3697490000000	14.317
B11	2016-11-14 00:10:33	2337	12610	232.713	3162100000	2.12158e22	4480500000000	19.121
B15	2016-11-13 00:10:33	1729	12610	221.229	3246500000	1.69801e22	4603650000000	36.938
B35	2016-10-16 00:10:37	1329	12602	226.128	4108000000	3.46127e22	4351460000000	21.034
C11	2016-10-12 00:10:38	1151	12599	2410.61	26944000000	3.76589e23	46228800000000	668.37

Showing 1 to 10 of 845 entries

First | Previous | 1 | 2 | 3 | 4 | 5 | Next | Last

Figure 2.1 Screenshot showing our online flare database.

Table 2.2 Numbers of Flares and Active Regions Per Solar Flare Class

Flare Class	Number of Flares	Number of ARs
B	128	88
C	552	281
M	142	88
X	23	15

machines (ELM) [79, 80]. RF is a tree-based algorithm comprised of multiple binary classification and regression trees (CART) while both MLP and ELM are feed-forward artificial neural networks. All the three algorithms are well suited for multi-class classification. In addition, we develop an ensemble (ENS) algorithm, which works by taking the majority vote of RF, MLP and ELM. If there is no majority vote for a test data sample, ENS outputs "no verdict" for the test data sample.

2.1.4 Performance Evaluation

We conducted a series of experiments to evaluate the performance of the machine learning algorithms presented in Section 2.1.3 using the database described in Section 2.1.2. To avoid bias and to keep the data as balanced as possible, we created 100 csv (comma separated values) datasets of which each dataset included all B, M, and X classes and randomly selected 142 data samples of class C. We used 10-fold cross validation in which for each data set, we randomly formed 10-fold partitions using the KFold function provided by the scikit-learn library in Python [134]. Each machine learning algorithm was trained by nine of the 10-folds, and the 10th fold was used for testing. To overcome errors associated with cross validation, we repeated the procedure 100 times for each of the 100 datasets that resulted in 10000 iterations to produce the final result.

We converted the multiple-class classification problem at hand into four binary classification problems for the four classes B, C, M, and X. For example, consider the binary classification problem for class B. Here, we say a data sample is positive if it is in class B, or negative if it is not in class B, *i.e.*, it is in class C, M, or X. We define TP (true positive), FP (false positive), TN (true negative), FN (false negative) as follows. TP is a data sample where an algorithm predicts the data sample to be positive and the data sample is indeed positive. FP is a data sample where the algorithm predicts the data sample to be positive while the data sample is actually negative. TN is a data sample where the algorithm predicts the data sample to be negative and the data sample is indeed negative. FN is a data sample where the algorithm predicts the data sample to be negative while the data sample is actually positive. We also use TP (FP, TN, FN, respectively) to represent the number of true positives (false positives, true negatives, false negatives, respectively). Because we are tackling an imbalanced classification problem (see Table 2.2), we adopt two performance metrics, balanced accuracy (BACC) and true skill statistics (TSS), where BACC is defined as

follows:

$$\mathrm{BACC} = \frac{1}{2}\left(\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} + \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}\right)$$

and TSS is defined as follows:

$$\mathrm{TSS} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} - \frac{\mathrm{FP}}{\mathrm{TN}+\mathrm{FP}}$$

BACC and TSS are well suited for imbalanced classification of solar eruptions [120, 121, 117]. We obtain BACC and TSS for each binary classification problem. There are four binary classification problems. We then calculate the average of the BACC and TSS values obtained from the four classification problems, and use the average as the result for the multi-class classification problem.

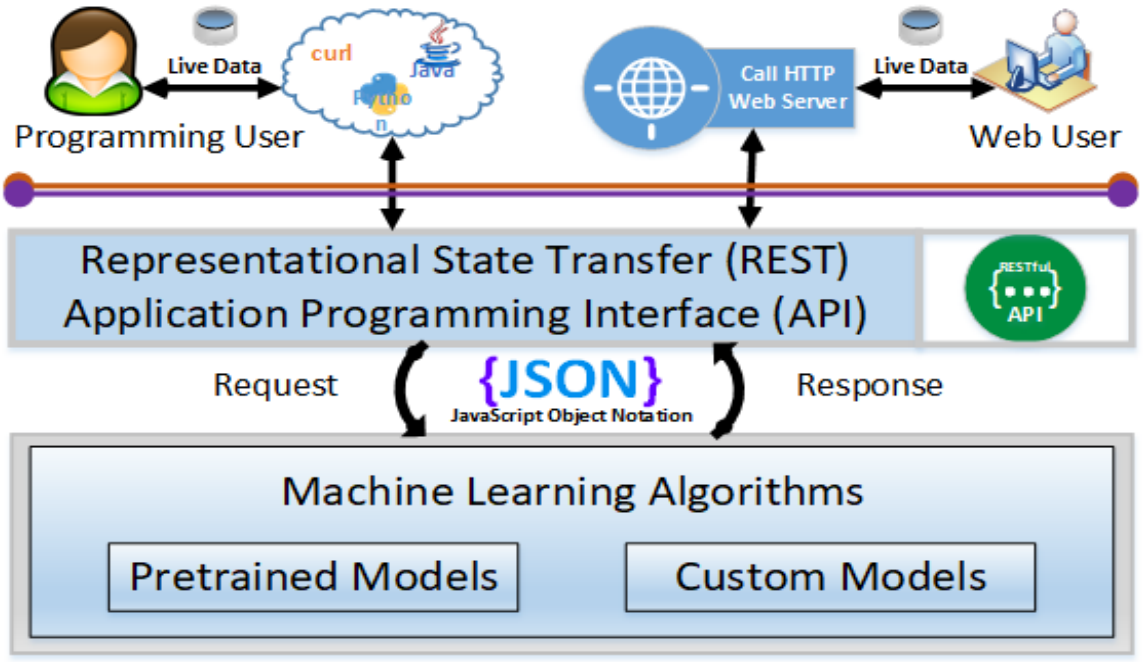


Figure 2.2 Overview of DeepSun.

We implemented the machine learning algorithms in Python leveraging the scikit-learn packages [134]. Each algorithm has different optimization parameters to be tuned based on the training and test datasets. We used random forests (RF)

composed of 500 to 1000 trees and set the number of features to six to find the best node split. For multilayer perceptrons (MLP) and extreme learning machines (ELM), we set the number of hidden layers to 200. These parameter values were chosen to maximize TSS values.

Table 2.3 compares the BACC and TSS values of the machine learning algorithms at hand for each binary classification problem and for the overall multi-class classification problem where the highest performance metric values are highlighted in boldface. It can be seen from the table that the proposed ENS algorithm is better than the existing algorithms RF, MLP and ELM. However, all the four algorithms perform poorly in predicting X-class flares. This happens probably because the X class has much fewer flares than the other classes. Overall, there were approximately less than 2% data samples receiving "no verdict."

Table 2.3 Flare Prediction Results Using 13 SHARP Parameters and Four Machine Learning Algorithms

	Class B	**Class C**	**Class M**	**Class X**	**Average**
BACC					
ENS	**0.871**	**0.691**	**0.790**	**0.670**	**0.756**
RF	0.834	0.663	0.749	0.645	0.723
MLP	0.818	0.659	0.757	0.599	0.708
ELM	0.791	0.641	0.721	0.608	0.690
TSS					
ENS	**0.745**	**0.380**	**0.551**	**0.362**	**0.507**
RF	0.708	0.378	0.537	0.330	0.488
MLP	0.661	0.285	0.526	0.010	0.371
ELM	0.618	0.296	0.446	0.227	0.397

2.1.5 DeepSun Framework and System Design

The four machine learning algorithms (ENS, RF, MLP, ELM) have been implemented into our DeepSun system where the algorithms are used as a back-end, also known

as the server-side, engine for the machine-learning-as-a-service (MLaaS) platform. Figure 2.2 presents the overall contextual architecture of the DeepSun framework. The system supports two different types of users: Web and programming. The Web user invokes the service by accessing a graphical user interface (GUI) to perform flare predictions. The programming user can use any programming language that supports HTTP requests, such as Java, C++, Python, Node.js, JavaScript modules in React or other frameworks to perform flare predictions.

MLaaS is a representational state transfer (REST) application programming interface (API) that supports JSON (JavaScript Object Notation) formatted payloads in the request and response. JSON is a plain-text and lightweight data-interchange format. It is structured with attributes and values in an easy way for humans to read and write. JSON is language independent but it is easy to parse; therefore almost every programming language supports it. The request transmits the user's data from the front-end to the back-end and must include well defined JSON formatted test data to predict or training data to create a predictive model. The response transmits the result from the back-end to the front-end, which is a well formatted prediction result or the predictive model identifier. Here, the front-end means the client-side that can be a Web-designed interface for the Web user or a program for the programming user.

2.1.6 System Implementation

When a user visits DeepSun's home page, the user sees three options. Option 1 allows the user to select the pretrained models provided by DeepSun. Option 2 allows the user to upload his/her own training data to create his/her own machine learning models for solar flare prediction. Option 3 allows the user to perform solar flare prediction using RESTful services. Figure 2.3 shows DeepSun's home page[3].

[3]Retrieved on 05/15/2021 from `https://nature.njit.edu/spacesoft/DeepSun/`

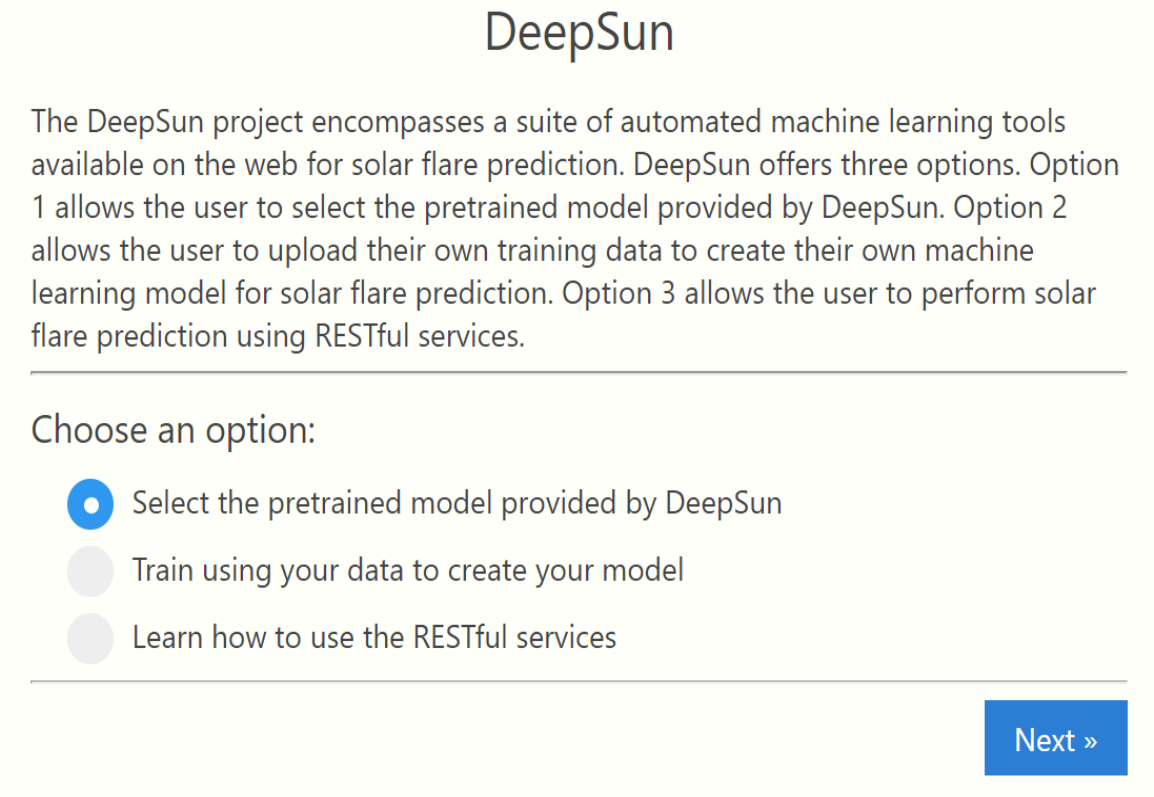


Figure 2.3 Screenshot showing the home page of DeepSun.

Pretrained Models in DeepSun The pretrained models are ready-to-use models that were created using the database described in Section 2.1.2. With the pretrained models, a user has multiple options to load test data samples containing the 13 physical parameters or features listed in Table 2.1: (1) Manually enter the data samples with values of the 13 physical parameters one by one in the provided text boxes. (2) Load sample data provided by the DeepSun engine. (3) Load the user's own data in a file, in which each line contains the values of the 13 physical parameters. The user may invoke the services to predict all the loaded, or entered, test data at once or make predictions one by one. Figure 2.4 shows the webpage of pretrained models on which four predictions were made using the ENS algorithm.

Custom Models in DeepSun DeepSun allows the user to load his/her data to train and create his/her custom model to predict solar flares. The training data

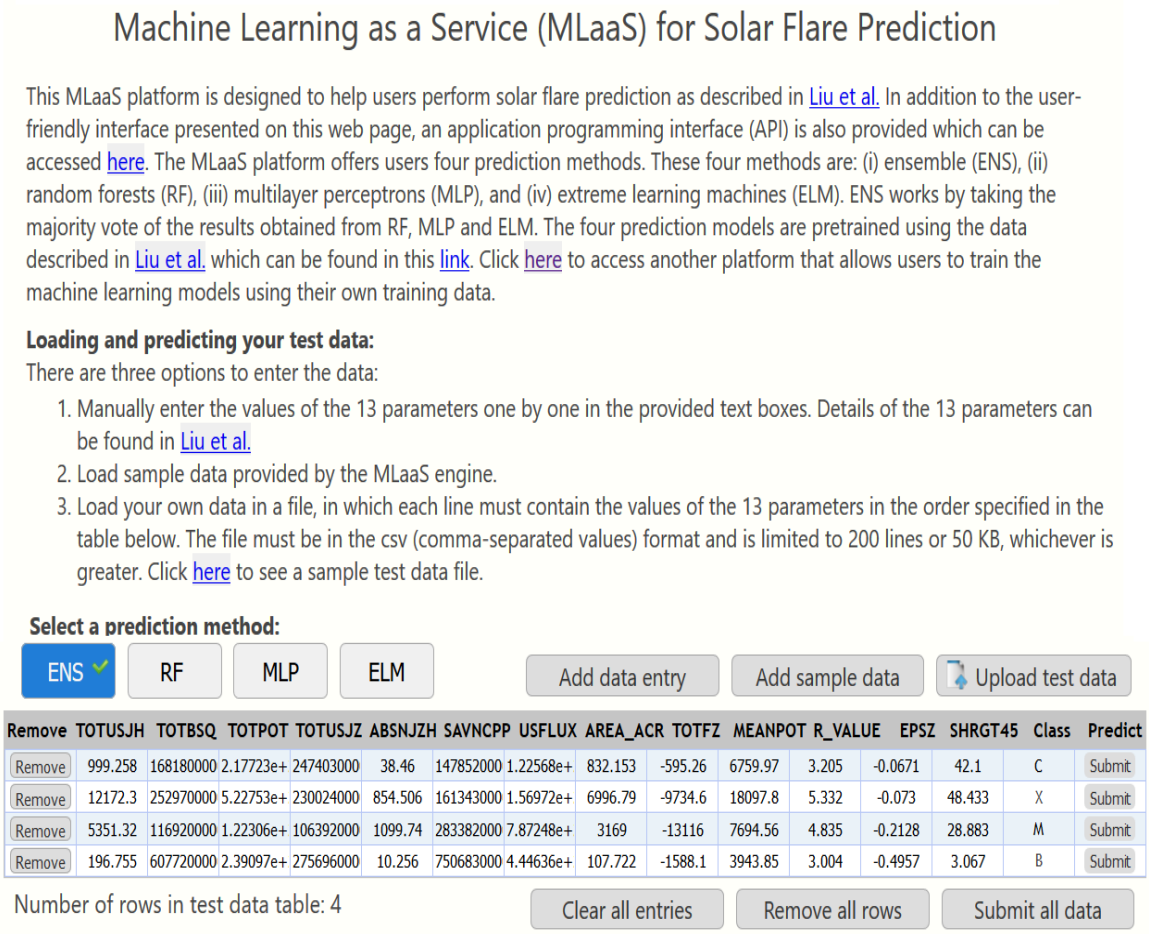


Figure 2.4 Screenshot showing the webpage with pretrained models of DeepSun.

are saved in a file meeting DeepSun's format requirement. When the user creates a custom model, a model identifier (id) is assigned to the current session. If the created model is idle for 24 hours, it will be deleted. Once the model is ready, the user goes to the DeepSun's graphical user interface with the assigned model id to perform flare predictions as done with the pretrained models. The model id is used to distinguish between the custom models and pretrained models. Figure 2.5 shows the webpage of custom models with example training data displayed.

RESTful API for DeepSun The RESTful API is designed to help the programming user perform solar flare predictions using the pretrained or custom models. The API supports the POST request to predict solar flare occurrence or create a custom model,

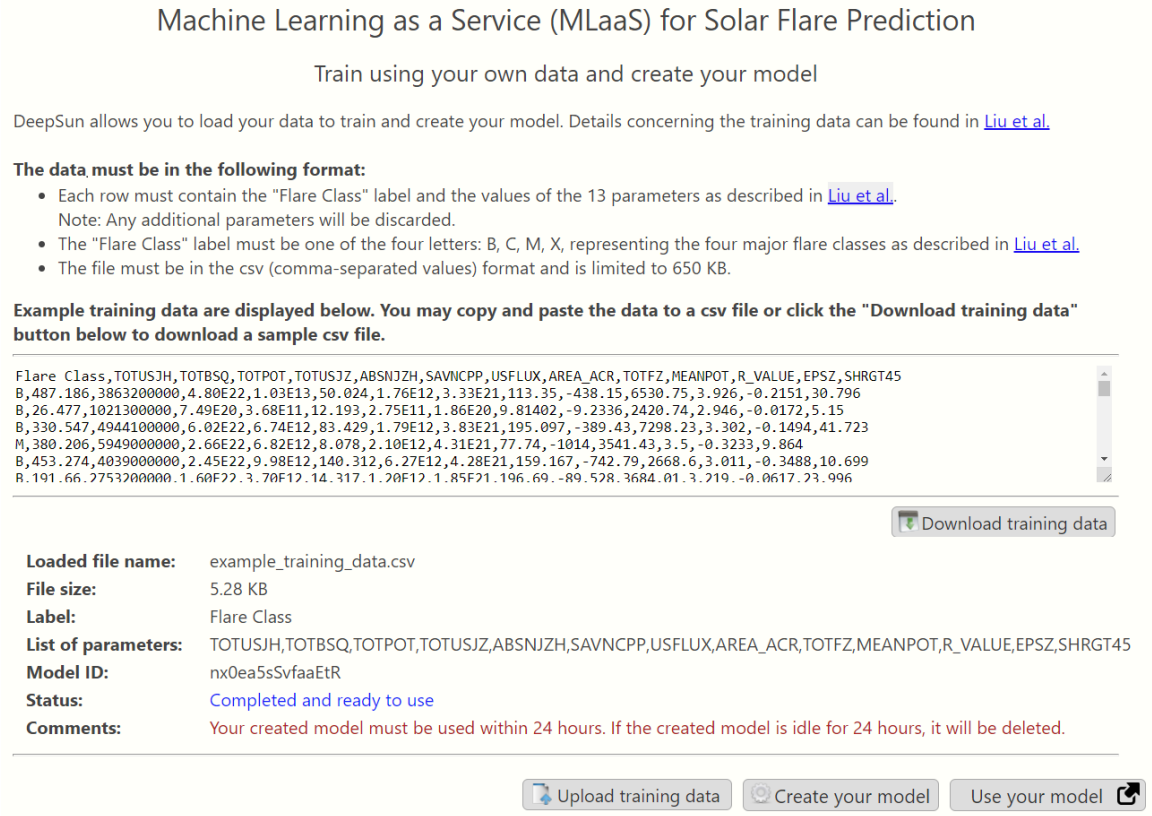


Figure 2.5 Screenshot showing the webpage with custom models of DeepSun.

and the GET request to get a random data sample from our training database. The interface supports JSON formatted strings for requests' body and their results. The interface also supports two different debug levels; they are (i) INFO which is the default debug mode and (ii) DEBUG to return additional data with the result.

The return result from the POST request is a JSON object including the predicted solar flare occurrence and its class. Each test data sample is associated with a JSON object that includes two attributes. One attribute is "fcnumber" which is the numerical representation for the solar flare class where we use "1" ("2", "3", "4", respectively) to represent class B (C, M, X, respectively). The other attribute is "fcname" which is the solar flare class name.

In addition, the RESTful API uses the POST request to create a custom model. The body of the request must be JSON formatted strings for an array of JSON objects.

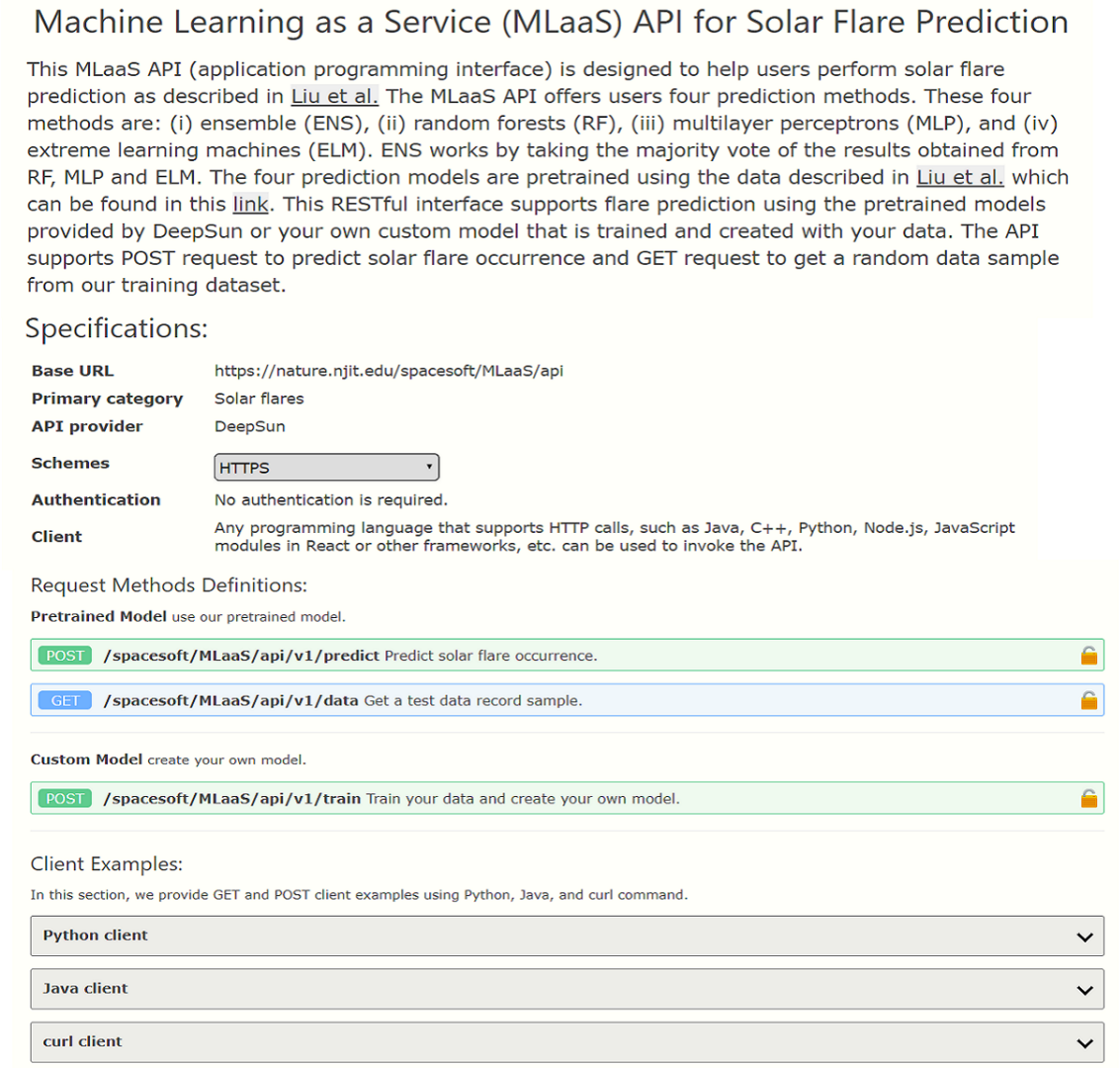


Figure 2.6 Screenshot showing the RESTful API page of DeepSun.

Each object must contain the 13 physical parameters and its flare class label where the label must be one of B, C, M, X. The return result of this POST request is a JSON object that contains the custom model identifier (id) which can be used for flare prediction. The custom model includes all the four algorithms (ENS, RF, MLP, ELM). Since the API is a RESTful interface, any programming language that supports HTTP calls, such as Java, C++, Python, Node.js, JavaScript modules in React or other frameworks can be used to invoke the API. Figure 2.6 shows the RESTful

API page on which the definitions of the available methods and client examples are displayed.

2.1.7 Related Work

There are two groups of work that are closely related to ours. The first group is concerned with solar flare forecasting. Many studies in this group used parameters derived from the line-of-sight (LOS) component of the photospheric magnetic field and produced probability outputs for the occurrence of a certain magnitude flare in a time period [117]. Some researchers [63] used sunspot classification and Poisson statistics to provide probabilities for an active region (AR) to produce flares with different magnitudes within 24 hours. Song *et al.* [156] used three LOS magnetic parameters together with the ordinal logistic regression (OLR) method to predict the probabilities of a one-day flare. Bloomfield *et al.* [27] suggested that the prediction probabilities should be converted into a binary (*i.e.*, yes-or-no) forecast before they can be translated as flare-imminent or flare-quiet. Following this suggestion, Yuan *et al.* [185] employed support vector machines (SVMs) to obtain a clear true or false flare prediction for different flare classes.

On the other hand, the full vector data provide more information about the photospheric magnetic field structure compared to the LOS field. This type of information may provide better flare prediction performance, but due to the limitation imposed by ground-based vector magnetic field observations, the work on flare forecast is limited. For example, Leka and Barnes [110] used a small sample of vector magnetograms from the Mees Solar Observatory and applied a discriminant analysis to differentiate between flare-producing and flare-quiet ARs within few hours. The authors later extended their work and used a larger number of samples with a 24-hour prediction window on producing probabilistic forecasts [19].

Since May 2010, the Helioseismic and Magnetic Imager (HMI) onboard the Solar Dynamics Observatory (SDO) [29] has been producing high quality photospheric vector magnetograms with high-cadence and full-disk coverage data. Using these data, Bobra and Couvidat [29] calculated a number of magnetic parameters for each AR. They selected 13 from all the available parameters and achieved good prediction performance using an SVM method for flares greater than M1.0 class. Nishizuka *et al.* [133] applied a number of machine learning algorithms to HMI data and produced prediction models for $\geq$M and X-class flares with reasonably high performance. More recently, we employed a long short-term memory network for flare prediction [120].

The second group of related work is concerned with services computing. Benmerar *et al.* [21] developed a brain diffusion MRI (magnetic resonance imaging) application to overcome the SaaS (software-as-a-service) limitations caused by intensive computation. The application provides APIs that tackle browser paradigms to reduce the parallel computation rendered in the client side of the browser.

Wu *et al.* [181] developed an automated testing technique to detect cross-browser compatibility issues so that they can be fixed. These cross-browser issues cause problems for an organization to create JavaScript web applications. The authors employed an existing record-and-play technique, Mugshot [128], to design an incremental cross-browser incompatibility algorithm. The system starts off by injecting the record library into the browsers, collects traces and events to be replayed, and runs the detection algorithm to find different types of incompatibilities among the browsers.

Song *et al.* [157] presented a machine learning algorithm for IT support services to automate the problem determination and classification, and also find the root cause of a problem. The algorithm is an on-line perceptron that learns about the user's problems from the data that were generated from logs and monitoring information across different systems. The algorithm then categorizes the problems by finding

the actual root cause from what it learned from the data. The algorithm employs an incremental learning technique and is able to automatically adjust the classifier parameters.

Li *et al.* [112] described a new software documentation recommendation methodology that adopts a learn-to-rank (LTR) technique. LTR is an application of supervised and semi-supervised machine learning techniques. Their strategy combines the social context from a questions-and-answers online system and the content of official software documentation to build the LTR model to provide accurate and relevant software documentation recommendations. Their experimental results showed that this approach outperforms traditional code search engines including the Google search engine.

Our DeepSun system differs from the above works in two ways. First, DeepSun provides services dedicated to solar flare prediction, which has not been addressed by the existing services computing systems. Second, in the solar flare forecasting area, DeepSun is the first MLaaS system, to our knowledge, that allows scientists to perform multi-class flare prediction through the Internet.

2.1.8 Summary

We present a machine-learning-as-a-service framework (DeepSun) for solar flare prediction. This framework provides two interfaces: a web server where the user enters the information through a graphical interface and a programmable interface that can be used by any RESTful client. DeepSun employs three existing machine learning algorithms, namely random forests (RF), multilayer perceptrons (MLP), extreme learning machines (ELM), and an ensemble algorithm (ENS) that combines the three machine learning algorithms. Our experimental results demonstrated the good performance of the ensemble algorithm and its superiority over the three existing machine learning algorithms.

In the current work, we focus on data samples composed of SHARP physical parameters. We collect 845 data samples belonging to four flare classes: B, C, M, and X across 472 active regions. In addition, the Helioseismic Magnetic Imager (HMI) aboard the Solar Dynamics Observatory (SDO) produces continuous full-disk observations (solar images). In future work, we plan to incorporate these HMI images into our DeepSun framework and extend our previously developed deep learning techniques [123, 77, 78] to directly process the images. We also plan to combine our recently developed deep learning algorithms using the SHARP parameters [120] with the image-based techniques and machine learning algorithms described in this research for more accurate solar flare prediction.

2.2 Flare Prediction with Deep Learning

2.2.1 Background and Related Work

Solar flares are known to be the largest explosion in the solar system. They usually associated with coronal mass ejections (CMEs) or solar energetic particles (SEPs) [117, 122, 2]. As described in Chapter 2.1, there were many efforts to monitor and predict solar flares because they could affect and damage the near-Earth environments such as satellites, space stations, and technologies [53]. Therefore, accurate prediction of flares occurrences is becoming crucial to scientist for disaster recovery and preparedness. The prediction task is still challenging and researchers are tying to invent algorithms and models to provide accurate flares prediction. Many machine learning algorithms have been created to predict solar flares occurrences such as [119, 4]. In this work, we continue our previous effort to predict solar flares as described in Section 2.1 and we provide a novel algorithm to introduce an operational forecasting algorithm to predict solar flares with high accuracy. We propose a new hybrid-Transformer model that combines multiple recurrent neural network (RNN)

components to forecast solar flares for the next 24-72 hours ahead for flares types ≥C, ≥M, and ≥M5.0 using the *SDO*/HMI vector magnetic data products.

2.2.2 Data

Following [117, 119, 4, 2], we conducted our research using the Space-weather HMI Active Region Patches (SHARP) [31] that were developed by the *SDO*/HMI toward the end of 2012. The data are available and can be downloaded from the Joint Science Operations Center (JSOC)[4]. We combined the data from two different data series. The first one is the hmi.sharp that has the SHARP parameters. The other data series is the cgem.Lorentz which was built to include estimations of integrated Lorentz forces [60]. We collected the SHARP data samples from both data series using the Python package SunPy [159] at a 12 minutes cadence. The SHARP parameters are described as follows.

- Absolute value of the net current helicity (ABSNJZH),
- Area of strong field pixels in the active region (AREA_ACR),
- Mean characteristic twist parameter (MEANALP),
- Mean angle of field from radial (MEANGAM),
- Mean gradient of horizontal field (MEANGBH),
- Mean gradient of total field (MEANGBT),
- Mean gradient of vertical field (MEANGBZ),
- Mean vertical current density (MEANJZD),
- Mean current helicity (MEANJZH),
- Mean photospheric magnetic free energy (MEANPOT),

[4]Retrieved on 11/12/2022 from `http://jsoc.stanford.edu/`

- Mean shear angle (MEANSHR),
- Sum of flux near polarity inversion line (R_VALUE),
- Sum of the modulus of the net current per polarity (SAVNCPP),
- Fraction of area with shear $> 45°$ (SHRGT45),
- Total photospheric magnetic free energy density (TOTPOT),
- Total unsigned current helicity (TOTUSJH),
- Total unsigned vertical current (TOTUSJZ),
- Total unsigned flux (USFLUX),
- Sum of x-component of Lorentz force (TOTFX),
- Sum of y-component of Lorentz force (TOTFY),
- Sum of z-component of Lorentz force (TOTFZ),
- Sum of x-component of normalized Lorentz force (EPSX),
- Sum of y-component of normalized Lorentz force (EPSY),
- Sum of z-component of normalized Lorentz force (EPSZ).

We selected a subset of the parameters to perform the prediction task as described by [119] feature ranking. The nine parameters we focused on are: TOTUSJH, TOTUSJZ, USFLUX, MEANALP, R_VALUE, TOTPOT, SAVNCPP, AREA_ACR, and ABSNJZH The SHARP parameters values are in different scales and units, therefore, we normalized each parameter values using the min-max normalization method following [122, 2]. Let p_i^k be the original value of the ith parameter of the kth data sample. Let q_i^k be the normalized value of the ith parameter of the kth

data sample. Let min_i be the minimum value of the ith parameter. Let max_i be the maximum value of the ith parameter. Then

$$q_i^k = \frac{p_i^k - min_i}{max_i - min_i}. \tag{2.1}$$

Data quality is essential in machine learning. Therefore, when we collected the data we focused on high quality and discarded the low data. We collected B-, C-, M- and X-class flares that occurred between 2010 and 2022 for the active regions from the *GOES* X-ray flare catalogs that provided by the National Centers for Environmental Information (NCEI). We followed the data sample quality measures defined by [30, 119, 2]. We discarded any active regions that were outside $\pm$ 70° of the central meridian. These ARs are near the limb and have projection effects that make the calculation of the SHARP parameters that produced by those ARs to incorrect. In addition, we discard the data samples if the following conditions are met:

1. Flares with an absolute value of the radial velocity of *SDO* that is greater than 3500 m s^{-1}
2. HMI data with low-quality as defined by [76],
3. Records with missing values and incomplete SHARP parameters.

The data are divided into two subsets: the training data using the data samples from 2010 to 2021 and the testing data using the data samples from 2022.

2.2.3 Methodology

Prediction Task: In this work, we are proposing flare occurrence forecasting within the next 24-hour (48- and 72-hour, respectively) using binary classification. That is for any given active region (AR), will it produce K-class within the next 24-hour (48- and 72-hour, respectively), where K can be $\geq$C, $\geq$M, or $\geq$M5.0. We divided the problem into three different binary classifications so that each K class is treated individually. We focused on these K classes following [29, 89, 132, 119] because they

highlighted and showed the importance of each class especially for ≥M5.0. As [119] indicated, there are few X-class flares, therefore, they used ≥M5.0 because it means the GOES X-ray flux value of the flare is greater than 5×10^{-5} Wm^{-2} and is considered a major flare.

Data Labeling: Data labeling is an important task in machine learning to develop good models and algorithms. After we collect the high quality data, the next step is to properly label the data and create data samples for training and testing. Specifically, for a given data sample x_t that is collected from an AR at time t, we collect all data samples between t and $t - 24hours$ (48 and 72 hours, respectively). For ≥C class, if the flare at time point t is C, M, or X, the collected samples are labeled positive, otherwise, the data samples are labeled negative. For ≥M, if the flare at time point t is M or X, the data samples are labeled positive, otherwise they are labeled negative. Similarly for ≥M5.0, if the flare at time point t is M with intensity ≥ 5.0, the data samples are labeled positive, otherwise, they are labeled negative. Figure 2.7 illustrates the collection of positive and negative data samples for flare class ≥C. Collection and construction of data samples are done in the same way for flare classes ≥M and ≥M5.0. The data samples are collected at 12 minutes cadence and due to the removal of low quality data sets as describe in Section 2.2.2, some data samples might be missing. Therefore, we use zero-padding technique to fill in the sequence of data within the 24-72 hours [119, 2].

Data Augmentation The data set used in the study is considered imbalanced especially for the ≥M and ≥M5.0 which poses challenges and causes over-fitting during the training process and result in poor model performance. In order to overcome the imbalanced data, one may use data augmentation technique. Data augmentation is an important technique that enriches the training data and increases the generalization of the model [55]. We use the Gaussian white noise (GWN) data

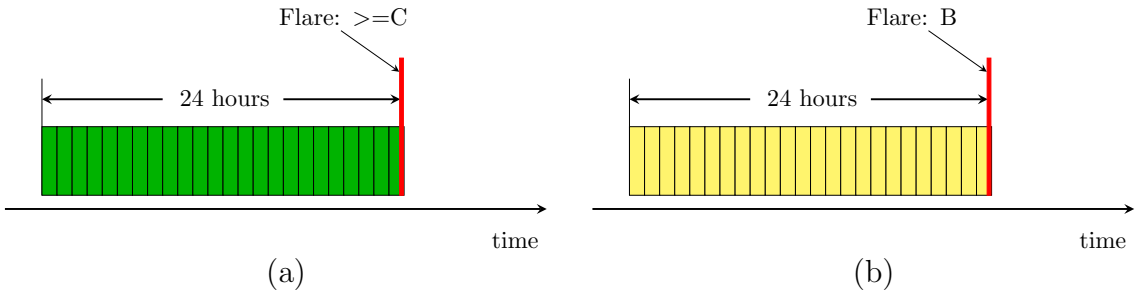


Figure 2.7 Example for collecting positive and negative data samples on a AR for flare class ≥C. The red vertical line shows the start time of ≥C (or ≥M, or ≥M5.0). (a) represents the positive labeled data samples because the occurring flare is ≥C. (b) represents the negative labeled data samples because the occurring flare is B.

augmentation because it has showing significant improvement when it is applied as data augmentation technique[165, 113]. GWN has a property that any two values are statistically independent regardless of how close they are in time. The stationary random values of GWN are generated using zero mean and 5% of the data standard deviation. The data augmentation is applied to the minority class during the training, leaving the majority class as is. The data is pre-processed and fed into the model input layer. During testing, the data were left without any augmentation so that the model predicts the actual testing data only to avoid any misleading performance.

The SolarFlareNet Architecture: Figure 2.8 represents the overall contextual architecture of the SolarFlareNet model used in the study. The algorithm and model components are built and configured using the Tensorflow framework[5]. The model is a hybrid-based transformer that combines a one dimension (1D) convolution neural network (CNN) (Conv1D), long short-term memory (LSTM)[4], transformer encoder blocks, and additional helping layers consist of batch normalization layers, dropout layers, and dense layers. The first layer of the model is the input layer which takes the input data. Since the data in this research is time series, each input data sample to

[5]Retrieved on 10/28/2022 from `https://www.tensorflow.org`

the network is configured with time series sequence so that the data sample consists of m consecutive records x_{t-m+1}, x_{t-m+2} $\ldots x_{t-1}$, x_t where x_t is the data sample at time point t following [2]. The input data is augmented as specific in Section 2.2.3. The input layer is followed by a batch normalization layer (BNL, that is also added to different location of the model architecture). BNL is an add-on technique to the stabilize the neural network, makes it faster, and help in avoiding over-fitting during training. Here, we use BNL instead of layer normalization to avoid the effect of outlier in time series which does not exist in nature language processing (NLP) where layer normalization is proven to be better than BNL[187]. BNL applies transformation to try to maintain the mean output close to 0 and the output standard deviation close to 1. We apply BNL after the input layer, LSTM layer, and within the transformer encoder to make sure the network is stable throughout the training phase. BNL is followed by Conv1D layer to process the time series because time series have a strong one dimension (1D) time locality that can be extracted by convolutions [101]. Then, LSTM layer that has regularization to also avoid over-fitting. LSTM is very famous of handling the time series data and its temporal correlation and dependency. Adding LSTM layer on top of Conv1D layer has shown significant improvement to process time series [5, 6, 3]. The LSTM layer passes the learned features and patterns to the BNL to stabilize the network before the data goes to the transformer encoder (TE). In this study, we are using the transformer encoder without the decoder this is because the problem in hand is dealing with time series not NLP where decoder is required to decode the words for sentence translation. The number of TE blocks is set to 4 not 8 as defined by [169] because the number of encoder blocks has significant effect on the overall model performance and results and dependent on the data and its dimensions. In this study, if we less than 4 blocks the model is not able to learn useful patterns and is under-fitted, and if we use more than 4 to 8, it causes overhead on the encoder processing and the model tends to do excessive over-fitting and tends to lean

toward the majority class in the data ignoring the positive class entirely. Each TE block is configured with a dropout, multi-head attention (MHA), a BNL Conv1D, and an LSTM layers. The multi-head attention layer is the most important layer in the encoder because it provides the transformation on the sequence values to obtain the distinct metrics. The MHA layer is configured with 4 heads and the each attention head is set to 4 as well. The dropout (DO) layers are mainly used to overcome the over-fitting due to the imbalanced data. The DO layers drops a percentage of the neurons from the architecture that causes the internal architecture of the model changes and allows for better performance and stability. Finally, the softmax function is used as the final activation function because it produces a probability prediction of positive and negative classes.

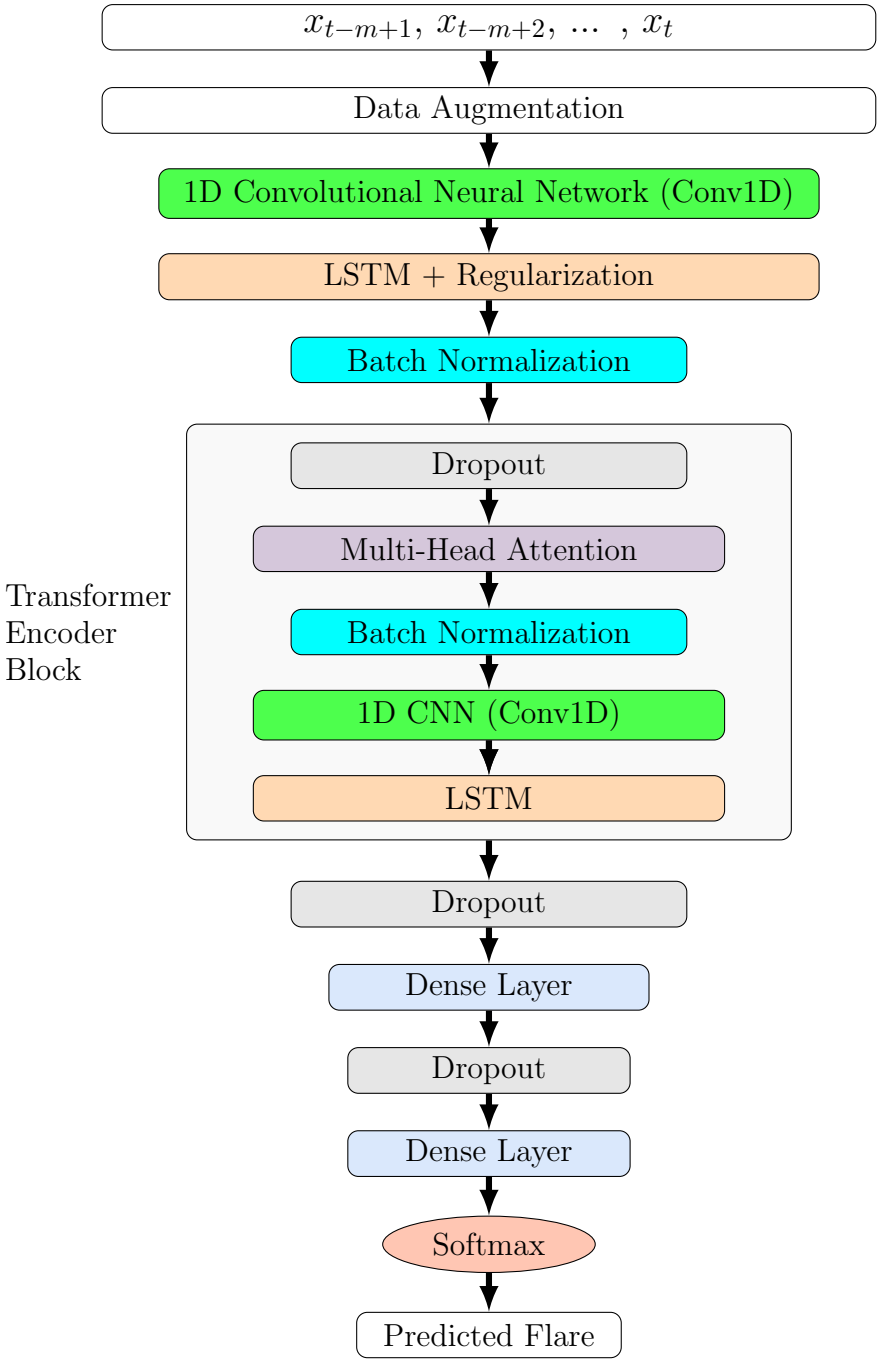


Figure 2.8 The overall contextual architecture used for the Transformer-based operational flare forecasting model (SolarFlareNet).

2.2.4 Comparative Study

In this section, we provide a comparative study between the proposed framework, SolarFlareNet, and a previously developed algorithm using LSTM with an attention

layer[119] who predict the same binary classification proposed in our study, and three existing machine learning algorithms namely multilayer perceptron (MLP) [139], support vector machine (SVM) [50], and random forest (RF) [33]. All algorithms and trained and tested on the same data sets as specified in Section 2.2.2. The validation data is set during the training by configuring the model to use 10% of the training data as validation. The performance metrics used in the comparison are those used by [119] as follows.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{2.2}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{2.3}$$

$$\text{Accuracy (ACC)} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}, \tag{2.4}$$

$$\begin{aligned} &\text{Balanced Accuracy (BACC)} \\ &= \frac{1}{2}\left(\frac{\text{TP}}{\text{TP} + \text{FN}} + \frac{\text{TN}}{\text{TN} + \text{FP}}\right), \end{aligned} \tag{2.5}$$

$$\begin{aligned} &\text{Heidke Skill Score (HSS)} \\ &= \frac{2 \times (\text{TP} \times \text{TN} - \text{FP} \times \text{FN})}{(\text{TP} + \text{FN}) \times (\text{FN} + \text{TN}) + (\text{TP} + \text{FP}) \times (\text{FP} + \text{TN})}, \end{aligned} \tag{2.6}$$

$$\text{True Skill Statistics (TSS)} = \frac{\text{TP}}{\text{TP} + \text{FN}} - \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{2.7}$$

Table 2.4 shows the comparison results between the proposed SolarFlareNet and the other closely related machine learning algorithms. It can be seen from the table that our model outperforms all algorithms in all the performance metrics.

In addition, We also performed a cross-validation study to evaluate our model and also compare it with LSTM [119] only because it is the second best performing algorithms among the compared algorithms. The cross validation is based on the cross

validation strategy provided by [119]. Table 2.5 shows the cross-validation results for the SolarFlareNet and the LSTM models. The highest performance values are in boldface with the standard deviation between the parenthesis. The table also shows that our model still outperforms LSTM [119] in all performance metrics in the cross validation study.

In addition, we trained and tested the proposed model with the training and testing data used by [119] to assess our model against the data and compare the performance with results produced by [119] for the single run given the training from 2010 to 2013, the validation data sets from 2014, and the testing data from 2015 to 2018. We also performed the same cross-validation technique in this data set. Tables 2.6 and 2.7 show the results for the single run and the Cross validation, respectively. The results show that our model still outperforms the LSTM model in both single and cross-validation testing.

Furthermore, we performed a cross validation as specified by [119] on the 2010-2018 data and compared the results.

2.2.5 Ablation Study

To further assess our model and its components to avoid any redundancy or overhead in the model, we performed an ablation experiments by removing one main component at the time while keeping the other components to perform the training and testing. We created the following sub-models from SolarFlareNet:

1. SolarFlareNet-C is a sub-model without the Conv1D layer, captures the temporal patterns from the raw data without the strong correlation that is built by the Conv1D layer and passes to the LSTM layer.

2. SolarFlareNet-L is a sub-model without the LSTM layer, captures the 1 dimension time correlation from the input data, but it does not build temporal correlation that is learned by the LSTM layer.

Table 2.4 Performance Comparison of SolarFlareNet, MLP, SVM, RF, and LSTM [119]

		≥M5.0 class	≥M class	≥C class
Recall	MLP	0.624	0.646	0.593
	SVM	0.616	0.689	0.651
	RF	0.850	0.683	0.912
	LSTM	0.835	0.706	0.937
	SolarFlareNet	**0.872**	**0.879**	**0.953**
Precision	MLP	0.432	0.523	0.533
	SVM	0.542	0.601	0.482
	RF	0.904	0.852	0.792
	LSTM	0.929	0.832	0.815
	SolarFlareNet	**0.972**	**0.953**	**0.905**
ACC	MLP	0.709	0.821	0.793
	SVM	0.801	0.774	0.793
	RF	0.892	0.834	0.812
	LSTM	0.907	0.940	0.854
	SolarFlareNet	**0.962**	**0.942**	**0.922**
BACC	MLP	0.682	0.794	0.739
	SVM	0.744	0.747	0.721
	RF	0.812	0.846	0.796
	LSTM	0.891	0.850	0.848
	SolarFlareNet	**0.936**	**0.938**	**0.920**
HSS	MLP	0.314	0.296	0.371
	SVM	0.412	0.396	0.409
	RF	0.759	0.613	0.562
	LSTM	0.878	0.754	0.703
	SolarFlareNet	**0.891**	**0.910**	**0.844**
TSS	MLP	0.433	0.387	0.396
	SVM	0.453	0.441	0.462
	RF	0.829	0.701	0.674
	LSTM	0.834	0.702	0.697
	SolarFlareNet	**0.881**	**0.877**	**0.841**

3. SolarFlareNet-CL is a sub-model without the Conv1D and LSTM layers, and the transformer block is working on the data directly without the learned temporal patterns from Conv1D and LSTM layers.

Table 2.5 Performance Comparison of SolarFlareNet and LSTM [119] Using Cross Validation.

		≥M5.0 class	≥M class	≥C class
Recall	LSTM	0.890 (0.055)	0.759 (0.038)	0.944 (0.012)
	SolarFlareNet	**0.914** (0.016)	**0.873** (0.017)	**0.958** (0.011)
Precision	LSTM	0.949 (0.034)	0.784 (0.034)	0.808 (0.025)
	SolarFlareNet	**0.975** (0.018)	**0.911** (0.019)	**0.873** (0.037)
ACC	LSTM	0.957 (0.001)	0.929 (0.002)	0.851 (0.019)
	SolarFlareNet	**0.968** (0.021)	**0.940** (0.001)	**0.903** (0.025)
BACC	LSTM	0.945 (0.027)	0.874 (0.019)	0.845 (0.020)
	SolarFlareNet	**0.957** (0.022)	**0.935** (0.009)	**0.899** (0.026)
HSS	LSTM	0.917 (0.036)	0.760 (0.027)	0.698 (0.039)
	SolarFlareNet	**0.942** (0.023)	**0.886** (0.013)	**0.803** (0.050)
TSS	LSTM	0.868 (0.054)	0.751 (0.037)	0.696 (0.040)
	SolarFlareNet	**0.897** (0.043)	**0.880** (0.017)	**0.819** (0.052)

Table 2.6 Performance Comparison of SolarFlareNet and LSTM [119] Using the 2010-2018 Data Created by [119]

		≥M5.0 class	≥M class	≥C class
Recall	LSTM	0.978	0.881	0.762
	SolarFlareNet	**0.893**	**0.884**	**0.829**
Precision	LSTM	0.038	0.222	0.544
	SolarFlareNet	**0.975**	**0.910**	**0.806**
ACC	LSTM	0.899	0.909	0.829
	SolarFlareNet	**0.968**	**0.941**	**0.927**
BACC	LSTM	0.938	0.895	0.803
	SolarFlareNet	**0.946**	**0.920**	**0.890**
HSS	LSTM	0.074	0.347	0.539
	SolarFlareNet	**0.931**	**0.870**	**0.772**
TSS	LSTM	0.877	0.790	0.607
	SolarFlareNet	**0.893**	**0.839**	**0.780**

4. SolarFlarenet-T is a subm-model without the transformer encoder block, relies only the patterns learned from the Con1D and LSTM layers without doing the transformation on the sequence values to obtain the distinct metrics from the multi-head attention and other components.

Figure 4.16 shows the ablation test result for the TSS performance metric. It can be seen in the figure that the full SolarFlareNet outperforms the other sub-models. In addition, it can be seen that removing the Conv1D and LSTM layers at the same time as well as removing the transformer encoder blocks result in poor performance. This

Table 2.7 Performance Comparison of SolarFlareNet and LSTM [119] Using the 2010-2018 Data Created by [119] for Cross Validation

		≥M5.0 class	≥M class	≥C class
Recall	LSTM	0.960 (0.017)	0.885 (0.017)	0.773 (0.027)
	SolarFlareNet	**0.887** (0.003)	**0.866** (0.02)	**0.842** (0.038)
Precision	LSTM	0.048 (0.008)	0.222 (0.023)	0.541 (0.030
	SolarFlareNet	**0.973** (0.013)	**0.913** (0.019)	**0.867** (0.036)
ACC	LSTM	0.921 (0.014)	0.907 (0.013)	0.826 (0.015)
	SolarFlareNet	**0.968** (0.021)	**0.992** (0.021)	**0.943** (0.015)
BACC	LSTM	0.940 (0.007)	0.896 (0.004)	0.806 (0.004)
	SolarFlareNet	**0.943** (0.002)	**0.931** (0.016)	**0.905** (0.019)
HSS	LSTM	0.084 (0.015)	0.323 (0.030)	0.526 (0.021
	SolarFlareNet	**0.907** (0.002)	**0.884** (0.015)	**0.818** (0.034)
TSS	LSTM	0.881 (0.014)	0.792 (0.008)	0.612 (0.009)
	SolarFlareNet	**0.886** (0.003)	**0.857** (0.02)	**0.794** (0.039)

is because when removing the Conv1D and LSTM the sub-model loses big portion of the temporal correlation between the data. On the other hand, removing the transformer encoder losed the transformation of from the multi-head attentions and less patterns to learn. Therefore, combing all components together allows the model to learn both the temporal correlation as well as the transformation from the attention layers. Similar behaviour and result are produced for other performance metrics.

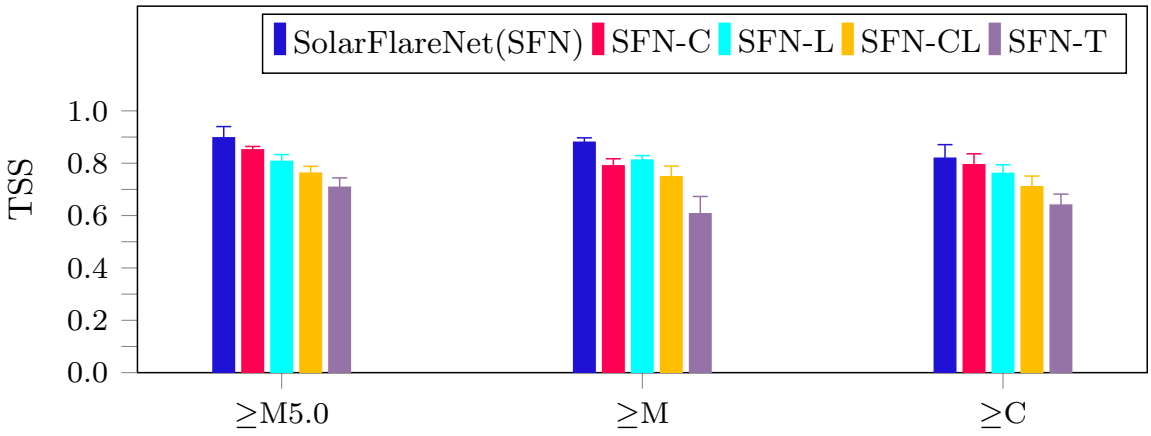


Figure 2.9 Ablation study result showing the TSS performance metrics.

2.2.6 Probabilistic Forecasting and Calibration

In this section, we provide probabilistic forecasting and calibration [2]. The two machine learning methods, SolarFlareNet and LSTM [119], are capable of producing probabilistic forecasting models and produce a probability between 0 and 1. We compare the probability with a preset threshold, which is set to 0.5, to determine the output produced by each machine learning method. Since, the problem studied in this research is a binary classification for each class K, where K is ≥C, ≥M, ≥M5.0, the probability of the positive class is 1 and probability of the negative class is 0. Generally, the model produces a value v where $0 \geq v \leq 1$ and we use v as the probabilistic forecasting output value from the model and interpreted as follows. At any time point t, the data sample x_t in an active region (AR), the probability of AR to produce K class within 24 hours (48 and 72, respectively) is v and v is the probabilistic estimate of how likely the AR will produce K-class flare.

In addition, we adopted a probability calibration method using the isotonic regression [104, 147] to adjust the probabilities and avoid the mismatch between the distribution of the predicted and expected probabilistic values [2]. We add a suffix "+C" to SolarFlareNet and LSTM to indicate the calibrated version of the model.

Similar to [122, 2], to evaluate the performance of a probabilistic and calibrated probabilistic forecasting behaviour, we use the Brier Score (BS)[177] and Brier Skill Score (BSS)[177], defined as follows:

$$\mathrm{BS} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{2.8}$$

$$\mathrm{BSS} = 1 - \frac{BS}{\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2}. \tag{2.9}$$

Where, N is the total number of data sequences each having m consecutive data samples in the test set; y_i denotes the observed probability and $\hat{y}_i$ denotes the predicted probability of the ith test data sequence, respectively; $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$

denotes the mean of all the observed probabilities. The BS values range from 0 to 1, with 0 being a perfect score, whereas the BSS values range from $-\infty$ to 1, with 1 being a perfect score.

Table 2.8 represents the performance comparison of the probabilistic forecasting for SolarFlareNet and LSTM [119]. The table presents the mean BS and BSS values the average of our cross-validation scheme with standard deviations enclosed in parentheses. The best values for BS and BSS values are highlighted in boldface. It can be seen from Table 3.5 that the probabilistic forecasting results of our model is performing better than LSTM [119]. Furthermore, the calibrated version of a model is better than the model without calibration. Overall, SolarFlarNet+C performs better in terms of BS and BSS. Similarly, Tables 2.9 and 2.10 show the results for the 48-hour (and 72-hour, respectively). It also can be seen that SolarFlareNet outperforms the LSTM[119] which is line with the result for 24-hour predictions. It should also be noticed that 48- and 72-hour predictions have less performance than 24-hour. This is expected because the longer the prediction ahead, the less performance due to data deviation over time.

Table 2.8 Probabilistic Forecasting Results of SolarFlareNet and LSTM [119] With and Without Calibration for 24-hour.

		≥M5.0 class	≥M class	≥C class
BS	LSTM	0.315 (0.062)	0.324 (0.087)	0.401 (0.066)
	LSTM+C	0.302 (0.073)	0.319 (0.039)	0.389 (0.086)
	SolarFlareNet	0.268 (0.047)	0.277 (0.062)	0.321 (0.065)
	SolarFlareNet+C	**0.236** (0.032)	**0.247** (0.048)	**0.293** (0.042)
BSS	LSTM	0.395 (0.104)	0.382 (0.124)	0.377 (0.126)
	LSTM+C	0.426 (0.120)	0.399 (0.109)	0.386 (0.115)
	SolarFlareNet	0.516 (0.055)	0.462 (0.134)	0.452 (0.093)
	SolarFlareNet+C	**0.587** (0.037)	**0.503** (0.086)	**0.498** (0.076)

Table 2.9 Probabilistic Forecasting Results of SolarFlareNet and LSTM [119] With and Without Calibration for 48-hour.

		≥M5.0 class	≥M class	≥C class
BS	LSTM	0.355 (0.072)	0.375 (0.066)	0.463 (0.062)
	LSTM+C	0.324 (0.078)	0.367 (0.092)	0.427 (0.084)
	SolarFlareNet	0.301 (0.051)	0.346 (0.059)	0.359 (0.052)
	SolarFlareNet+C	**0.289** (0.042)	**0.328** (0.042)	**0.344** (0.039)
BSS	LSTM	0.388 (0.092)	0.366 (0.102)	0.351 (0.122)
	LSTM+C	0.417 (0.122)	0.372 (0.100)	0.376 (0.180)
	SolarFlareNet	0.497 (0.073)	0.464 (0.093)	0.447 (0.079)
	SolarFlareNet+C	**0.571** (0.047)	**0.499** (0.089)	**0.485** (0.088)

Table 2.10 Probabilistic Forecasting Results of SolarFlareNet and LSTM [119] With and Without Calibration for 72-hour.

		≥M5.0 class	≥M class	≥C class
BS	LSTM	0.348 (0.039)	0.388 (0.069)	0.474 (0.101)
	LSTM+C	0.336 (0.058)	0.382 (0.104)	0.438 (0.093)
	SolarFlareNet	0.321 (0.063)	0.353 (0.076)	0.366 (0.071)
	SolarFlareNet+C	**0.294** (0.050)	**0.338** (0.055)	**0.351** (0.050)
BSS	LSTM	0.493 (0.088)	0.351 (0.051)	0.344 (0.106)
	LSTM+C	0.521 (0.051)	0.367 (0.092)	0.401 (0.088)
	SolarFlareNet	0.502 (0.058)	0.455 (0.077)	0.453 (0.094)
	SolarFlareNet+C	**0.566** (0.049)	**0.480** (0.062)	**0.478** (0.064)

CHAPTER 3

PREDICTION OF SOLAR ENERGETIC PARTICLES

3.1 Background and Related Work

Solar eruptions including flares and coronal mass ejections (CMEs) can endanger modern civilization. Solar flares are large bursts of radiation released into space; they appear as sudden and unexpected brightening in the solar atmosphere with a duration ranging from minutes to hours. CMEs are significant discharges of plasma and magnetic fields produced by the solar corona into the interplanetary medium [115]. They are considered to be the largest-scale solar eruptions in the solar system and occur on a quasi-regular basis [46, 176, 94]. Research shows that both flares and CMEs are magnetic events, sharing a similar physical process [73, 23], though more work is performed to understand the correlation between them [182, 92]. Large flares and accompanied CMEs cause solar energetic particles (SEPs). SEPs, composed of electrons, protons and heavy ions, are expedited by magnetic reconnection or shock waves associated with the CMEs [34, 81]. When SEP events are strong, they cause nuclear cascades in the Earth's upper atmosphere and also represent a radiation hazard to equipment in space that is not adequately protected [142, 90, 144].

Active regions (ARs), which manifest complex magnetic geometry and properties [22], are the source of flares and CMEs [46, 167]. The lifetime of ARs ranges from days to months [167]. Recently, researchers combine machine learning with physical parameters derived from vector magnetograms provided by the Helioseismic and Magnetic Imager (HMI) [148] on board the *Solar Dynamics Observatory (SDO)* [135] to predict flares, CMEs, and SEPs. These physical parameters, including magnetic helicity and magnetic flux [109, 149, 130], are part of the vector magnetic

data products, named the Space-weather HMI Active Region Patches (SHARP) [31], produced by the *SDO*/HMI team.

Machine learning (ML) has been popular in predictive analytics for many years. ML is able to learn patterns from historical data and make predictions on unseen or future data [10, 67]. For example, [117] used random forests (RF) and the SHARP parameters to predict the occurrence of a certain class of flares in a given AR within 24 hours. [89] employed machine learning to extract relevant information from photospheric and coronal image data to perform flare prediction. [61] adopted multiple machine learning algorithms including RF, multilayer perceptrons (MLP) and support vector machines (SVM) for flare forecasting. More recently, researchers started to use deep learning (DL), which is a branch of machine learning focusing on the use of deep neural networks, to enhance the learning outcome [67]. [82] designed a convolutional neural network to learn patterns from line-of-sight magnetograms of ARs and used the patterns to forecast flares. [119] adopted a long short-term memory (LSTM) network for flare prediction. [47] employed LSTM and the SHARP parameters to identify solar flare precursors; the authors later extended their work by investigating solar cycle dependence [174]. Similar ML and DL methods have been applied to CME and SEP forecasting. [30] used SVM to predict CMEs; [122] extended their work by adopting recurrent neural networks including LSTM and gated recurrent units. [84] employed SVM and MLP to forecast if flares would be accompanied with CMEs and SEPs.

In this research, we proposed a new deep learning method, specifically a bidirectional long short-term memory (biLSTM) network, for SEP prediction using the *SDO*/HMI vector magnetic data products. We aim to solve two binary prediction problems: (i) predict whether an AR would produce an SEP event given that the AR will produce an M- or X-class flare and a CME associated with the flare (referred to as the FC_S problem); (ii) predict whether an AR would produce an SEP event given

that the AR will produce an M- or X-class flare regardless of whether or not the flare is associated with a CME (referred to as the F_S problem). The proposed biLSTM is an extension of LSTM [75], both of which are well suited for time series forecasting [108, 67]. Unlike LSTM, which works in one direction, biLSTM works back and forth on the input data and then the patterns learned from the two directions are joined together to strengthen the learning outcome. In SEP prediction, the observations and physical parameters associated with ARs form time series, and hence biLSTM is suitable for our study.

The rest of this work is organized as follows. Section 3.2 explains the data and data collection procedure used in our study. Section 3.3 describes our proposed deep learning method. Section 3.4 reports experimental results and discusses extensions of our approach for probabilistic forecasting and calibration. Section 3.5 concludes the research.

3.2 Data

In this work, we adopt the Space-weather HMI Active Region Patches (SHARP) [31] that were produced by the *SDO*/HMI team and released at the end of 2012. These data are available for download, in the data series hmi.sharp, from the Joint Science Operations Center (JSOC).[1] The SHARP data provide physical parameters of active regions (ARs) that have been used to predict flares, CMEs and SEPs [30, 117, 84, 119, 122]. We collected SHARP data samples from the data series, hmi.sharp_cea_720s, using the Python package SunPy [159] at a cadence of 12 minutes. In collecting the data samples, we focused on the 18 physical parameters previously used for SEP prediction [84]. These 18 SHARP parameters include the absolute value of the net current helicity (ABSNJZH), area of strong field pixels in the active region (AREA_AC), mean characteristic twist parameter (MEANALP), mean angle of field

[1]Retrieved on 09/03/2022 from `http://jsoc.stanford.edu/`

from radial (MEANGAM), mean gradient of horizontal field (MEANGBH), mean gradient of total field (MEANGBT), mean gradient of vertical field (MEANGBZ), mean vertical current density (MEANJZD), mean current helicity (MEANJZH), mean photospheric magnetic free energy (MEANPOT), mean shear angle (MEANSHR), sum of flux near polarity inversion line (R_VALUE), sum of the modulus of the net current per polarity (SAVNCPP), fraction of area with shear $> 45^\circ$ (SHRGT45), total photospheric magnetic free energy density (TOTPOT), total unsigned current helicity (TOTUSJH), total unsigned vertical current (TOTUSJZ), and total unsigned flux (USFLUX).

Since the 18 SHARP parameters have different units and scales, we normalized the parameter values using the min-max normalization procedure as done in [122]. Each data sample contains the 18 SHARP parameters. Let p_i^k be the original value of the ith parameter of the kth data sample. Let q_i^k be the normalized value of the ith parameter of the kth data sample. Let min_i be the minimum value of the ith parameter. Let max_i be the maximum value of the ith parameter. Then

$$q_i^k = \frac{p_i^k - min_i}{max_i - min_i}. \tag{3.1}$$

Appropriately labeling the data samples is crucial for machine learning. We surveyed M- and X-class flares that occurred between 2010 and 2021 with identified active regions in the *GOES* X-ray flare catalogs provided by the National Centers for Environmental Information (NCEI). As done in [30], we excluded ARs that were outside $\pm$ 70° of the central meridian because the SHARP parameters cannot be calculated correctly based on the vector magnetograms of the ARs that are near the limb due to foreshortening and projection effects.[2] We also excluded flares with an

[2]Notice that flaring ARs outside $\pm$ 70° of the central meridian may produce eruptive events that have increased probabilities to result in SEPs due to the magnetic connectivity with Earth. Excluding these flaring ARs may reduce the number of SEP events considered in the study. This is a limitation of our approach.

absolute value of the radial velocity of *SDO* being greater than 3500 m s^{-1}, low-quality HMI data as described by [76], and data samples with incomplete SHARP parameters. In this way, we excluded data samples of low quality, and kept qualified data samples of high quality in our study. Furthermore, we collected and extracted information from NASA's Space Weather Database of Notifications, Knowledge, Information (DONKI)[3] to tag, for any given M- or X-class flare, whether it produced a CME and/or SEP event. We cross-checked the flare records in DONKI and *GOES* X-ray flare catalogs to ensure that each flare record was associated with an active region; otherwise the flare record was removed from our study.

We then created two databases of active regions (ARs) for the period between 2010 and 2021. ARs from 2010, 2016, and 2018-2021 were excluded from the study due to the lack of qualified data samples or the absence of SEP events associated with M-/X-class flares and CMEs. Thus, the databases contain ARs from six years, namely 2011-2015 and 2017. In our first database, referred to as the FC_S database, each record corresponds to an AR, contains an M- or X-class flare as well as a CME associated with the flare, and is tagged by whether the flare/CME produce an SEP event. In this database, there are 31 records tagged by "yes" indicating they are associated with SEP events while there are 97 records tagged by "no" indicating they are not associated with SEP events. In our second database, referred to as the F_S database, each record corresponds to an AR, contains an M- or X-class flare, and is tagged by whether the flare produces an SEP event regardless of whether or not the flare initiates a CME. In this database, there are 40 records tagged by "yes" indicating they are associated with SEP events while there are 700 records tagged by "no" indicating they are not associated with SEP events.

[3]Retrieved on 09/03/2022 from `http://kauai.ccmc.gsfc.nasa.gov/DONKI/`

3.3 Methodology

3.3.1 Prediction Tasks

As mentioned in Section 2.1.1, we aim to solve the following two binary prediction problems. [**FC_S problem**] Given a data sample x_t at time point t in an AR where the AR will produce an M- or X-class flare within the next T hours of t and the flare initiates a CME, we predict whether x_t is positive or negative. Predicting x_t to be positive means that the AR will produce an SEP event associated with the flare/CME. Predicting x_t to be negative means that the AR will not produce an SEP event associated with the flare/CME. [**F_S problem**] Given a data sample x_t at time point t in an AR where the AR will produce an M- or X-class flare within the next T hours of t regardless of whether or not the flare initiates a CME, we predict whether x_t is positive or negative. Predicting x_t to be positive means that the AR will produce an SEP event associated with the flare. Predicting x_t to be negative means that the AR will not produce an SEP event associated with the flare. For both of the two binary prediction problems, we consider T ranging from 12 to 72 in 12-hour intervals as frequently considered in the literature [7, 30, 84, 122].

In solving the two binary prediction problems, we first show how to collect and construct positive and negative data samples used in our study. Figure 3.1(a) (Figure 3.1(b), respectively) illustrates how to construct positive (negative, respectively) data samples for the FC_S problem where $T = 24$ hours. Refer to the FC_S database described in Section 5.2, which indicates whether a flaring AR that already produces an M- or X- class flare/CME will initiate an SEP event associated with the flare/CME. For the flaring AR, we collect data samples that are within the $T = 24$ hours prior to the peak time of the flare.

- If the flare/CME are associated with an SEP event, the data samples belong to the positive class and are colored (labeled) by blue as shown in Figure 3.1(a). Thus, for each blue (positive) data sample, there is an M- or X- class flare that is within the next 24 hours of the occurrence time of the data sample, the flare initiates a CME and the flare/CME are associated with an SEP event.

- If the flare/CME are not associated with an SEP event, the data samples belong to the negative class and are colored (labeled) by green as shown in Figure 3.1(b). Thus, for each green (negative) data sample, there is an M- or X- class flare that is within the next 24 hours of the occurrence time of the data sample, the flare initiates a CME but the flare/CME are not associated with an SEP event.

Constructing positive and negative data samples for the F_S problem is done similarly and its description is omitted.

Table 3.1 Numbers of Positive and Negative Data Samples Constructed for Different Hours for the FC_S and F_S Problems, Respectively

		12 hr	24 hr	36 hr	48 hr	60 hr	72 hr
FC_S	Positive	994	2017	3055	4143	5221	6336
	Negative	2952	5522	7864	9976	11687	13135
F_S	Positive	1260	2561	3863	5207	6517	7864
	Negative	19593	31534	40619	48189	54718	59821

Table 3.1 shows the numbers of positive and negative data samples constructed for the FC_S and F_S problems, respectively. Consider the FC_S problem. The positive and negative data samples are constructed based on the 31 records tagged by "yes" and 97 records tagged by "no" in the FC_S database described in Section 5.2. When $T = 24$ hours and the cadence is 12 minutes, one would expect the total number of positive data samples to be 24 hr $\times$ 60 minutes/hr $\times$ (1/12 minutes) $\times$ 31 = 3720, and the total number of negative data samples to be 24 hr $\times$ 60 minutes/hr $\times$ (1/12 minutes) $\times$ 97 = 11640. However, the total number of positive (negative, respectively) data samples is 2017 (5522, respectively). This happens because we removed many data samples of low quality as described in Section 5.2. If a gap occurs in the middle of a time series due to the removal, we use a zero-padding strategy as done in [122] to create a synthetic data sample to fill the gap. The synthetic data sample has zero values for all the 18 SHARP parameters. The synthetic data sample is added after the normalization of the SHARP parameter values, and hence the synthetic data sample does not affect the normalization procedure.

After explaining how to construct the positive and negative data samples, we now show how to solve the binary prediction problems. Consider again the FC_S problem where $T = 24$ hours. Here we want to predict whether a given test data sample x_t at time point t is positive (blue) or negative (green) given that there will be an M- or X- class flare within the next 24 hours of t and the flare initiates a CME. If there is an SEP event associated with the flare/CME, and we predict x_t to be positive (blue), then this is a correct prediction as illustrated in Figure 3.1(c). If there is an SEP event associated with the flare/CME, but we predict x_t to be negative (green), then this is a wrong prediction as illustrated in Figure 3.1(e). On the other hand, if there is no SEP event associated with the flare/CME, and we predict x_t to be negative (green), then this is a correct prediction as illustrated in Figure 3.1(d). If there is no SEP event associated with the flare/CME, but we predict x_t to be positive (blue), then this is a wrong prediction as illustrated in Figure 3.1(f). The F_S problem is solved similarly. In the following subsection we describe how to train our model and use the trained model to make predictions.

3.3.2 Prediction Method

We consider one of recurrent neural networks (RNNs) that is called long short-term memory (LSTM) [75, 67] to build our model. LSTM has shown good results in solar eruption prediction [119, 122]. We create a model using bidirectional LSTM (biLSTM). Generally, a bidirectional RNN [150] functions by duplicating the initial recurrent layer in the network to obtain two layers so that one layer uses the input as is and the other duplicated layer uses the input in a reverse order. This design allows biLSTM to discover additional patterns that cannot be found by LSTM with only one recurrent layer [153]. In addition, the data used in our study is time series and biLSTM has shown an improvement over LSTM for general time series forecasting

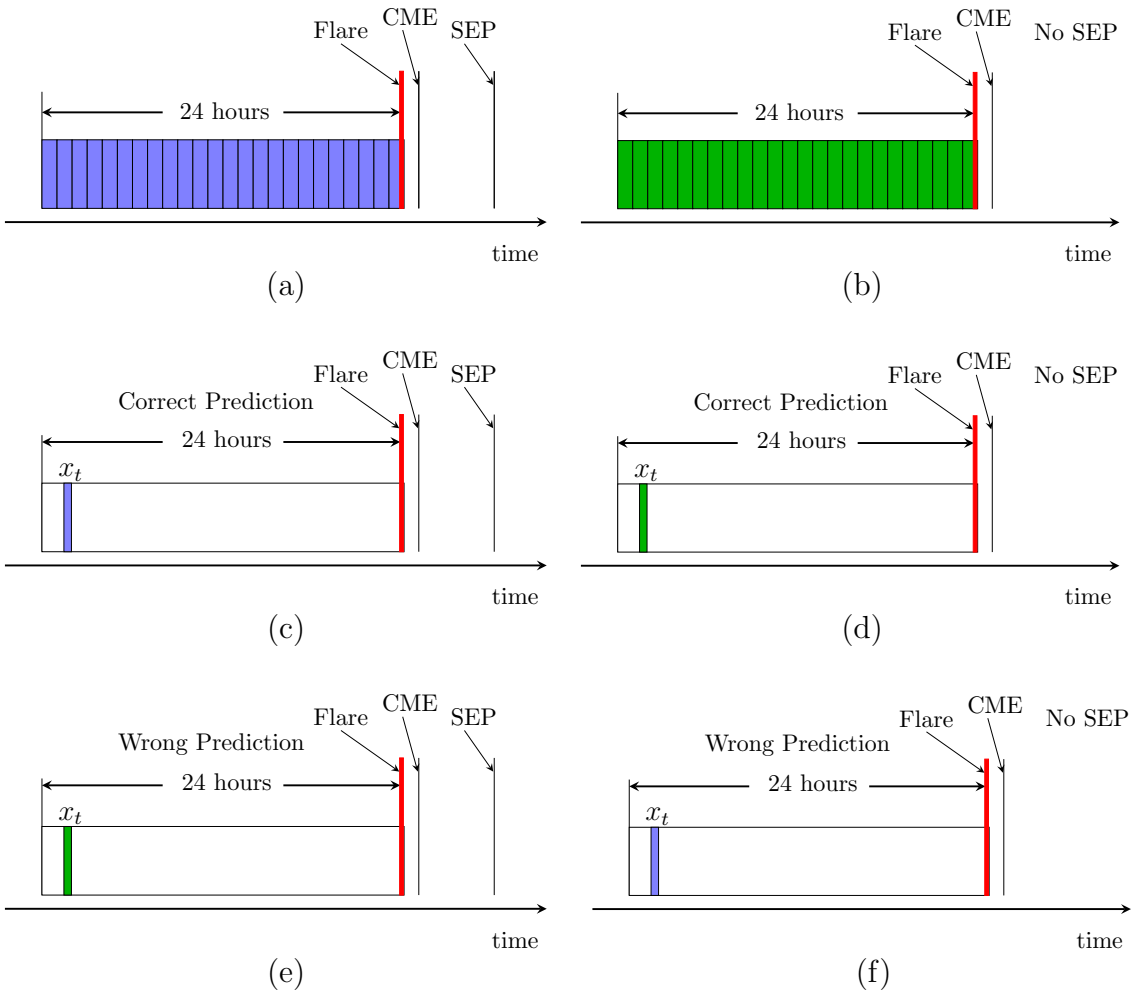


Figure 3.1 Collecting and constructing positive and negative data samples on a flaring AR for the FC_S problem where $T = 24$ hours and making predictions based on the collected data samples. The data samples are collected at a cadence of 12 minutes. Each rectangular box corresponds to 1 hour and contains 5 data samples. The red vertical line shows the peak time of an M- or X- class flare.

[11, 91]. As our experimental results show later, biLSTM also outperforms LSTM in SEP prediction.

Figure 3.2 presents the architecture of our neural network, which accepts as input a data sequence with m consecutive data samples. (In the study presented here, m is set to 10.) The neural network consists of a biLSTM layer configured with 400 neurons. In addition, the neural network contains an attention layer motivated by [67] to direct the network to focus on important information and characteristics of input data samples. The attention layer is designed to map and capture the alignment

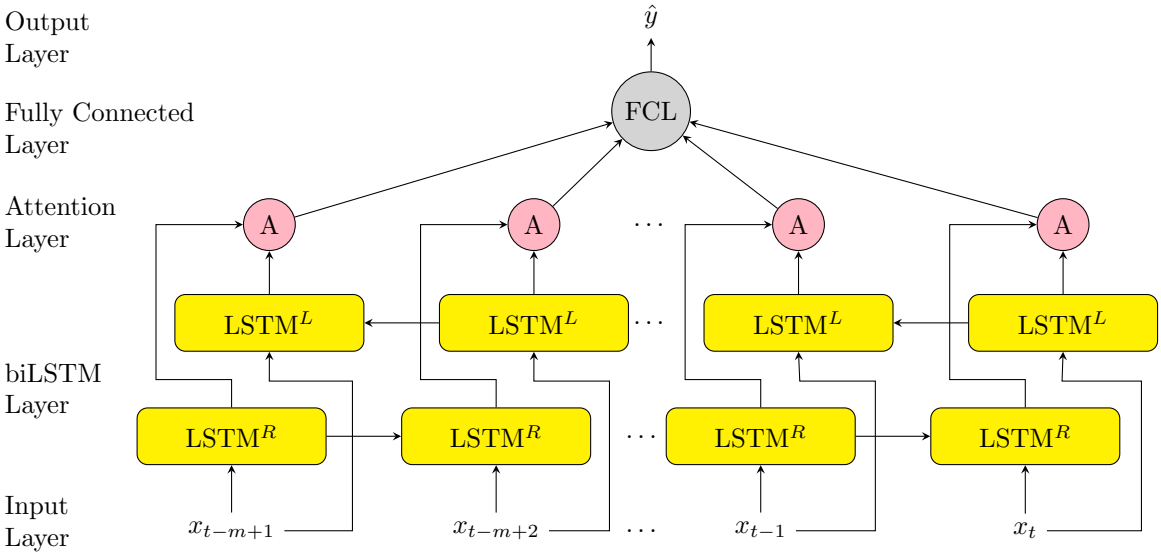


Figure 3.2 Architecture of the proposed biLSTM network. Yellow boxes represent biLSTM cells. These cells are connected to an attention layer (A) that contains m neurons, which are connected to a fully connected layer (FCL). (In the study presented here, m is set to 10.) During testing/prediction, the input to the network is a test data sequence with m consecutive data samples x_{t-m+1}, x_{t-m+2} $\ldots x_{t-1}$, x_t where x_t is the test data sample at time point t. The trained biLSTM network predicts the label (color) of the test data sequence, more precisely the label (color) of x_t. The output layer of the biLSTM network calculates a probability ($\hat{y}$) between 0 and 1. If $\hat{y}$ is greater than or equal to a threshold, which is set to 0.5, the biLSTM network outputs 1 and predicts x_t to be positive, *i.e.*, predicts the label (color) of x_t to be blue; see Figure 3.1. Otherwise, the biLSTM network outputs 0 and predicts x_t to be negative, *i.e.*, predicts the label (color) of x_t to be green; see Figure 3.1.

between the input and output by calculating a weighted sum for input data sequences. Specifically, the attention context vector for the output $\hat{y}_i$, denoted $\boldsymbol{CV}_i$, is calculated as follows:

$$\boldsymbol{CV}_i = \sum_{j=1}^{m} \boldsymbol{W}_{i,j} \boldsymbol{H}_j, \tag{3.2}$$

where m is the input sequence length, $\boldsymbol{H}_j$ is the hidden state corresponding to the input data sample x_j and $\boldsymbol{W}$ contains weights applied to the hidden state. $\boldsymbol{W}$ is computed by a softmax function as follows:

$$\boldsymbol{W}_{i,j} = \frac{e^{S_{i,j}}}{\sum_{k=1}^{m} e^{S_{i,k}}}. \tag{3.3}$$

Here $S_{i,j}$ is a score function calculated as follows:

$$S_{i,j} = \boldsymbol{V} \times \tanh(\boldsymbol{W}^{'}(\boldsymbol{S}_i, \boldsymbol{H}_j)), \tag{3.4}$$

where $\tanh(\cdot)$ is the hyperbolic tangent function, $\boldsymbol{S}_i$ is the output state corresponding to the output $\hat{y}_i$, $\boldsymbol{V}$ and $\boldsymbol{W}^{'}$ are weight matrices learned by the neural network. The attention layer passes its resulting vector to a fully connected layer.

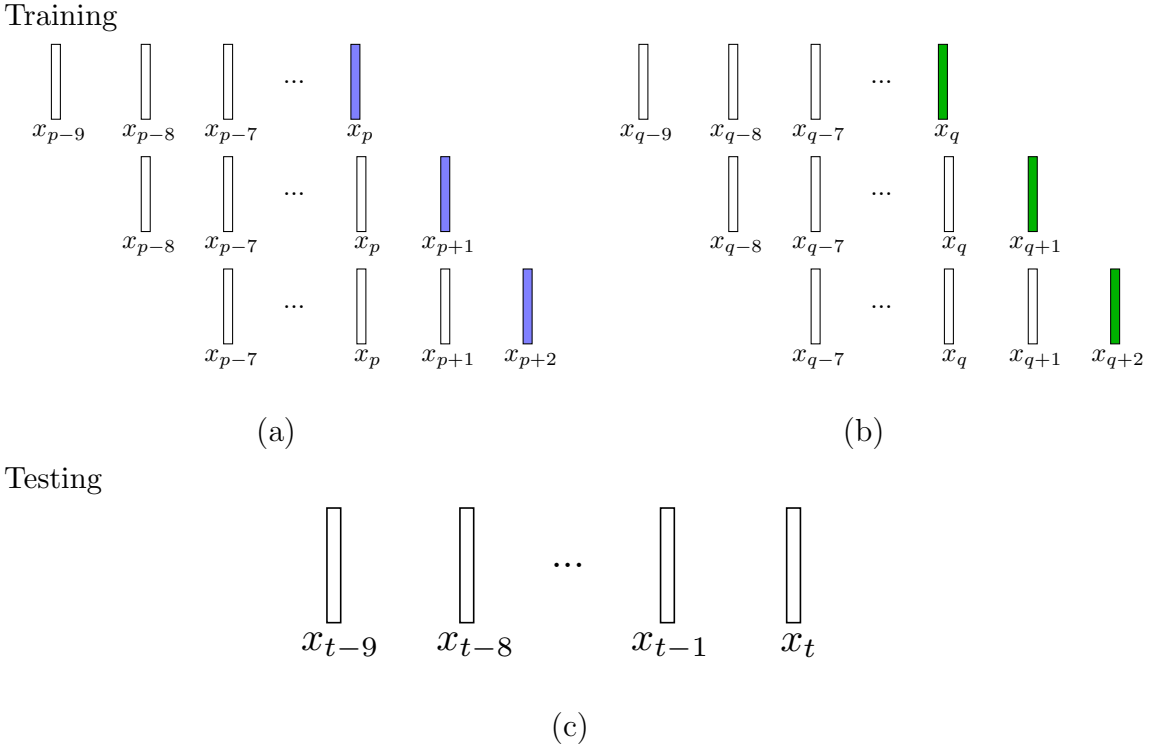


Figure 3.3 Example data sequences used to train and test our biLSTM network where each data sequence contains 10 consecutive data samples. (a) Three positive training data sequences taken from a flaring AR. (b) Three negative training data sequences taken from a flaring AR. In (a) and (b), the label (color) of a training data sequence is defined to be the label (color) of the last data sample in the training data sequence while the labels (colors) of the other nine data samples in the training data sequence are ignored. (c) A test data sequence formed for predicting the label (color) of the last data sample x_t in a flaring AR.

During training, our biLSTM network takes as input overlapping data sequences where each data sequence contains $m = 10$ consecutive data samples. The label (color) of a training data sequence is defined to be the label (color) of the last (*i.e.*,

10th) data sample in the data sequence while the labels (colors) of the other nine data samples in the data sequence are ignored. Thus, if the 10th data sample is positive (blue), then the training data sequence is positive; if the 10th data sample is negative (green), then the training data sequence is negative. We feed one training data sequence at a time to our biLSTM network when training the model. Figure 3.3(a) illustrates three positive data sequences used to train our biLSTM model. Figure 3.3(b) illustrates three negative data sequences used to train our biLSTM model.

The loss function used in our biLSTM model is the weighted binary cross-entropy (WBCE) [67, 122]. Let N denote the total number of data sequences each having m consecutive data samples in the training set. Let w_0 denote the weight for the positive class (*i.e.*, minority class) and let w_1 denote the weight for the negative class (*i.e.*, majority class). The weights are calculated based on the ratio of majority and minority class sizes with more weight assigned to the minority class. Let y_i denote the observed probability of the ith data sequence; y_i is 1 if the ith data sequence is positive and 0 if the ith data sequence is negative. Let $\hat{y}_i$ denote the predicted probability of the ith data sequence. The WBCE, calculated as follows, is suitable for imbalanced datasets such as those tackled here where the negative class has more data samples than the positive class; see Table 3.1.

$$\text{WBCE} = \sum_{i=1}^{N} w_0 y_i \log(\hat{y}_i) + w_1(1 - y_i) \log(1 - \hat{y}_i). \tag{3.5}$$

We configure the network to use a fraction (1/10) of the training set as the internal validation subset. We employ progressive learning with early stopping and adopt the strategy of saving the highest performing model during the iterative learning process. The performance of a model is measured by the WBCE on the internal validation subset where the smaller the WBCE is, the better performance the model has. In each iteration, the process checks the performance of the models in the current

and previous iterations to decide which model to use for the next iteration. If the model in the current iteration has better performance, the process copies its weights as starting weights for the next iteration; otherwise, it copies the weights of the model in the previous iteration as starting weights for the next iteration. This progressive process improves the weights of the network's hidden layers and as a result the overall performance of the network is also improved. In addition, during the iterations, if the performance of the network degrades, the process stops and selects the highest performing model it identifies within the iterations.

During testing/prediction, we are given a test data sample x_t and our biLSTM model will predict the label (color) of x_t, *i.e.*, predict whether x_t is positive or negative. We pack the $m-1$ data samples preceding x_t, namely x_{t-m+1}, x_{t-m+2}, ..., x_{t-1}, along with x_t into a test data sequence with m consecutive data samples and feed this test data sequence to our biLSTM model as shown in the input layer in Figure 3.2. Figure 3.3(c) illustrates a test data sequence where m is 10. The output layer of our biLSTM model calculates a probability between 0 and 1 for the test data sequence. We compare the probability with a threshold, which is set to 0.5. If the probability is greater than or equal to the threshold, our biLSTM model outputs 1 indicating the test data sequence, more precisely the test data sample x_t, is positive; otherwise our model outputs 0 indicating the test data sequence, more precisely x_t, is negative.

3.4 Results

3.4.1 Performance Metrics and Experiment Setup

We conducted a series of experiments to evaluate the performance of the proposed method and compare it with related machine learning methods. For the data sample x_t at time point t, we define:

- x_t to be true positive (TP) if our model (network) predicts that x_t is positive and x_t is indeed positive, *i.e.*, an SEP event will be produced with respect to x_t;

- x_t to be false positive (FP) if our model predicts that x_t is positive while x_t is actually negative, *i.e.*, no SEP event will be produced with respect to x_t;

- x_t to be true negative (TN) if our model predicts x_t is negative and x_t is indeed negative;

- x_t to be false negative (FN) if our model predicts x_t is negative while x_t is actually positive.

We also use TP (FP, TN, FN, respectively) to denote the total number of true positives (false positives, true negatives, false negatives, respectively) produced by a method.

The following performance metrics are used in our study:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{3.6}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{3.7}$$

$$\text{Balanced Accuracy (BACC)} = \frac{1}{2}\left(\frac{\text{TP}}{\text{TP} + \text{FN}} + \frac{\text{TN}}{\text{TN} + \text{FP}}\right), \tag{3.8}$$

$$\text{Heidke Skill Score (HSS)} = \frac{2 \times (\text{TP} \times \text{TN} - \text{FP} \times \text{FN})}{(\text{TP} + \text{FN}) \times (\text{FN} + \text{TN}) + (\text{TP} + \text{FP}) \times (\text{FP} + \text{TN})}, \tag{3.9}$$

$$\text{True Skill Statistics (TSS)} = \frac{\text{TP}}{\text{TP} + \text{FN}} - \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{3.10}$$

BACC [64] is an accuracy measure mainly for imbalanced datasets. HSS [74] and TSS [27] are commonly used for flare, CME and SEP predictions [27, 61, 84, 119, 122]. HSS ranges from $-\infty$ to $+1$. The higher HSS value a method has, the better performance the method achieves. TSS ranges from -1 to $+1$. Like HSS,

the higher TSS value a method has, the better performance the method achieves. In addition, we use the weighted area under the curve (WAUC) [20] in our study. The area under the curve (AUC) in a receiver operating characteristic (ROC) curve [125] indicates how well a method is capable of distinguishing between two classes in binary prediction with the ideal value of one. When calculating the AUC, we do not distinguish between the accuracy on the minority class (positive class) and the accuracy on the majority class (negative class). In contrast, when calculating the WAUC, which is an extension of the AUC and mainly for imbalanced datasets like those tackled here, the accuracy on the minority class has a larger contribution to the overall performance of a model than the accuracy on the majority class. As a consequence, we assign more weight to the accuracy on the minority class where the weight is defined to be the ratio of the sizes of the minority and majority classes. All the metrics mentioned above are calculated using the confusion matrices obtained from the cross-validation (CV) scheme. With CV, we train a model using a subset of data, called the training set, and test the model using another subset of data, called the test set, where the training set and test set are disjoint. We consider six years, namely 2011-2015 and 2017, as mentioned in Section 5.2. Data samples from each year in turn are used for testing in a run and data samples from all the other five years together are used for training in the run. There are six years, and hence there are six runs in total. For each performance metric, the mean and standard deviation over the six runs are calculated and recorded.

3.4.2 Parameter Ranking and Selection

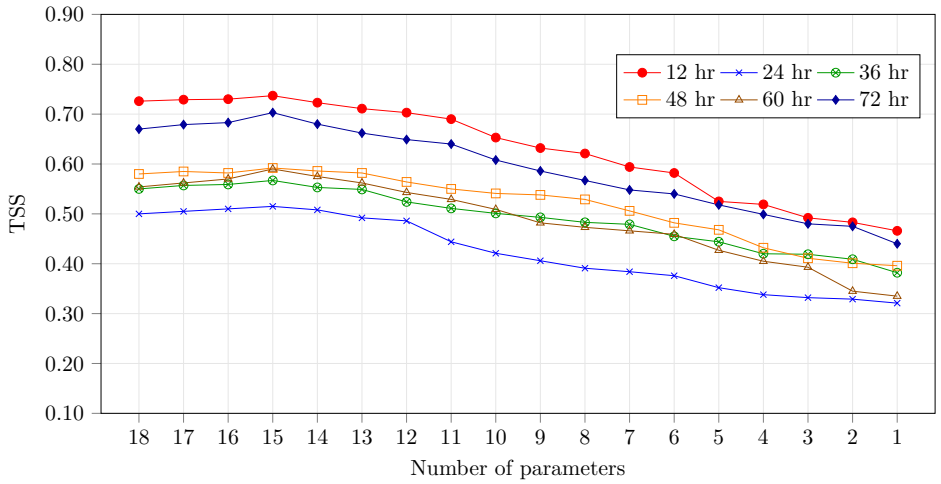
We first assessed the importance of the 18 SHARP parameters described in Section 5.2 to understand which parameters are the most important ones with the greatest predictive power by utilizing a parameter ranking method, called Stability Selection [127]. This method is based on the LASSO (Least Absolute Shrinkage and Selection

Operator) algorithm [161]. Table 3.2 presents the rankings of the parameters with respect to T = 12, 24, 36, 48, 60 and 72 for the FC_S and F_S problems, respectively. The parameter ranked first is the most important one while the parameter ranked 18th is the least important one. ABSNJZH is ranked consistently high for the FC_S problem while SAVNCPP and TOTUSJH are ranked high for the F_S problem. AREA_ACR, TOTUSJZ and USFLUX are ranked consistently low for both of the FC_S and F_S problems.

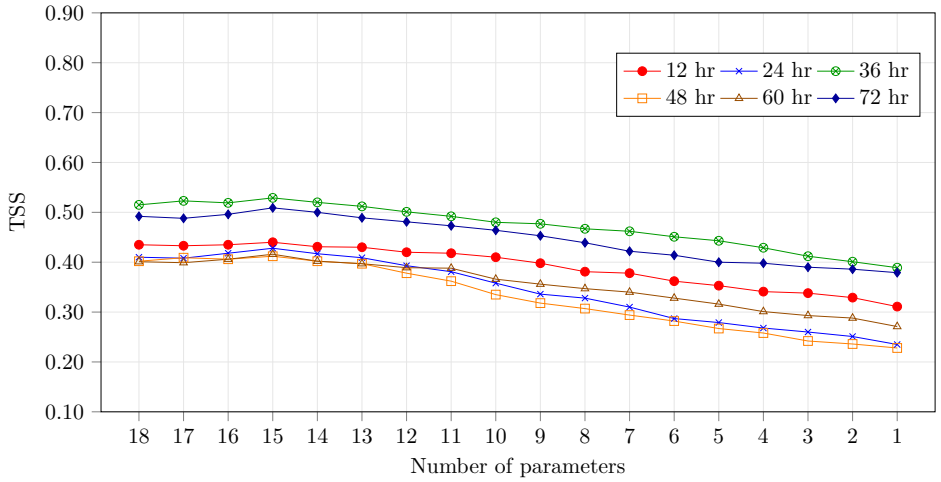
Table 3.2 Importance Rankings of the 18 SHARP Parameters Used in Our Study for the FC_S and F_S Problems, Respectively

SHARP	12 hr		24 hr		36 hr		48 hr		60 hr		72 hr	
Keyword	FC_S	F_S	FC_S	F_S	FC_S	F_S	FC_S	F_S	FC_S	F_S	FC_S	F_S
ABSNJZH	3	3	1	4	1	10	1	10	1	2	5	1
AREA_ACR	17	16	17	16	17	16	17	16	16	16	16	16
MEANALP	13	15	3	15	3	15	3	15	2	15	6	15
MEANGAM	4	14	4	14	4	14	4	14	4	14	8	14
MEANGBH	5	13	5	13	5	13	5	13	14	13	14	7
MEANGBT	6	12	6	12	6	12	6	12	13	12	13	13
MEANGBZ	7	11	7	3	7	4	7	4	12	11	4	4
MEANJZD	8	10	8	11	8	11	8	11	11	10	12	12
MEANJZH	2	9	2	10	2	5	2	5	10	9	11	10
MEANPOT	10	8	10	9	10	9	10	9	9	8	1	11
MEANSHR	11	7	11	8	11	8	11	8	8	7	10	3
R_VALUE	12	6	12	7	12	1	12	3	7	6	2	2
SAVNCPP	1	2	13	2	13	3	13	1	6	1	3	5
SHRGT45	14	5	14	6	14	7	14	7	5	5	9	9
TOTPOT	15	4	15	5	15	6	15	6	15	4	15	8
TOTUSJH	9	1	9	1	9	2	9	2	3	3	7	6
TOTUSJZ	16	17	16	17	16	17	16	17	17	17	17	17
USFLUX	18	18	18	18	18	18	18	18	18	18	18	18

We then used the recursive parameter elimination algorithm [37] in combination with our biLSTM model to select a set of parameters that achieves the best performance where the performance is measured by TSS. The parameter elimination algorithm is an interactive procedure. It selects parameters by recursively considering smaller and smaller sets of parameters where the least important parameters are



(a)



(b)

Figure 3.4 Parameter selection results for (a) the FC_S problem, and (b) the F_S problem.

successively pruned from the current set of parameters. Figure 3.4 presents the parameter selection results for the FC_S and F_S problems, respectively. It can be seen from the figure that using the top 15 most important parameters achieves the best performance for both of the FC_S and F_S problems. When using the top k, $1 \leq k \leq 14$, most important parameters, the less parameters we use, the worse performance our model achieves. Using the top-ranked, most important parameter alone would yield a lower TSS than using all the top 15 most important parameters together. In subsequent experiments, we used the top 15 most important parameters

for our biLSTM model. That is, we removed the three least important parameters AREA_ACR, TOTUSJZ and USFLUX from data samples and each data sample contained only the top 15 most important SHARP parameters.

3.4.3 Performance Comparison

Next, we compared our biLSTM network with four related machine learning methods, including multilayer perceptrons (MLP), support vector machines (SVM), random forests (RF) and long short-term memory (LSTM) [119]. These four methods are commonly used to predict solar flares, CMEs and SEPs [30, 117, 61, 84, 47, 119, 122, 174, 4].

MLP [145, 12] is a feed-forward artificial neural network [32] that consists of an input layer, an output layer, and one or more hidden layers. The number of hidden layers is set to 3 with 200 neurons in each hidden layer. SVM [52] is trained with the Radial Basis Function (RBF) kernel and the cache size is set to 20000 to speed the training process. RF [33] is an ensemble algorithm that has two hyperparameters for performance tuning: m (the number of SHARP magnetic parameters randomly selected and used to split a node in a tree of the forest) and n (the number of trees to grow). We set m to 2 and n to 500. The implementation of LSTM follows that described in [122]. The hyperparameters not specified here are set to their default values provided by the scikit-learn library in Python [134].

As done in biLSTM, we used the recursive parameter elimination algorithm [37] to identify and select the best parameters for the four related machine learning methods based on the importance rankings of the 18 SHARP parameters in Table 3.2 for the FC_S and F_S problems, respectively. Our experiments showed that, like biLSTM, using the top 15 most important parameters achieved the best performance for the four related machine learning methods. Consequently, we used the top 15

most important parameters for the four machine learning methods in the experimental study.

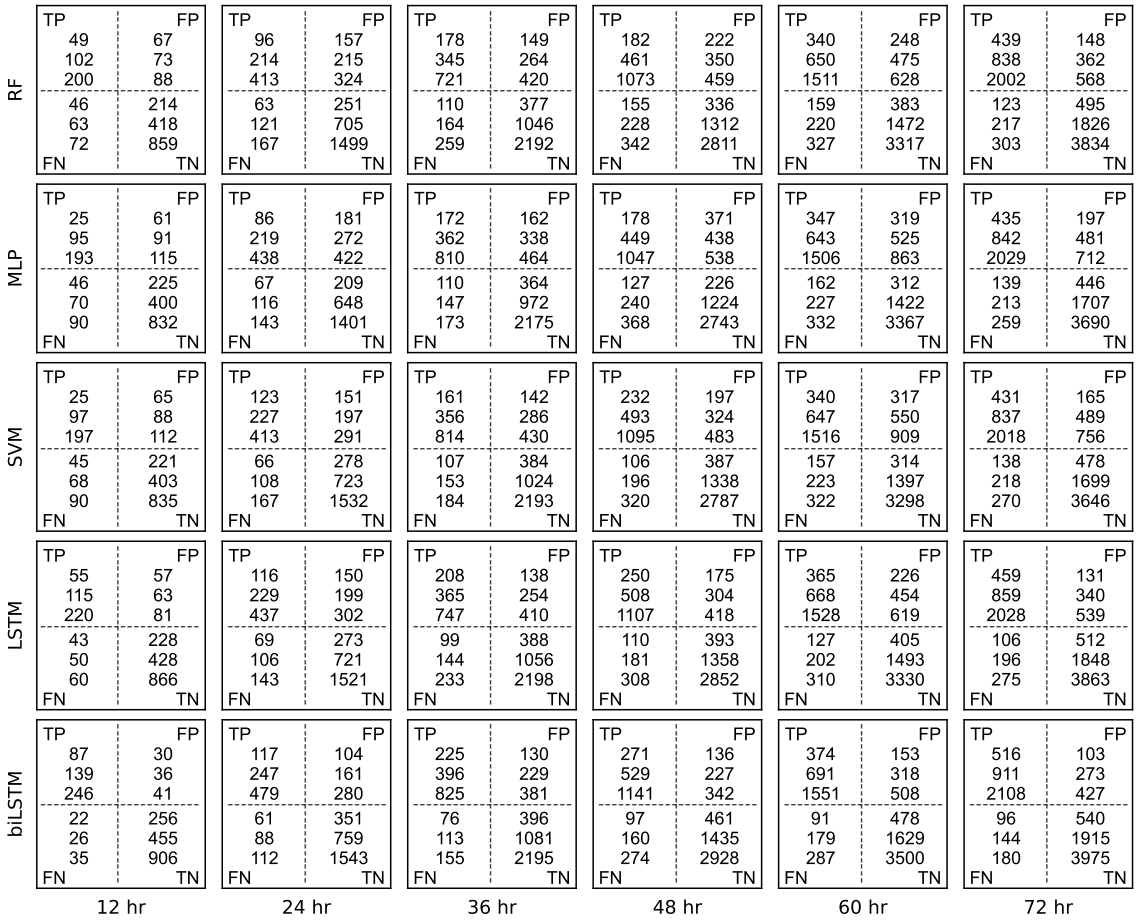


Figure 3.5 Confusion matrices of RF, MLP, SVM, LSTM and biLSTM for the FC_S problem. For each T, T = 12, 24, 36, 48, 60, 72, and each machine learning method, the figure shows the minimum, average, maximum (displayed from top to bottom) TP, FN, TN, and FP from the six runs based on our cross-validation scheme.

Figures 3.5 and 3.6 present the confusion matrices of the five machine learning methods (RF, MLP, SVM, LSTM, biLSTM) for the FC_S and F_S problems, respectively. For each T, T = 12, 24, 36, 48, 60, 72, and each machine learning method, the figures show the minimum, average, maximum (displayed from top to bottom) TP, FN, TN, and FP from the six runs based on our cross-validation scheme. For example, refer to T = 12 and biLSTM in Figure 3.5. The minimum (maximum,

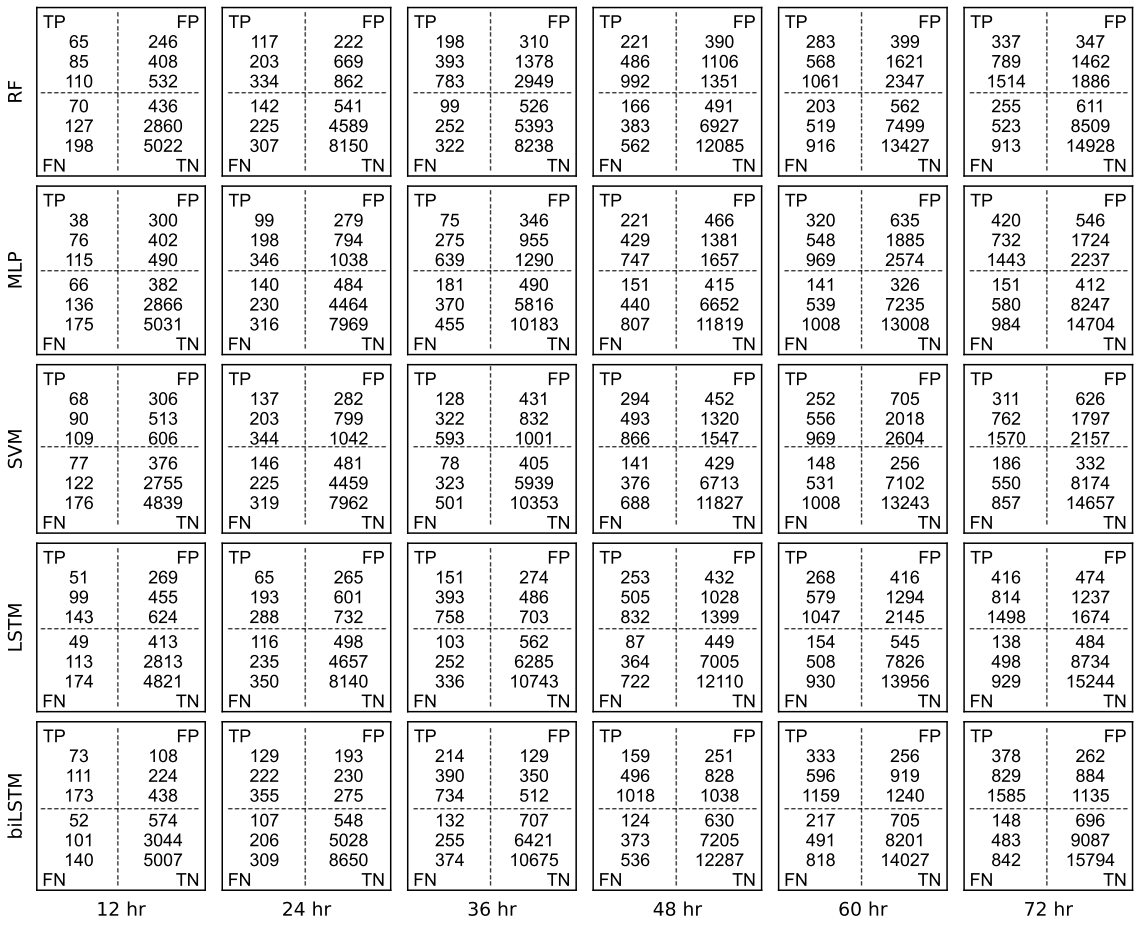


Figure 3.6 Confusion matrices of RF, MLP, SVM, LSTM and biLSTM for the F_S problem. For each T, T = 12, 24, 36, 48, 60, 72, and each machine learning method, the figure shows the minimum, average, maximum (displayed from top to bottom) TP, FN, TN, FP, respectively from the six runs based on our cross-validation scheme.

respectively) TP obtained by biLSTM from the six runs is 87 (246, respectively); the average TP over the six runs is 139. It can be seen from Figures 3.5 and 3.6 that the average TN values are much larger than the average FP values for both of the FC_S and F_S problems. This happens because there are many negative training data samples in our datasets (see Table 3.1). As a consequence, the machine learning methods gain sufficient knowledge about the negative data samples and hence can detect them relatively easily. For the FC_S problem, the average TP values (TN values, respectively) are consistently larger than the average FN values (FP

values, respectively), indicating that the machine learning methods can solve the FC_S problem reasonably well. For the F_S problem, the average TP values are close to, or even smaller than, the average FN values in many cases, suggesting that the machine learning methods have difficulty in detecting positive data samples. This is understandable given that there are much fewer positive training data samples than negative training data samples for the F_S problem (see Table 3.1).

Tables 3.3 and 3.4 compare the performance of the five machine learning methods for the FC_S and F_S problems, respectively. The tables present the mean performance metric values averaged over the six runs based on our cross-validation scheme with standard deviations enclosed in parentheses. Best average metric values are highlighted in boldface. It can be seen from Tables 3.3 and 3.4 that our biLSTM network outperforms the four related machine learning methods in terms of BACC, HSS, TSS and WAUC. Furthermore, the five machine learning methods generally perform better in solving the FC_S problem than in solving the F_S problem. This result indicates that one can predict SEP events more accurately when active regions will produce both flares and associated CMEs. Using flare information alone to predict SEP events is harder and would produce less reliable prediction results.

3.4.4 Probabilistic Forecasting and Calibration

The five machine learning methods (RF, MLP, SVM, LSTM, biLSTM) studied here are inherently probabilistic forecasting models in the sense that they calculate a probability between 0 and 1. We compare the probability with a threshold, which is set to 0.5, to determine the output produced by each machine learning method. The output is either 1 or 0 (see Figure 3.2), and hence each method is essentially a binary prediction model. In addition to comparing the methods used as binary prediction models, we also compare the methods used as probabilistic forecasting models, where the output produced by each model is interpreted as follows. [**FC_S problem**] Given

Table 3.3 Performance Comparison of RF, MLP, SVM, LSTM and biLSTM Based on Our Cross-Validation Scheme for the FC_S Problem

		12 hr	24 hr	36 hr	48 hr	60 hr	72 hr
Recall	RF	0.593 (0.106)	0.617 (0.120)	0.660 (0.078)	0.632 (0.134)	0.721 (0.082)	0.767 (0.095)
	MLP	0.542 (0.175)	0.617 (0.149)	0.679 (0.110)	0.618 (0.143)	0.711 (0.088)	0.766 (0.093)
	SVM	0.543 (0.175)	0.666 (0.088)	0.663 (0.125)	0.686 (0.123)	0.714 (0.090)	0.761 (0.093)
	LSTM	0.658 (0.113)	0.663 (0.099)	0.699 (0.064)	0.714 (0.100)	0.745 (0.071)	0.788 (0.089)
	biLSTM	**0.825** (0.058)	**0.710** (0.112)	**0.758** (0.064)	**0.748** (0.086)	**0.777** (0.077)	**0.843** (0.058)
Precision	RF	0.554 (0.113)	0.483 (0.151)	0.556 (0.163)	0.532 (0.165)	0.548 (0.155)	0.661 (0.171)
	MLP	0.478 (0.153)	0.432 (0.166)	0.501 (0.155)	0.471 (0.161)	0.524 (0.164)	0.605 (0.161)
	SVM	0.486 (0.168)	0.520 (0.132)	0.543 (0.158)	0.574 (0.163)	0.514 (0.176)	0.603 (0.175)
	LSTM	0.609 (0.102)	0.516 (0.142)	0.581 (0.158)	0.599 (0.155)	0.569 (0.153)	0.680 (0.166)
	biLSTM	**0.777** (0.061)	**0.587** (0.148)	**0.619** (0.149)	**0.669** (0.155)	**0.656** (0.160)	**0.739** (0.133)
BACC	RF	0.707 (0.045)	0.672 (0.047)	0.719 (0.036)	0.688 (0.051)	0.713 (0.030)	0.790 (0.043)
	MLP	0.663 (0.061)	0.641 (0.062)	0.690 (0.049)	0.641 (0.051)	0.691 (0.046)	0.752 (0.023)
	SVM	0.668 (0.070)	0.706 (0.022)	0.709 (0.045)	0.730 (0.032)	0.687 (0.053)	0.753 (0.029)
	LSTM	0.751 (0.047)	0.704 (0.035)	0.744 (0.029)	0.750 (0.031)	0.734 (0.025)	0.807 (0.041)
	biLSTM	**0.868** (0.029)	**0.757** (0.040)	**0.784** (0.029)	**0.796** (0.037)	**0.795** (0.032)	**0.852** (0.021)
HSS	RF	0.404 (0.091)	0.315 (0.100)	0.406 (0.088)	0.354 (0.100)	0.386 (0.063)	0.545 (0.099)
	MLP	0.314 (0.119)	0.249 (0.123)	0.343 (0.105)	0.259 (0.098)	0.343 (0.098)	0.468 (0.067)
	SVM	0.325 (0.138)	0.377 (0.063)	0.391 (0.101)	0.431 (0.080)	0.332 (0.114)	0.468 (0.083)
	LSTM	0.489 (0.090)	0.373 (0.086)	0.450 (0.084)	0.468 (0.079)	0.423 (0.057)	0.579 (0.099)
	biLSTM	**0.722** (0.057)	**0.481** (0.097)	**0.522** (0.080)	**0.562** (0.086)	**0.551** (0.086)	**0.669** (0.064)
TSS	RF	0.413 (0.090)	0.344 (0.094)	0.437 (0.072)	0.376 (0.101)	0.426 (0.061)	0.579 (0.085)
	MLP	0.326 (0.123)	0.281 (0.125)	0.379 (0.098)	0.283 (0.101)	0.382 (0.092)	0.504 (0.046)
	SVM	0.336 (0.140)	0.413 (0.045)	0.417 (0.091)	0.459 (0.063)	0.374 (0.106)	0.507 (0.057)
	LSTM	0.501 (0.093)	0.407 (0.071)	0.487 (0.059)	0.499 (0.063)	0.468 (0.051)	0.615 (0.082)
	biLSTM	**0.737** (0.057)	**0.515** (0.081)	**0.567** (0.059)	**0.592** (0.073)	**0.590** (0.063)	**0.703** (0.041)
WAUC	RF	0.453 (0.056)	0.375 (0.071)	0.476 (0.048)	0.410 (0.063)	0.459 (0.040)	0.621 (0.057)
	MLP	0.354 (0.033)	0.301 (0.032)	0.415 (0.088)	0.304 (0.072)	0.410 (0.024)	0.543 (0.085)
	SVM	0.361 (0.087)	0.453 (0.068)	0.457 (0.023)	0.503 (0.064)	0.405 (0.040)	0.553 (0.085)
	LSTM	0.541 (0.074)	0.436 (0.071)	0.526 (0.020)	0.535 (0.052)	0.510 (0.051)	0.671 (0.026)
	biLSTM	**0.794** (0.041)	**0.563** (0.039)	**0.609** (0.086)	**0.646** (0.043)	**0.642** (0.048)	**0.764** (0.073)

Table 3.4 Performance Comparison of RF, MLP, SVM, LSTM and biLSTM Based on Our Cross-Validation Scheme for the F_S Problem

		12 hr	24 hr	36 hr	48 hr	60 hr	72 hr
Recall	RF	0.414 (0.099)	0.468 (0.066)	0.592 (0.137)	0.550 (0.123)	0.522 (0.120)	0.590 (0.109)
	MLP	0.367 (0.138)	0.457 (0.098)	0.398 (0.172)	0.501 (0.148)	0.520 (0.153)	0.568 (0.150)
	SVM	0.433 (0.064)	0.471 (0.058)	0.495 (0.189)	0.575 (0.106)	0.518 (0.160)	0.567 (0.145)
	LSTM	0.468 (0.146)	0.456 (0.166)	0.592 (0.155)	**0.591** (0.140)	0.546 (0.155)	0.624 (0.127)
	biLSTM	**0.520** (0.103)	**0.508** (0.122)	**0.597** (0.095)	0.545 (0.197)	**0.548** (0.090)	**0.629** (0.120)
Precision	RF	0.178 (0.052)	0.253 (0.119)	0.267 (0.159)	0.314 (0.144)	0.285 (0.166)	0.370 (0.177)
	MLP	0.164 (0.073)	0.215 (0.106)	0.227 (0.140)	0.254 (0.136)	0.247 (0.138)	0.315 (0.158)
	SVM	0.152 (0.035)	0.218 (0.096)	0.278 (0.130)	0.287 (0.129)	0.234 (0.136)	0.306 (0.154)
	LSTM	0.184 (0.076)	0.252 (0.114)	0.432 (0.124)	0.340 (0.141)	0.329 (0.162)	0.404 (0.161)
	biLSTM	**0.366** (0.159)	**0.473** (0.110)	**0.527** (0.155)	**0.377** (0.200)	**0.405** (0.170)	**0.485** (0.166)
BACC	RF	0.627 (0.033)	0.656 (0.030)	0.684 (0.052)	0.681 (0.038)	0.650 (0.025)	0.702 (0.036)
	MLP	0.599 (0.049)	0.635 (0.031)	0.605 (0.062)	0.634 (0.036)	0.618 (0.040)	0.665 (0.032)
	SVM	0.616 (0.052)	0.641 (0.028)	0.655 (0.039)	0.676 (0.035)	0.605 (0.055)	0.654 (0.052)
	LSTM	0.647 (0.046)	0.653 (0.079)	0.739 (0.061)	0.703 (0.022)	0.677 (0.059)	0.720 (0.029)
	biLSTM	**0.721** (0.046)	**0.714** (0.047)	**0.765** (0.037)	**0.706** (0.076)	**0.708** (0.015)	**0.754** (0.027)
HSS	RF	0.154 (0.031)	0.218 (0.066)	0.239 (0.117)	0.265 (0.077)	0.212 (0.069)	0.311 (0.086)
	MLP	0.128 (0.073)	0.171 (0.049)	0.156 (0.098)	0.181 (0.050)	0.152 (0.048)	0.235 (0.049)
	SVM	0.119 (0.040)	0.175 (0.038)	0.233 (0.082)	0.235 (0.042)	0.128 (0.064)	0.218 (0.087)
	LSTM	0.170 (0.064)	0.216 (0.091)	0.414 (0.110)	0.303 (0.068)	0.270 (0.100)	0.352 (0.078)
	biLSTM	**0.365** (0.137)	**0.418** (0.100)	**0.493** (0.099)	**0.345** (0.159)	**0.353** (0.079)	**0.448** (0.089)
TSS	RF	0.254 (0.066)	0.313 (0.060)	0.368 (0.103)	0.362 (0.075)	0.301 (0.050)	0.405 (0.071)
	MLP	0.198 (0.098)	0.270 (0.062)	0.211 (0.125)	0.269 (0.072)	0.236 (0.080)	0.329 (0.064)
	SVM	0.233 (0.104)	0.282 (0.057)	0.309 (0.078)	0.353 (0.070)	0.210 (0.110)	0.309 (0.104)
	LSTM	0.293 (0.092)	0.305 (0.158)	0.479 (0.122)	0.405 (0.044)	0.353 (0.119)	0.440 (0.058)
	biLSTM	**0.441** (0.093)	**0.428** (0.093)	**0.529** (0.075)	**0.412** (0.151)	**0.416** (0.031)	**0.509** (0.055)
WAUC	RF	0.276 (0.022)	0.338 (0.067)	0.403 (0.021)	0.388 (0.035)	0.330 (0.021)	0.431 (0.042)
	MLP	0.212 (0.084)	0.290 (0.079)	0.226 (0.024)	0.294 (0.034)	0.256 (0.085)	0.359 (0.075)
	SVM	0.254 (0.084)	0.307 (0.048)	0.334 (0.062)	0.381 (0.074)	0.230 (0.060)	0.334 (0.027)
	LSTM	0.319 (0.056)	0.328 (0.047)	0.514 (0.033)	0.432 (0.038)	0.382 (0.073)	0.474 (0.020)
	biLSTM	**0.480** (0.076)	**0.467** (0.034)	**0.574** (0.085)	**0.450** (0.035)	**0.448** (0.087)	**0.552** (0.016)

a data sample x_t at time point t in an AR where the AR will produce an M- or X-class flare within the next T hours of t and the flare initiates a CME, based on the SHARP parameters in x_t and its preceding $m-1$ data samples x_{t-m+1}, x_{t-m+2}, ..., x_{t-1}, we calculate and output a probabilistic estimate of how likely it is that the AR will produce an SEP event associated with the flare and CME. [**F_S problem**] Given a data sample x_t at time point t in an AR where the AR will produce an M- or X-class flare within the next T hours of t regardless of whether or not the flare initiates a CME, based on the SHARP parameters in x_t and its preceding $m-1$ data samples x_{t-m+1}, x_{t-m+2}, ..., x_{t-1}, we calculate and output a probabilistic estimate of how likely it is that the AR will produce an SEP event associated with the flare.

The distribution and behavior of the predicted probabilistic values may not match the expected distribution of observed probabilities in the training data. One can adjust the distribution of the predicted probabilities to better match the expected distribution observed in the training data through calibration. Here, we adopt isotonic regression [104, 147] to adjust the probabilities. Isotonic regression works by fitting a free-form line to a sequence of data points such that the fitted line is non-decreasing (or non-increasing) everywhere, and lies as close to the data points as possible. Calibrated models often produce more accurate results. We add a suffix "+C" to each model to denote the calibrated version of the model.

To quantitatively assess the performance of a probabilistic forecasting model, we adopt the Brier Score (BS) [177] and Brier Skill Score (BSS) [177], defined as follows:

$$\mathrm{BS} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{3.11}$$

$$\mathrm{BSS} = 1 - \frac{BS}{\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2}. \tag{3.12}$$

Here, N is the total number of data sequences each having m consecutive data samples in the test set (see Figure 3.2 where a test data sequence with m consecutive data samples is fed to our biLSTM model); y_i denotes the observed probability and $\hat{y}_i$ denotes the predicted probability of the ith test data sequence; $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$ denotes the mean of all the observed probabilities. The BS values range from 0 to 1, with 0 being a perfect score, whereas the BSS values range from $-\infty$ to 1, with 1 being a perfect score.

Table 3.5 compares the performance of the five machine learning methods used as probabilistic forecasting models for the FC_S and F_S problems, respectively. The table presents the mean BS and BSS values averaged over the six runs based on our cross-validation scheme with standard deviations enclosed in parentheses. Best BS and BSS values are highlighted in boldface. It can be seen from Table 3.5 that the probabilistic forecasting models generally perform better in solving the FC_S problem than in solving the F_S problem, suggesting that F_S is a harder problem and hence the forecasting results for the F_S problem would be less reliable. These findings are consistent with those in Tables 3.3 and 3.4 where the machine learning methods are used as binary prediction models. Furthermore, the calibrated version of a model is better than the model without calibration. Overall, biLSTM+C performs the best among all the models in terms of both BS and BSS.

3.5 Summary

We develop a bidirectional long short-term memory (biLSTM) network for SEP prediction. We consider two prediction tasks. In the first task (FC_S), given a data sample x_t at time point t in an AR where the AR will produce an M- or X-class flare within the next T hours of t and the flare initiates a CME, based on the SHARP parameters in x_t and its preceding $m-1$ data samples x_{t-m+1}, x_{t-m+2}, ..., x_{t-1}, our biLSTM, when used as a binary prediction model, can predict whether the AR

Table 3.5 Probabilistic Forecasting Results of RF, MLP, SVM, LSTM and biLSTM With and Without Calibration for the FC_S and F_S Problems, Respectively

		12 hr	24 hr	36 hr	48 hr	60 hr	72 hr
FC_S							
BS	RF	0.372 (0.083)	0.342 (0.075)	0.365 (0.060)	0.342 (0.092)	0.355 (0.051)	0.269 (0.040)
	RF+C	0.332 (0.070)	0.331 (0.101)	0.324 (0.056)	0.302 (0.085)	0.328 (0.047)	0.252 (0.037)
	MLP	0.362 (0.136)	0.393 (0.080)	0.335 (0.095)	0.315 (0.050)	0.335 (0.083)	0.280 (0.025)
	MLP+C	0.329 (0.124)	0.366 (0.076)	0.309 (0.088)	0.283 (0.050)	0.301 (0.073)	0.255 (0.027)
	SVM	0.359 (0.052)	0.344 (0.037)	0.337 (0.075)	0.353 (0.049)	0.306 (0.087)	0.298 (0.034)
	SVM+C	0.322 (0.047)	0.303 (0.030)	0.297 (0.066)	0.306 (0.035)	0.284 (0.080)	0.267 (0.035)
	LSTM	0.337 (0.064)	0.356 (0.054)	0.300 (0.058)	0.288 (0.032)	0.298 (0.038)	0.274 (0.028)
	LSTM+C	0.271 (0.062)	0.302 (0.045)	0.262 (0.053)	0.244 (0.032)	0.273 (0.035)	0.232 (0.021)
	biLSTM	0.248 (0.028)	0.272 (0.052)	0.270 (0.048)	0.281 (0.040)	0.297 (0.021)	0.279 (0.015)
	biLSTM+C	**0.215** (0.021)	**0.249** (0.038)	**0.223** (0.025)	**0.235** (0.033)	**0.270** (0.043)	**0.202** (0.014)
BSS	RF	0.273 (0.166)	0.316 (0.164)	0.282 (0.124)	0.316 (0.180)	0.325 (0.118)	0.466 (0.067)
	RF+C	0.341 (0.140)	0.343 (0.189)	0.362 (0.115)	0.396 (0.167)	0.340 (0.109)	0.501 (0.062)
	MLP	0.290 (0.262)	0.274 (0.100)	0.320 (0.202)	0.382 (0.082)	0.323 (0.179)	0.436 (0.048)
	MLP+C	0.355 (0.239)	0.325 (0.094)	0.372 (0.187)	0.445 (0.087)	0.392 (0.158)	0.486 (0.052)
	SVM	0.281 (0.128)	0.310 (0.086)	0.333 (0.142)	0.295 (0.101)	0.388 (0.178)	0.406 (0.057)
	SVM+C	0.355 (0.115)	0.392 (0.065)	0.412 (0.125)	0.389 (0.074)	0.432 (0.164)	0.469 (0.059)
	LSTM	0.338 (0.124)	0.306 (0.114)	0.388 (0.128)	0.425 (0.073)	0.406 (0.074)	0.458 (0.052)
	LSTM+C	0.466 (0.121)	0.395 (0.099)	0.466 (0.115)	0.513 (0.068)	0.456 (0.068)	0.542 (0.037)
	biLSTM	0.513 (0.050)	0.450 (0.110)	0.464 (0.097)	0.424 (0.086)	0.417 (0.046)	0.450 (0.018)
	biLSTM+C	**0.578** (0.035)	**0.498** (0.080)	**0.558** (0.046)	**0.518** (0.065)	**0.470** (0.087)	**0.587** (0.020)
F_S							
BS	RF	0.393 (0.094)	0.391 (0.075)	0.449 (0.126)	0.459 (0.077)	0.381 (0.063)	0.317 (0.056)
	RF+C	0.341 (0.078)	0.351 (0.062)	0.380 (0.109)	0.383 (0.063)	0.334 (0.043)	0.276 (0.044)
	MLP	0.433 (0.042)	0.429 (0.053)	0.395 (0.063)	0.404 (0.105)	0.394 (0.134)	0.366 (0.072)
	MLP+C	0.376 (0.031)	0.381 (0.046)	0.340 (0.057)	0.357 (0.100)	0.341 (0.111)	0.329 (0.069)
	SVM	0.429 (0.071)	0.391 (0.043)	0.381 (0.032)	0.379 (0.075)	0.403 (0.067)	0.390 (0.100)
	SVM+C	0.390 (0.073)	0.354 (0.038)	0.336 (0.025)	0.336 (0.061)	0.363 (0.067)	0.346 (0.082)
	LSTM	0.373 (0.093)	0.359 (0.077)	0.381 (0.048)	0.347 (0.042)	0.377 (0.074)	0.276 (0.034)
	LSTM+C	0.341 (0.088)	0.315 (0.074)	0.336 (0.049)	0.314 (0.036)	0.319 (0.052)	0.247 (0.034)
	biLSTM	0.267 (0.054)	0.345 (0.038)	0.318 (0.069)	0.344 (0.040)	0.346 (0.034)	0.307 (0.028)
	biLSTM+C	**0.231** (0.042)	**0.294** (0.035)	**0.226** (0.060)	**0.291** (0.044)	**0.289** (0.021)	**0.220** (0.029)
BSS	RF	0.267 (0.182)	0.206 (0.160)	0.232 (0.224)	0.228 (0.066)	0.313 (0.131)	0.360 (0.093)
	RF+C	0.322 (0.150)	0.329 (0.133)	0.293 (0.216)	0.354 (0.078)	0.322 (0.093)	0.441 (0.072)
	MLP	0.282 (0.078)	0.138 (0.109)	0.226 (0.105)	0.284 (0.076)	0.259 (0.222)	0.345 (0.085)
	MLP+C	0.336 (0.059)	0.235 (0.093)	0.335 (0.088)	0.368 (0.069)	0.358 (0.183)	0.411 (0.086)
	SVM	0.122 (0.163)	0.228 (0.083)	0.251 (0.051)	0.247 (0.151)	0.204 (0.137)	0.310 (0.119)
	SVM+C	0.201 (0.165)	0.301 (0.075)	0.339 (0.043)	0.332 (0.122)	0.283 (0.135)	0.388 (0.091)
	LSTM	0.256 (0.204)	0.297 (0.150)	0.230 (0.092)	0.309 (0.080)	0.342 (0.152)	0.430 (0.129)
	LSTM+C	0.319 (0.193)	0.383 (0.144)	0.323 (0.096)	0.385 (0.067)	0.447 (0.145)	0.489 (0.122)
	biLSTM	0.464 (0.106)	0.309 (0.092)	0.451 (0.144)	0.314 (0.083)	0.351 (0.072)	0.437 (0.091)
	biLSTM+C	**0.535** (0.083)	**0.410** (0.084)	**0.513** (0.100)	**0.420** (0.087)	**0.457** (0.047)	**0.521** (0.089)

will produce an SEP event associated with the flare/CME. Furthermore, our biLSTM, when used as a probabilistic forecasting model, can provide a probabilistic estimate of how likely it is that the AR will produce an SEP event associated with the flare/CME. In the second task (F_S), given a data sample x_t at time point t in an AR where the AR will produce an M- or X-class flare within the next T hours of t, based on the SHARP parameters in x_t and its preceding $m-1$ data samples x_{t-m+1}, x_{t-m+2}, ..., x_{t-1}, our biLSTM, when used as a binary prediction model, can predict whether the AR will produce an SEP event associated with the flare, and when used as a probabilistic forecasting model, can provide a probabilistic estimate of how likely it is that the AR will produce an SEP event associated with the flare, regardless of

whether or not the flare initiates a CME. For both tasks, T ranges from 12 to 72 in 12 hr intervals.

We surveyed and collected data samples from the JSOC website, in the period between 2010 and 2021. Each data sample contains 18 SHARP parameters. Active regions (ARs) from 2010, 2016, and 2018-2021 were excluded from the study due to the lack of qualified data samples or the absence of SEP events associated with M-/X-class flares and CMEs. We then performed a cross-validation study on the remaining six years (2011-2015 and 2017). In the cross-validation study, training and test sets are disjoint, and hence our biLSTM model can make predictions on ARs that were never seen before. We evaluated the performance of our model and compared it with four related machine learning algorithms, namely RF [117], MLP [84], SVM [30] and a previous LSTM network [119]. The five machine learning methods including our biLSTM can be used both as binary prediction models and as probabilistic forecasting models. Our main results are summarized as follows.

1. The data samples in an AR are modeled as time series. We employ the biLSTM network to predict SEP events based on the time series. To our knowledge, this is the first study using a deep neural network to learn the dependencies in the temporal domain of the data for SEP prediction.

2. We evaluate the importance of the 18 SHARP parameters used in our study. It is found that using the top 15 SHARP parameters achieves the best performance for both the FC_S and F_S tasks. This finding is consistent with the literature which indicates using fewer high-quality SHARP parameters often achieves better performance for eruption prediction than using all the SHARP parameters including low-quality ones [10, 30, 122].

3. Our experiments show that the proposed biLSTM outperforms the four related machine learning methods in performing binary prediction and probabilistic forecasting for both the FC_S and F_S tasks. Furthermore, we introduce a calibration mechanism to enhance the accuracy of probabilistic forecasting. Overall, the calibrated biLSTM achieves the best performance among all the probabilistic forecasting models studied here.

4. When both an M-/X-class flare and its associated CME will occur, predicting whether there is an SEP event associated with the flare and CME is an easier problem (FC_S). Our biLSTM can solve the FC_S problem with relatively high accuracy. In contrast, when an M-/X-class flare will occur in the absence of CME information, predicting whether there is an SEP event associated with the flare is a harder problem (F_S). Our biLSTM solves the F_S problem with relatively low accuracy, and hence the prediction results would be less reliable.

5. The findings reported here are based on the cross-validation (CV) scheme in which six years (2011-2015 and 2017) are considered, data samples from each year in turn are used for testing, and data samples from the other five years together are used for training. To further understand the behavior of our biLSTM network and the four related machine learning methods, we have performed additional experiments using a random division (RD) scheme. With RD, we randomly select 10% of all positive data sequences and 10% of all negative data sequences, and use them together as the test set. The remaining 90% of the positive data sequences and 90% of the negative data sequences are used together as the training set. We repeat this experiment 100 times. The average values and standard deviations of the performance metrics are calculated. Tables 3.6 and 3.7 in section 3.6 present results of the five machine learning methods used as binary prediction models for the FC_S and F_S problems, respectively. Table 3.8 presents results of the five machine learning methods used as probabilistic forecasting models for the FC_S and F_S problems, respectively. It can be seen from these tables that the results obtained from the random division scheme are consistent with those from the cross-validation scheme, though the performance metric values from the RD scheme are generally better than those from the CV scheme. This happens probably because with the RD scheme the machine learning methods are trained by more diverse data and hence are more knowledgeable, yielding more accurate results than with the CV scheme.

It should be pointed out that, in solving the FC_S problem, the condition in which we have a data sample x_t at time point t in an AR where the AR will produce an M- or X-class flare within the next T hours of t and the flare initiates a CME is given. That is, we assume an M- or X-class flare and its associated CME will occur. In an operational system, one can determine in two phases if an AR will produce an M- or X-class flare within the next T-hours of a given time point t and if the flare initiates a CME, as follows [118]. In the first phase, one can use a flare prediction tool such as [117, 61, 89, 132, 119] to predict whether there will be an M- or X-class

flare within the next T hours of t. If the answer is yes, then in the second phase one can use a CME prediction tool such as [122] to predict whether the flare initiates a CME. If the answer is also yes, then one can use the proposed biLSTM to predict whether there is an SEP event associated with the flare and CME. On the other hand, to solve the F_S problem, one only needs to execute the first phase. If the answer from the first phase indicates that an M- or X-class flare will occur within the next T hours of t, one can then go ahead to use the proposed biLSTM to predict whether there is an SEP event associated with the flare. Thus, the proposed biLSTM does not function in a stand-alone manner. Rather, it first requires the other tools to provide flare/CME predictions. As such, the performance of the operational biLSTM system depends on the performance of the other tools. A wrong prediction from the other tools would affect the accuracy of our approach.

3.6 Additional Experiments and Results

Tables 3.6 and 3.7 present results of the five machine learning methods (RF, MLP, SVM, LSTM, biLSTM) used as binary prediction models for the FC_S and F_S problems, respectively. Table 3.8 presents results of the five machine learning methods used as probabilistic forecasting models for the FC_S and F_S problems, respectively. The tables show the mean performance metric values averaged over the 100 experiments based on the random division scheme with standard deviations enclosed in parentheses. Best average metric values are highlighted in boldface.

Table 3.6 Performance Comparison of RF, MLP, SVM, LSTM and biLSTM Based on the Random Division Scheme for the FC_S Problem

		12 hr	24 hr	36 hr	48 hr	60 hr	72 hr
Recall	RF	0.699 (0.149)	0.686 (0.148)	0.657 (0.169)	0.701 (0.165)	0.689 (0.087)	0.817 (0.080)
	MLP	0.655 (0.164)	0.690 (0.132)	0.688 (0.149)	0.692 (0.149)	0.679 (0.101)	0.798 (0.091)
	SVM	0.666 (0.170)	0.693 (0.147)	0.681 (0.159)	0.701 (0.153)	0.716 (0.088)	0.810 (0.091)
	LSTM	0.786 (0.080)	0.804 (0.071)	0.840 (0.059)	0.860 (0.047)	0.870 (0.054)	0.884 (0.061)
	biLSTM	**0.911** (0.056)	**0.844** (0.067)	**0.860** (0.057)	**0.882** (0.047)	**0.875** (0.052)	**0.906** (0.054)
Precision	RF	0.680 (0.184)	0.670 (0.113)	0.654 (0.069)	0.636 (0.088)	0.675 (0.036)	0.719 (0.046)
	MLP	0.650 (0.214)	0.630 (0.079)	0.659 (0.069)	0.627 (0.073)	0.659 (0.040)	0.715 (0.045)
	SVM	0.733 (0.163)	0.649 (0.090)	0.707 (0.083)	0.684 (0.093)	0.692 (0.042)	0.725 (0.046)
	LSTM	0.682 (0.135)	0.692 (0.130)	0.706 (0.103)	0.683 (0.120)	0.710 (0.106)	0.727 (0.071)
	biLSTM	**0.788** (0.149)	**0.721** (0.132)	**0.722** (0.103)	**0.705** (0.118)	**0.752** (0.107)	**0.749** (0.067)
BACC	RF	0.781 (0.073)	0.776 (0.068)	0.761 (0.078)	0.768 (0.083)	0.770 (0.042)	0.831 (0.042)
	MLP	0.750 (0.085)	0.768 (0.057)	0.774 (0.069)	0.759 (0.067)	0.761 (0.049)	0.822 (0.046)
	SVM	0.787 (0.093)	0.775 (0.065)	0.787 (0.080)	0.784 (0.080)	0.787 (0.046)	0.831 (0.047)
	LSTM	0.822 (0.051)	0.828 (0.051)	0.847 (0.044)	0.840 (0.054)	0.848 (0.042)	0.860 (0.046)
	biLSTM	**0.906** (0.041)	**0.854** (0.048)	**0.861** (0.042)	**0.858** (0.051)	**0.867** (0.045)	**0.878** (0.040)
HSS	RF	0.548 (0.152)	0.541 (0.124)	0.516 (0.130)	0.516 (0.150)	0.536 (0.072)	0.639 (0.076)
	MLP	0.492 (0.183)	0.517 (0.096)	0.537 (0.118)	0.500 (0.119)	0.516 (0.083)	0.624 (0.081)
	SVM	0.585 (0.178)	0.534 (0.110)	0.575 (0.141)	0.561 (0.148)	0.568 (0.080)	0.640 (0.082)
	LSTM	0.612 (0.128)	0.624 (0.125)	0.656 (0.105)	0.633 (0.131)	0.656 (0.108)	0.682 (0.093)
	biLSTM	**0.769** (0.124)	**0.670** (0.123)	**0.682** (0.103)	**0.667** (0.126)	**0.702** (0.112)	**0.717** (0.083)
TSS	RF	0.562 (0.146)	0.551 (0.136)	0.523 (0.155)	0.536 (0.165)	0.541 (0.084)	0.662 (0.084)
	MLP	0.501 (0.171)	0.536 (0.114)	0.549 (0.139)	0.519 (0.134)	0.522 (0.097)	0.644 (0.093)
	SVM	0.574 (0.186)	0.551 (0.131)	0.573 (0.161)	0.568 (0.161)	0.574 (0.091)	0.661 (0.093)
	LSTM	0.645 (0.103)	0.657 (0.102)	0.695 (0.088)	0.680 (0.109)	0.697 (0.085)	0.720 (0.091)
	biLSTM	**0.812** (0.081)	**0.708** (0.096)	**0.722** (0.073)	**0.715** (0.103)	**0.733** (0.091)	**0.756** (0.079)
WAUC	RF	0.619 (0.022)	0.601 (0.046)	0.577 (0.022)	0.579 (0.050)	0.599 (0.065)	0.729 (0.072)
	MLP	0.551 (0.015)	0.581 (0.013)	0.605 (0.042)	0.573 (0.051)	0.565 (0.024)	0.709 (0.035)
	SVM	0.633 (0.015)	0.597 (0.079)	0.630 (0.048)	0.618 (0.057)	0.625 (0.056)	0.730 (0.046)
	LSTM	0.708 (0.039)	0.725 (0.020)	0.757 (0.084)	0.744 (0.081)	0.761 (0.029)	0.782 (0.070)
	biLSTM	**0.895** (0.013)	**0.775** (0.041)	**0.799** (0.051)	**0.780** (0.069)	**0.796** (0.058)	**0.821** (0.063)

Table 3.7 Performance Comparison of RF, MLP, SVM, LSTM and biLSTM Based on the Random Division Scheme for the F_S Problem

		12 hr	24 hr	36 hr	48 hr	60 hr	72 hr
Recall	RF	0.734 (0.105)	0.797 (0.117)	0.749 (0.147)	0.710 (0.116)	0.717 (0.089)	0.756 (0.104)
	MLP	0.683 (0.087)	0.769 (0.123)	0.752 (0.109)	0.689 (0.121)	0.702 (0.103)	0.728 (0.083)
	SVM	0.767 (0.107)	0.791 (0.101)	0.786 (0.104)	0.730 (0.120)	0.740 (0.103)	0.743 (0.086)
	LSTM	0.774 (0.098)	0.770 (0.125)	0.817 (0.108)	0.782 (0.120)	0.768 (0.106)	0.772 (0.103)
	biLSTM	**0.834** (0.096)	**0.837** (0.120)	**0.856** (0.107)	**0.837** (0.122)	**0.818** (0.101)	**0.815** (0.102)
Precision	RF	0.243 (0.117)	0.244 (0.103)	0.261 (0.096)	0.308 (0.089)	0.396 (0.190)	0.434 (0.162)
	MLP	0.225 (0.107)	0.243 (0.100)	0.269 (0.082)	0.291 (0.069)	0.363 (0.156)	0.366 (0.097)
	SVM	0.235 (0.104)	0.231 (0.086)	0.242 (0.067)	0.326 (0.099)	0.383 (0.166)	0.393 (0.113)
	LSTM	0.250 (0.114)	0.252 (0.097)	0.291 (0.092)	0.349 (0.108)	0.413 (0.189)	0.443 (0.165)
	biLSTM	**0.279** (0.131)	**0.275** (0.107)	**0.306** (0.099)	**0.377** (0.119)	**0.483** (0.174)	**0.476** (0.173)
BACC	RF	0.770 (0.073)	0.774 (0.058)	0.758 (0.087)	0.760 (0.054)	0.760 (0.083)	0.795 (0.049)
	MLP	0.742 (0.065)	0.764 (0.058)	0.766 (0.054)	0.746 (0.056)	0.748 (0.069)	0.770 (0.045)
	SVM	0.783 (0.069)	0.770 (0.057)	0.764 (0.071)	0.772 (0.056)	0.770 (0.069)	0.783 (0.046)
	LSTM	0.791 (0.068)	0.772 (0.062)	0.799 (0.053)	0.801 (0.056)	0.787 (0.069)	0.804 (0.049)
	biLSTM	**0.825** (0.068)	**0.809** (0.059)	**0.821** (0.053)	**0.832** (0.057)	**0.841** (0.059)	**0.830** (0.049)
HSS	RF	0.284 (0.151)	0.274 (0.129)	0.284 (0.128)	0.327 (0.102)	0.390 (0.196)	0.442 (0.153)
	MLP	0.255 (0.138)	0.269 (0.125)	0.295 (0.098)	0.306 (0.089)	0.358 (0.166)	0.379 (0.110)
	SVM	0.280 (0.138)	0.260 (0.111)	0.267 (0.101)	0.350 (0.110)	0.387 (0.174)	0.409 (0.122)
	LSTM	0.298 (0.149)	0.284 (0.122)	0.330 (0.110)	0.385 (0.115)	0.418 (0.189)	0.455 (0.155)
	biLSTM	**0.340** (0.165)	**0.321** (0.131)	**0.354** (0.116)	**0.426** (0.124)	**0.515** (0.158)	**0.500** (0.162)
TSS	RF	0.540 (0.145)	0.548 (0.116)	0.516 (0.174)	0.519 (0.109)	0.521 (0.166)	0.589 (0.098)
	MLP	0.484 (0.130)	0.527 (0.116)	0.533 (0.108)	0.493 (0.111)	0.497 (0.138)	0.540 (0.090)
	SVM	0.566 (0.138)	0.540 (0.113)	0.528 (0.142)	0.545 (0.111)	0.539 (0.138)	0.567 (0.093)
	LSTM	0.581 (0.137)	0.544 (0.125)	0.598 (0.106)	0.601 (0.112)	0.574 (0.138)	0.607 (0.097)
	biLSTM	**0.651** (0.136)	**0.617** (0.119)	**0.641** (0.106)	**0.664** (0.114)	**0.682** (0.118)	**0.660** (0.097)
WAUC	RF	0.591 (0.057)	0.600 (0.021)	0.561 (0.016)	0.565 (0.038)	0.577 (0.043)	0.655 (0.057)
	MLP	0.528 (0.068)	0.577 (0.043)	0.578 (0.069)	0.543 (0.065)	0.547 (0.054)	0.597 (0.085)
	SVM	0.624 (0.025)	0.586 (0.027)	0.580 (0.032)	0.602 (0.039)	0.590 (0.069)	0.617 (0.065)
	LSTM	0.634 (0.052)	0.595 (0.048)	0.665 (0.026)	0.666 (0.027)	0.625 (0.028)	0.673 (0.062)
	biLSTM	**0.712** (0.068)	**0.680** (0.082)	**0.710** (0.057)	**0.730** (0.053)	**0.753** (0.031)	**0.732** (0.060)

Table 3.8 Probabilistic Forecasting Results of RF, MLP, SVM, LSTM and biLSTM With and Without Calibration Based on the Random Division Scheme for the FC_S and F_S Problems, Respectively

			12 hr	24 hr	36 hr	48 hr	60 hr	72 hr
FC_S	BS	RF	0.268 (0.068)	0.279 (0.076)	0.305 (0.102)	0.300 (0.119)	0.232 (0.043)	0.305 (0.040)
		RF+C	0.226 (0.057)	0.246 (0.066)	0.260 (0.087)	0.256 (0.102)	0.272 (0.038)	0.269 (0.036)
		MLP	0.311 (0.101)	0.303 (0.084)	0.339 (0.118)	0.312 (0.119)	0.303 (0.046)	0.311 (0.066)
		MLP+C	0.265 (0.087)	0.259 (0.071)	0.288 (0.101)	0.266 (0.102)	0.283 (0.040)	0.293 (0.058)
		SVM	0.261 (0.073)	0.264 (0.089)	0.271 (0.121)	0.261 (0.127)	0.285 (0.065)	0.282 (0.043)
		SVM+C	0.219 (0.061)	0.241 (0.076)	0.252 (0.103)	0.253 (0.109)	0.249 (0.058)	0.231 (0.040)
		LSTM	0.258 (0.041)	0.263 (0.041)	0.278 (0.035)	0.272 (0.044)	0.257 (0.034)	0.268 (0.036)
		LSTM+C	0.220 (0.037)	0.224 (0.037)	0.236 (0.031)	0.232 (0.040)	0.238 (0.031)	0.235 (0.033)
		biLSTM	0.214 (0.021)	0.230 (0.034)	0.203 (0.037)	0.214 (0.040)	0.193 (0.024)	0.185 (0.012)
		biLSTM+C	**0.183** (0.019)	**0.195** (0.027)	**0.153** (0.028)	**0.182** (0.035)	**0.164** (0.020)	**0.157** (0.011)
	BSS	RF	0.464 (0.138)	0.429 (0.154)	0.402 (0.207)	0.408 (0.235)	0.354 (0.099)	0.368 (0.092)
		RF+C	0.549 (0.118)	0.516 (0.132)	0.490 (0.176)	0.495 (0.201)	0.453 (0.088)	0.460 (0.084)
		MLP	0.381 (0.205)	0.399 (0.169)	0.320 (0.243)	0.378 (0.241)	0.338 (0.096)	0.221 (0.139)
		MLP+C	0.474 (0.175)	0.488 (0.143)	0.422 (0.209)	0.469 (0.207)	0.439 (0.084)	0.335 (0.122)
		SVM	0.476 (0.154)	0.431 (0.176)	0.433 (0.248)	0.386 (0.258)	0.415 (0.133)	0.409 (0.096)
		SVM+C	0.560 (0.129)	0.517 (0.151)	0.509 (0.210)	0.488 (0.222)	0.505 (0.120)	0.517 (0.087)
		LSTM	0.492 (0.085)	0.478 (0.082)	0.445 (0.074)	0.455 (0.092)	0.452 (0.077)	0.430 (0.079)
		LSTM+C	0.566 (0.075)	0.556 (0.072)	0.529 (0.067)	0.535 (0.086)	0.533 (0.067)	0.516 (0.070)
		biLSTM	0.573 (0.053)	0.545 (0.070)	0.596 (0.078)	0.569 (0.087)	0.610 (0.056)	0.627 (0.036)
		biLSTM+C	**0.635** (0.047)	**0.614** (0.060)	**0.696** (0.058)	**0.633** (0.077)	**0.668** (0.047)	**0.683** (0.032)
F_S	BS	RF	0.284 (0.076)	0.288 (0.061)	0.372 (0.067)	0.320 (0.130)	0.307 (0.094)	0.310 (0.052)
		RF+C	0.272 (0.099)	0.276 (0.087)	0.312 (0.056)	0.268 (0.109)	0.274 (0.121)	0.281 (0.101)
		MLP	0.332 (0.114)	0.329 (0.072)	0.394 (0.080)	0.352 (0.080)	0.316 (0.145)	0.332 (0.147)
		MLP+C	0.283 (0.098)	0.278 (0.063)	0.336 (0.071)	0.279 (0.123)	0.299 (0.124)	0.305 (0.124)
		SVM	0.276 (0.067)	0.279 (0.063)	0.298 (0.059)	0.276 (0.054)	0.295 (0.058)	0.290 (0.045)
		SVM+C	0.236 (0.059)	0.249 (0.055)	0.264 (0.050)	0.267 (0.045)	0.291 (0.049)	0.248 (0.038)
		LSTM	0.277 (0.065)	0.279 (0.059)	0.285 (0.051)	0.282 (0.080)	0.273 (0.066)	0.289 (0.046)
		LSTM+C	0.236 (0.056)	0.231 (0.051)	0.242 (0.044)	0.246 (0.068)	0.247 (0.058)	0.247 (0.040)
		biLSTM	0.260 (0.054)	0.247 (0.048)	0.260 (0.043)	0.265 (0.046)	0.253 (0.085)	0.237 (0.089)
		biLSTM+C	**0.221** (0.046)	**0.209** (0.041)	**0.221** (0.037)	**0.223** (0.039)	**0.214** (0.073)	**0.201** (0.076)
	BSS	RF	0.390 (0.133)	0.365 (0.077)	0.397 (0.071)	0.342 (0.139)	0.330 (0.164)	0.324 (0.162)
		RF+C	0.463 (0.158)	0.470 (0.140)	0.482 (0.089)	0.424 (0.172)	0.440 (0.194)	0.419 (0.167)
		MLP	0.289 (0.205)	0.348 (0.146)	0.215 (0.170)	0.260 (0.191)	0.264 (0.208)	0.200 (0.153)
		MLP+C	0.388 (0.187)	0.450 (0.127)	0.331 (0.150)	0.366 (0.186)	0.357 (0.204)	0.304 (0.165)
		SVM	0.447 (0.145)	0.420 (0.134)	0.403 (0.129)	0.378 (0.109)	0.404 (0.117)	0.391 (0.097)
		SVM+C	0.529 (0.126)	0.493 (0.117)	0.492 (0.109)	0.461 (0.091)	0.420 (0.101)	0.507 (0.083)
		LSTM	0.450 (0.139)	0.467 (0.128)	0.436 (0.103)	0.422 (0.145)	0.442 (0.139)	0.421 (0.101)
		LSTM+C	0.531 (0.118)	0.536 (0.110)	0.511 (0.088)	0.500 (0.146)	0.461 (0.121)	0.506 (0.086)
		biLSTM	0.484 (0.112)	0.505 (0.105)	0.489 (0.089)	0.466 (0.097)	0.428 (0.154)	0.432 (0.166)
		biLSTM+C	**0.592** (0.095)	**0.581** (0.090)	**0.566** (0.075)	**0.550** (0.082)	**0.504** (0.165)	**0.613** (0.074)

CHAPTER 4

PREDICTION OF GEOMAGNETIC INDICES

4.1 Prediction of SYM-H Index

4.1.1 Background and Related Work

Geomagnetic activities and events are known to have a substantial impact on Earth. They can damage and affect technological systems such as telecommunication networks, power transmission systems, and spacecrafts [13]. These activities are massive and scale in orders of magnitude [131]. It may take a few days to recover from the damage depending on its severity. These activities and events can not be ignored regardless if they are in regions at high, medium, or low latitudes [40, 65, 129, 162, 1]. There are several solar activity indices that measure the intensity of the geomagnetic effects. These indices characterize the magnitude of the disturbance over time. Modeling and forecasting these indices have become a crucial area of study in space weather research.

Some indices, such as Kp, describe the overall level of geomagnetic activity while others, such as the disturbance storm time (Dst) index [70, 178, 186], describe a specific area of geomagnetic activity. The Dst index aims to classify a storm based on its intensity. It is a superstorm when Dst is smaller than -250 nT, moderate when Dst is larger than -50 nT, and powerful when Dst is between -50 nT and -250 nT [71, 124]. Another important index is the symmetric H-component (SYM-H) index, which is used to represent the longitudinally symmetric disturbance of the ring current intensity during geomagnetic storms [140, 171, 175]. The SYM-H index is basically the same as the Dst index but with a high temporal resolution of 1 minute instead of 1 or 3 hours, which is very useful in studying a short period of temporal variations of geomagnetic activities [154]. On the other hand, ASY-H (the

asymmetric geomagnetic disturbance of the horizontal component) is quantified as the longitudinally asymmetric part of the geomagnetic disturbance field at low latitude to midlatitude. In addition, there are other indices that can be used to measure the storm activity as described in [126].

A lot of efforts have been devoted to developing strategies to alleviate the geomagnetic effects on technologies and humans, but it is almost impossible to offer complete protection from the effects [154]. Some of these strategies are to accurately forecast and predict the occurrence and intensity of geomagnetic storms to offer some level of mitigation of their damaging effects. [35] established an empirical connection between interplanetary circumstances and Dst using a linear forecasting model. [173] used differential equation models to examine the effect of the solar wind dynamic pressure on the decay and injection of the ring current. [13] performed predictions of global magnetic disturbance in near-Earth space in a case study for Kp index using Nonlinear AutoRegressive with eXogenous (NARX) models. Due to the intrinsic complex response of the circumterrestrial environment to changes in the interplanetary medium, these simple models were unable to properly and fully depict the evolution of the solar wind-magnetosphere-ionosphere system [49, 98, 154]. To surpass the limitations of the simple models and to acquire the complex response of the magnetosphere, researchers resorted to more advanced models such as artificial neural networks (ANNs).

The use of ANNs focused on the prediction of Dst and Kp indices. [66] constructed the first Dst prediction model by employing a time-delay ANN with solar wind parameters as input variables. The authors performed Dst index projections for 1 to 6 hours in advance. [105] created a particle swarm optimization method to train ANN connection weights to improve the accuracy of Dst index prediction. [15] combined ANNs and physical models with solar wind and interplanetary magnetic field parameters such as velocity, interplanetary magnetic field (IMF) magnitude, and

clock angle. [42] used Gaussian processes (GP) to build an auto-regressive model to predict the Dst index 1 hour in advance based on the past solar wind velocity, IMF B_z component, and Dst index values. This method generated a predictive distribution rather than a single prediction point. However, the mean values of the estimations are not as accurate as those generated by ANNs. [71] overcame the poor performance of GP and constructed a Dst index estimation model by merging GP with a long short-term memory (LSTM) network to obtain more accurate results. [141] compared the effectiveness of 30 models that forecast the Dst index and found that none of the models performed consistently best for all events.

Relatively few researchers have focused on the SYM-H index prediction. This happens probably because of the high temporal resolution of 1 minute for the SYM-H index, which gives rise to a more difficult problem to estimate SYM-H due to its highly oscillating nature [154]. Nevertheless, there have been some SYM-H index prediction techniques reported in literature. [38] presented the first 5-minute average estimates of the SYM-H index throughout large storms between 1998 and 2006 by using an NARX neural network with IMF and solar wind data. [24] predicted both SYM-H and ASY-H indices for solar cycle 24 by employing the NARX neural network in a similar manner. Both [24] and [38] made use of IMF magnitude (B), B_y and B_z components, as well as solar wind density and velocity as input data to their models. [154] provided a comprehensive examination of two well-known deep learning models, namely long short-term memory (LSTM) and a convolutional neural network (CNN), with an average temporal resolution of 5 minutes for the estimation of SYM-H index values (1 hour in advance). The authors made use of IMF B_z component, squared values of IMF magnitude B and of B_y component, measured at L1 by the ACE satellite in GSM coordinates. [48] created neural network models for SYM-H and ASY-H predictions by combining CNN and LSTM. The authors considered 42 geomagnetic storms between 1998 and 2018 for model training, validation, and testing purposes.

[85] developed a model using gradient boosting machines to predict the SYM-H index (1 and 2 hours in advance) with a temporal resolution of 5 minutes.

In this study, we present a novel method, named SYMHNet, that utilizes cooperative learning of a graph neural network (GNN) and a bidirectional long short-term memory (BiLSTM) network with Bayesian inference to conduct short-term (i.e., 1-6 hour ahead) predictions of the SYM-H index for solar cycles 23 and 24. We consider temporal resolutions of 1 minute and 5 minutes for the SYM-H index. Our method can quantify both model and data uncertainties when producing forecasting results.

The remainder of this study is organized as follows. Subsection 4.1.2 describes the data, including the solar wind and IMF parameters as well as geomagnetic storms, used in this study. Subsection 4.1.3 presents the architecture of SYMHNet and its uncertainty quantification algorithm. Subsection 4.1.5 compares SYMHNet with related machine learning methods and reports experimental results. Subsection 4.2.7 presents a discussion and concludes the paper.

4.1.2 Data

In training and evaluating SYMHNet, we build a database that combines the solar wind and IMF parameters with geomagnetic storms studied here. This database contains 42 storms selected from the past 2 solar cycles (#23 and #24) between 1996 and 2018. We describe the parameters and storms in detail below.

Solar Wind and IMF Parameters We consider seven solar wind, IMF, and derived parameters: IMF magnitude (B), B_y and B_z components, flow speed, proton density, electric field, and flow pressure. These parameters have been used in related studies [24, 38, 57, 85]. The parameters' values along with SYM-H index values are

collected from the NASA Space Science Data Coordinated Archive[1] [96]. The data is collected with both 1-minute and 5-minute resolutions.

Geomagnetic Storms We work on the same storms as those considered in the previous studies [48, 85, 154]. Table 4.1 lists the storms used for training SYMHNet. Table 4.2 lists the storms used for validating SYMHNet. Table 4.3 lists the storms used for testing SYMHNet. The training set, validation set and test set are disjoint. Therefore, SYMHNet can make predictions on storms that it has never seen during training.

Table 4.1 Storms Used to Train SYMHNet

Storm #	Start date	End date	Min SYM-H (nT)
1	02/14/1998	02/22/1998	−119
2	08/02/1998	08/08/1998	−168
3	09/19/1998	09/29/1998	−213
4	02/16/1999	02/24/1999	−127
5	10/15/1999	10/25/1999	−218
6	07/09/2000	07/19/2000	−335
7	08/06/2000	08/16/2000	−235
8	09/15/2000	09/25/2000	−196
9	11/01/2000	11/15/2000	−174
10	03/14/2001	03/24/2001	−165
11	04/06/2001	04/16/2001	−275
12	10/17/2001	10/22/2001	−210
13	10/31/2001	11/10/2001	−313
14	05/17/2002	05/27/2002	−113
15	11/15/2003	11/25/2003	−488
16	07/20/2004	07/30/2004	−208
17	05/10/2005	05/20/2005	−302
18	04/09/2006	04/19/2006	−110
19	10/09/2006	12/19/2006	−206
20	03/01/2012	03/11/2012	−149

[1]Retrieved on 09/10/2022 from `https://nssdc.gsfc.nasa.gov`

Table 4.2 Storms Used to Validate SYMHNet

Storm #	Start date	End date	Min SYM-H (nT)
21	04/28/1998	05/08/1998	−268
22	09/19/1999	09/26/1999	−160
23	10/25/2003	11/03/2003	−427
24	06/18/2015	06/28/2015	−207
25	09/01/2017	09/11/2017	−144

Table 4.3 Storms Used to Test SYMHNet

Storm #	Start date	End date	Min SYM-H (nT)
26	06/22/1998	06/30/1998	−120
27	11/02/1998	11/12/1998	−179
28	01/09/1999	01/18/1999	−111
29	04/13/1999	04/19/1999	−122
30	01/16/2000	01/26/2000	−101
31	04/02/2000	04/12/2000	−315
32	05/19/2000	05/28/2000	−159
33	03/26/2001	04/04/2001	−434
34	05/26/2003	06/06/2003	−162
35	07/08/2003	07/18/2003	−125
36	01/18/2004	01/27/2004	−137
37	11/04/2004	11/14/2004	−393
38	09/10/2012	10/05/2012	−138
39	05/28/2013	06/04/2013	−134
40	06/26/2013	07/04/2013	−110
41	03/11/2015	03/21/2015	−233
42	08/22/2018	09/03/2018	−205

4.1.3 Methodology

Machine learning (ML) and its subfield, deep learning (DL) [67], have been used extensively in the space weather community for predicting solar flares [4, 82, 119], flare precursors [47], coronal mass ejections [9, 122], solar energetic particles [2], and geomagnetic indices [15, 24, 48, 70, 105, 154]. Different from the existing methods, SYMHNet combines a graph neural network (GNN) and a bidirectional long short-term memory (biLSTM) network to jointly learn patterns from input data. As our experimental results show later, this combined learning framework works well and

performs better than the existing methods for SYM-H index forecasting. We describe our method in detail below.

Parameter Graph We construct an undirected unweighted fully connected graph (FCG) for the solar wind, IMF, and derived parameters considered in this study where each node represents a parameter and there is an edge between every two nodes. Because the parameter values' are time series, we obtain a time series of parameter graphs where the topologies of the graphs are the same but the node values vary as time goes. For example, Figure 4.1 shows three parameter graphs constructed at time points t, $t + 1$, $t + 2$ with a resolution of 1 minute for predicting the SYM-H index 1 hour in advance. In Figure 4.1, the leftmost graph at t contains the seven parameters' values, represented by seven nodes or circles, at time point t. The FCG symbol in the center indicates that this is a fully connected graph in which every two nodes are connected by an edge. (For simplicity, only a portion of the edges are shown in the figure.) In addition, the graph contains a node representing the SYM-H index value at time point $t + 1$ hour. The SYM-H index value is used as the label for this graph. The GNN in SYMHNet will learn the relationships among the parameters' values and the relationships between the parameters' values and the label. If we want to predict the SYM-H index k hours in advance, $1 < k \leq 6$, then the label will be the SYM-H index value at time point $t + k$ hours.

The middle graph at $t + 1$ in Figure 4.1 contains the seven parameters' values at time point $t + 1$ minutes. In addition, this graph contains the SYM-H index value at time point ($t + 1$ minutes) + 1 hour, which is the label for this graph. If we want to predict the SYM-H index k hours in advance, $1 < k \leq 6$, then the label will be the SYM-H index value at time point ($t + 1$ minutes) + k hours.

The rightmost graph at $t + 2$ in Figure 4.1 contains the seven parameters' values at time point $t + 2$ minutes. In addition, this graph contains the SYM-H index value

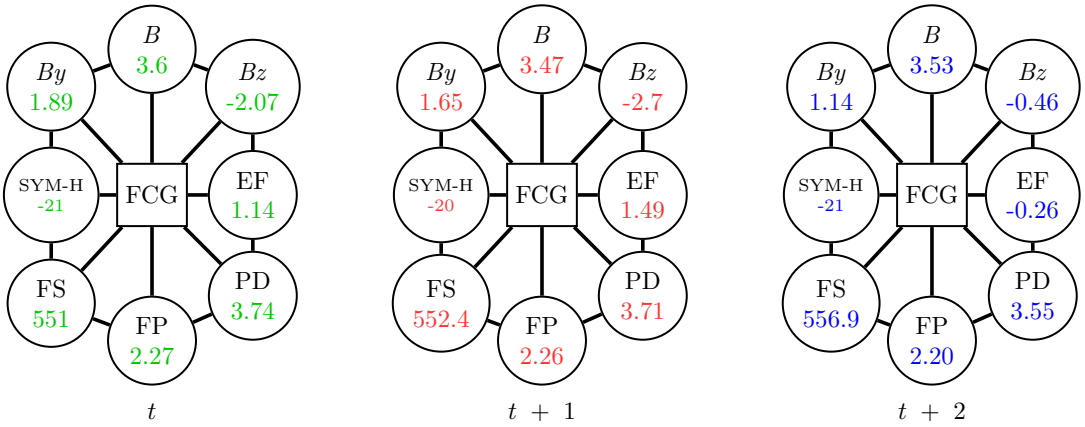


Figure 4.1 Illustration of the parameter graphs constructed at time points t, $t+1$, $t+2$ with a resolution of 1 minute for predicting the SYM-H index 1 hour in advance. Each graph contains seven parameters: IMF magnitude (B), B_y component, B_z component, electric field (EF), proton density (PD), flow pressure (FP), and flow speed (FS). The colored values in the graphs represent the parameters' values that change as time goes while the topologies of the graphs remain the same. The value in the SYM-H node in a graph is the label of the graph. The FCG symbol in a graph indicates that the graph is fully connected.

at time point (t + 2 minutes) + 1 hour, which is the label for this graph. If we want to predict the SYM-H index k hours in advance, $1 < k \leq 6$, then the label will be the SYM-H index value at time point (t + 2 minutes) + k hours.

The SYMHNet Architecture In this section, we describe how the SYMHNet network works as shown in Figure 4.2. The trained SYMHNet model takes a constructed parameter graph, as described in Section 4.1.3, and the time series of the time-varying parameters as input. The graph is used as an input to the graph neural network (GNN) component to learn and determine the topological correlations between the parameters. Specifically, at time t we create the virtual connection between each parameter. Figure 4.1 shows only one connection between each two nodes and omits all other edges to keep the figure clean for demonstration only. Next, for every node in the graph, we create time series sequence at times $t, t+1, ...t+n$ to represent the features' information at each time step as shown in Figure 4.1 and denoted with the numbers in each node in the graph. The time series data is used

as an input to the bidirectional long short-term memory (BiLSTM) component to learn the temporal correlation between the parameters data. Next, the SYMHNet network combines and correlates the graphical information and temporal patterns it finds to build the predictive network. SYMHNet network concatenates and joins the patterns learned from both GNN and BiLSTM to build a joint pattern and passes to the dropout layers for further processing. The network then trains the Bayesian deep-learning model for probabilistic prediction and uncertainties. We use the mean squared error (MSE) as the loss function of our network and the root mean squared propagation (RMSProp) with a learning rate that is set to 0.0002. RMSprop is an extension of gradient descent that uses a decaying average of partial gradients in the adaptation of the step size for each parameter to allow the algorithm to ignore early gradients and focus on the recently observed partial gradients during the progress, which improves the pattern discoveries from the parameters. In the following section, we describe each component of the SYMHNet architecture in detail.

Graph Neural Network Neural networks (NNs) have shown significant predictive power in learning and capturing hidden patterns from fixed-size, regular-structured, Euclidean data such as texts, time series, images, and videos. With the increasing size and different type of data that comes from different systems such as social media, citations, transactions, physical models, and/or non-Euclidean data that can be presented as graphs, NNs have a limitation and not able to accurately learning and predict them. To overcome the shortcoming of NNs, Graph Neural Networks (GNNs) were developed to learn patterns from these type of data.

GNNs are a class of neural networks that can learn from graph-structured data. GNNs extract and utilize the features from the graphs to perform more accurate prediction about the entities interactions such as node, edge and/or graph level.

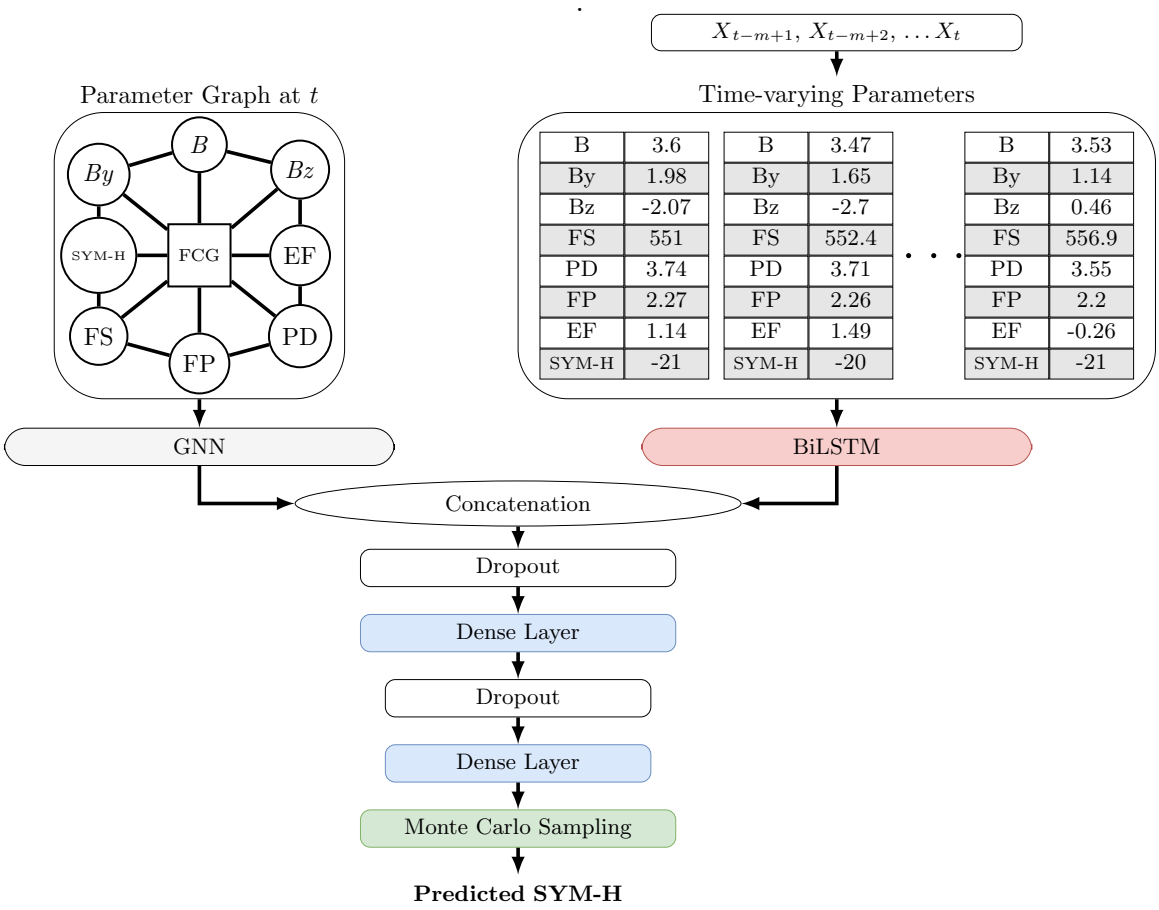


Figure 4.2 Architecture of SYMHNet. (a) is the overall architecture of SYMHNet. (b) is the GNN architecture. (c) is the BiLSTM architecture.

The general function of GNNs is identifying the labels graph's nodes. For instance, given an observed graph G that is constructed of non-empty set of N

vertices, also referred to as nodes, V and set of edges E so that $G = (V, E)$. For every node or vertex in the graph G, the derived features denoted by x_i. For a given subset of vertices $S \subset V$, the labels are measured by $Y_S = \{y_i : i \in S\}$. In categorical classification, the label y_i can one of categories such as true or false, active or inactive, and so on. In regression task, the label can be a real value. The goal of GNNs is estimate and predict the unlabeled nodes or vertices using the features x and the given observed graph G [190]. GNNs network perform graph convolution functions within the neural network architecture. The collection of feature vectors as the rows vectors X are calculated as follows [54, 97, 190]:

$$H^{(1)} = \sigma(A_g X W^{(0)}) \tag{4.1}$$

$$H^{(l+1)} = \sigma(A_g H^{(l)} W^{(l)}) \tag{4.2}$$

where $H^{(l)}$ represents the output features from layer $l-1$, A_g is a matrix derived from the observed graph G and decides, at each layer, how the output are mixed across the graph. X represent the feature matrix, $W^{(l)}$ represents the weights of the neural network at layer l. The final output (the predictions) of an L-layer network is denoted by $Z = H^{(L)}$. The network uses back-propagation to minimize the error metrics between the predicted labels Z and the ground truth or observed labels Y. In addition, the network performance can be improved by using techniques such as attention nodes [170] and skip connection and gates [95].

In our research, we are building a static graph structure with temporal time series signals that receives information at each time step t as illustrated it in Figure 4.1 to build a Temporal Graph Neural Networks (TGNNs) or Spatio-Temporal Graph Neural Networks (STGNNs) model. In this model, the graph structure does not change overtime this is because, in our research, the graph is built using the selected

solar wind parameters that are always present within the span of the given data. Each node (parameter) in the graph receives an information over time that is known as time-varying features values. The TGNNs represents and captures the dependency of the symh or asyh within the generated graph over time. All information is considered as a signal that is updated at every time point t.

Bidirectional LSTM (BiLSTM) The proposed algorithm and network include a bidirectional long short-term memory (LSTM) component that we refer to it by BiLSTM. BiLSTM is comprised of two LSTM [75] layers with opposite direction when they ready the data. It allows the network to use one LSTM layer to read the sequence from the start to the end, denoted as forward, and the other LSTM layer to read the sequence from the end to the start, denoted as backward. This behaviour allows the network to learn more patterns and features from the data. Figure 4.2(c) illustrates the BiLSTM component architecture in the proposed SYMHNet model. Both LSTM and BiLSTM have shown significant performance improvement in processing time series, but furthermore, BiLSTM outperforms LSTM as proven by many studies [2, 153].

4.1.4 Uncertainty Quantification

Uncertainty quantification is essential for a model's reproducibility and validation [172]. Uncertainty quantification with deep learning has been used in computer vision [93], space weather [71], and solar physics [87]. There are two types of uncertainty: aleatoric and epistemic. Aleatoric uncertainty captures the inherent randomness of data, hence also referred to as data uncertainty. Epistemic uncertainty occurs due to the inexact weight calculations in a neural network and is also referred to as model uncertainty.

In incorporating Bayesian inference into SYMHNet, our goal is to find the posterior distribution over the network's weights, W, given the observed training data,

X, and the labels Y, namely $P(W|X,Y)$. The posterior distribution is intractable [87], and one has to approximate the distribution of the weights [56]. We use variational inference as suggested by [68] to learn the variational distribution over the network's weights, $q(W)$, by minimizing the Kullback–Leibler (KL) divergence of $q(W)$ and $P(W|X,Y)$.

Training a network with dropout [158] is equivalent to a variational approximation on the network [62]. Furthermore, minimizing the cross entropy (CE) loss objective function [67] can have the same effect as minimizing the KL divergence term. Minimizing the CE loss in classification problems is equivalent to the minimization of the mean squared error (MSE) loss in regression problems [83, 99]. Therefore we use the MSE loss objective function and the root mean squared propagation (RMSProp) optimizer with a learning rate of 0.0002 to train SYMHNet. We use $\hat{q}(W)$ to represent the optimized weight distribution.

During testing, SYMHNet leverages the Monte Carlo (MC) dropout sampling technique [62] to quantify uncertainty. Specifically, we process the test data K times to generate K MC samples where K is set to 100 in this study. Each time, a set of weights is randomly drawn from $\hat{q}(W)$. We obtain a mean and variance for K samples. The mean is the anticipated SYM-H value. In accordance with [87], we split the variance into aleatory and epistemic uncertainties.

4.1.5 Experiments and Results

Evaluation Metrics To evaluate and compare our proposed network to the existing models, we followed the guidelines for geomagnetic index predictions and forecasting presented by [114] and used by previous work [48, 85, 154] and were introduced in the survey conducted by [39] for regression problems. We selected two common performance metrics, the root mean square error (RMSE), R-squared (R^2) which is known as coefficient of determination.

RMSE calculates the difference between prediction and ground-truth for each data point to captures the range of values of the data, in our case the SYM-H index values. RMSE is calculated as follows:

$$\text{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}, \tag{4.3}$$

where m is the total number of testing records in the test set, $\hat{y}_i$ (y_i, respectively) represents the predicted SYM-H index value (observed index value, respectively) at time point i. The smaller the RMSE, the more accurate a method is.

R^2 calculates the amount of variance of the observed data explained by the predicted data and is calculated as follows:

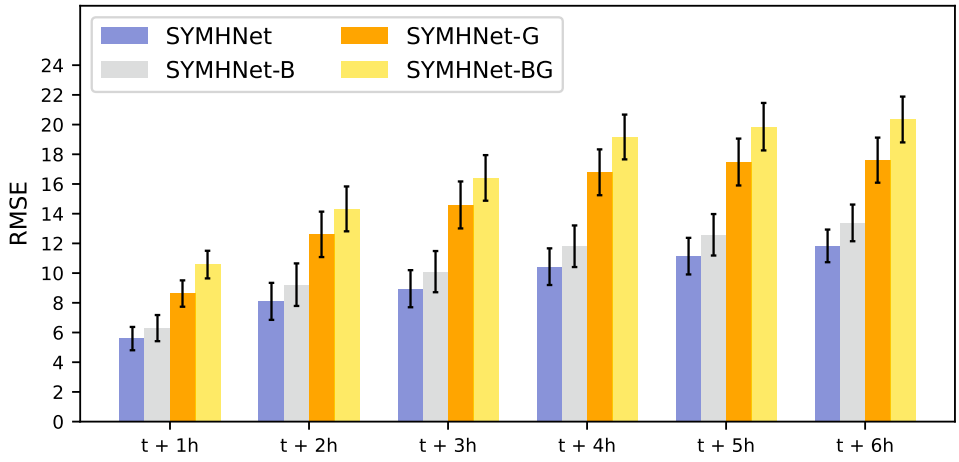
$$R^2 = 1 - \frac{\sum_i^m (y_i - \hat{y}_i)^2}{\sum_i^m (y_i - \bar{y})^2}, \tag{4.4}$$

where $\bar{y}$ is the mean of the observed Dst index values. The larger the R^2, the more accurate a method is.

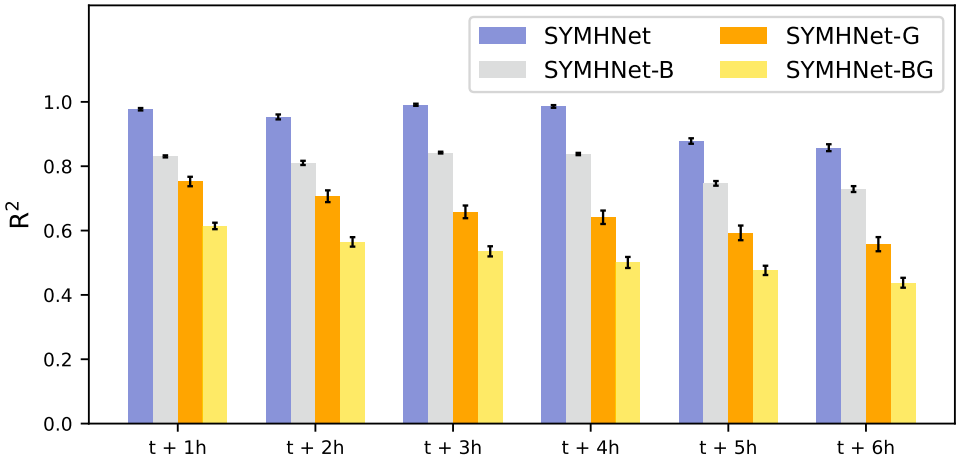
Results Based on 1-Minute Resolution Data Here we present the result for the 1-minute temporal resolution data set. The 1-minute temporal resolution experiments follow the same approach as the 5-minute data sets.

Ablation Study with 1-Minute Resolution Data Similar to Section 4.1.6, here we present the ablation tests we performed in the model to analyze and assess the components of SYMHNet network for 1-minute temporal resolution. We followed the same approach and built three variants from the original SYMHNet: SYMHNet-B, SYMHNet-G, and SYMHNet-BG, turned off the uncertainty quantification mechanism for all sub-networks, and used the same performance metrics: RMSE, R2, and STDE.

Figure 4.3 shows the RMSE and R^2 results for the ablation test comparing the SYMHNet model with the three sub-network variants for 1-minute cadence. It can be seen from the figure that SYMHNet outperforms all the variants. Similar to 5-minute results, the SYMHNet outperforms the sub-network variants, SYMHNet-B is the second in comparison, SYMHNet-G is third, and SYMHNet-BG performs the least among the sub-networks.



(a)



(b)

Figure 4.3 SYM-H 1-minute temporal resolution prediction results from the ablation experiment for 1-6 hour ahead prediction. We are comparing our proposed SYMHNet model with the three sub-network variants for the test storms number 26 to 42. (a) is the RMSE mean and standard deviation error (STDE). (b) is the R-squared mean and standard deviation error (STDE).

Case Study with 1-Minute Resolution Data In this section, we provide the case study for the January 2004 storm, storm number 36 for 1-minute temporal resolution data. First we show the regression fitting plots then the uncertainty quantification results.

Figure 4.4 shows the regression fitting plot between the predicted and observed values of the SYM-M with the correlation coefficient (CC) values for 1-minutes resolution data. The figure shows that our model is performing very well in predicting the storm values and the predicted values are well fitted for 1 to 6 hours ahead.

Figures 4.5 and 4.6 show 1-minute uncertainty quantification for 1-3 and 4-6 hours ahead, respectively. The figure show the entire span of the storm. The red lines in the figures represent the ground-truth observations of the SYM-H values while yellow lines represent the predicted values. The light gray region represents the aleatoric uncertainty (data uncertainty) and light blue region represents the epistemic uncertainty (model uncertainty) for SYM-H index. The figures show that the yellow lines (predicted values) are reasonably close to the red lines (ground truth values) and demonstrate the good performance of SYMHNet which is consistent with the results. Similar to 5-minute uncertainty quantification results, The light gray zone is tinier than the light blue hemisphere, denoting that model uncertainty is lower than data uncertainty. This is due to the fact that the uncertainty in the estimated outcome is primarily caused by noise in the input test data. In addition, the data demonstrate that the model uncertainty for 6-hour-ahead forecasts is marginally greater than for other hours. This is projected because the model has more time to predict.

4.1.6 Results Based on 5-Minute Resolution Data

In this section, we present the results based on the 5-minute resolution as well as provide a comparison results with the previous works. The comparison is only for

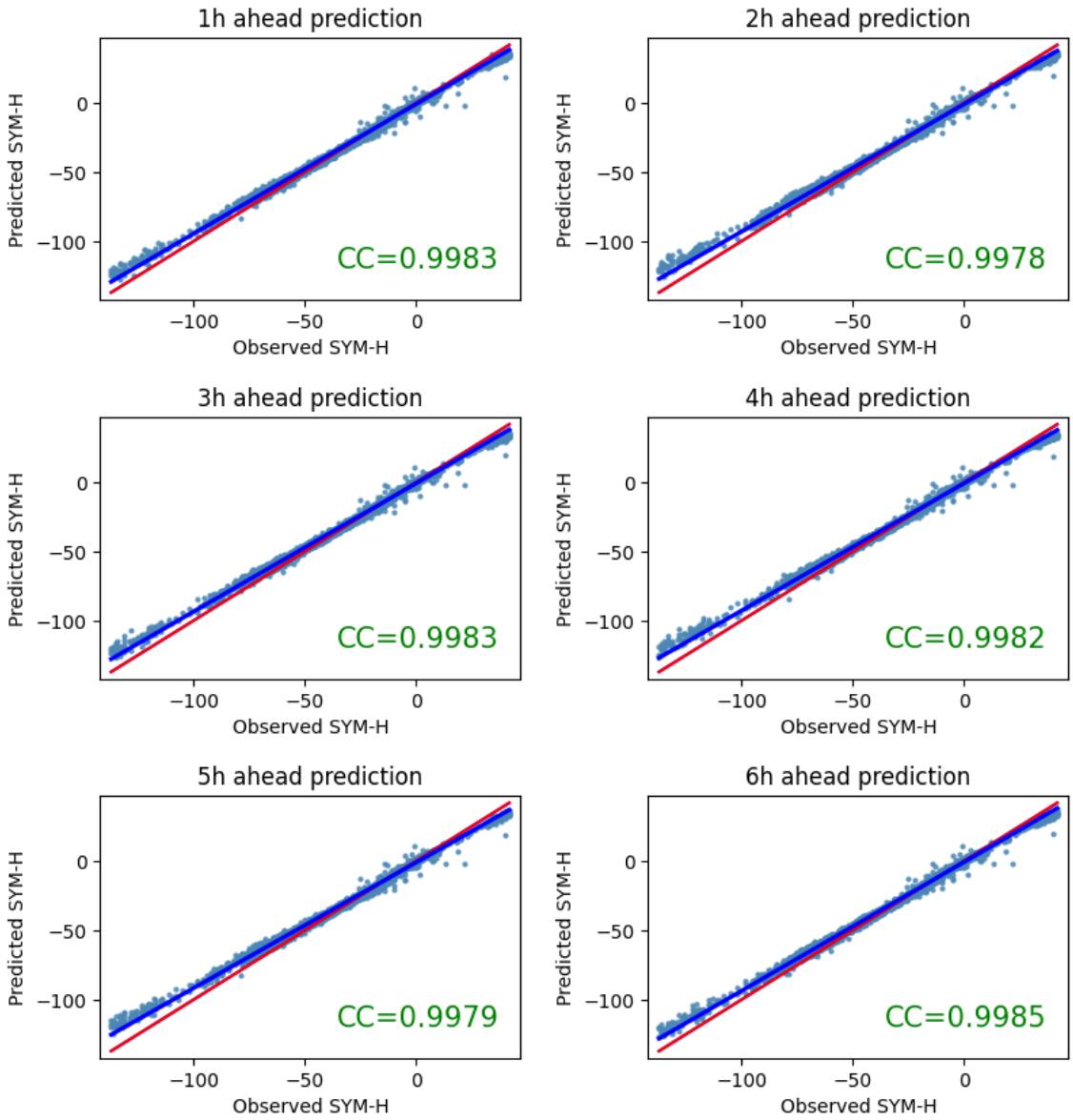


Figure 4.4 SYM-H 1-minute temporal resolution regression plot with correlation coefficient for the January 2004 storm.

5-minute resolution because previous works did not provide prediction for 1-minute resolution.

Comparative Study with 5-Minute Resolution Data We first compared SYMHNet with closely related methods. [48] combined long short-term memory (LSTM) and convolutional neural networks (CNNs), which is referred to as the

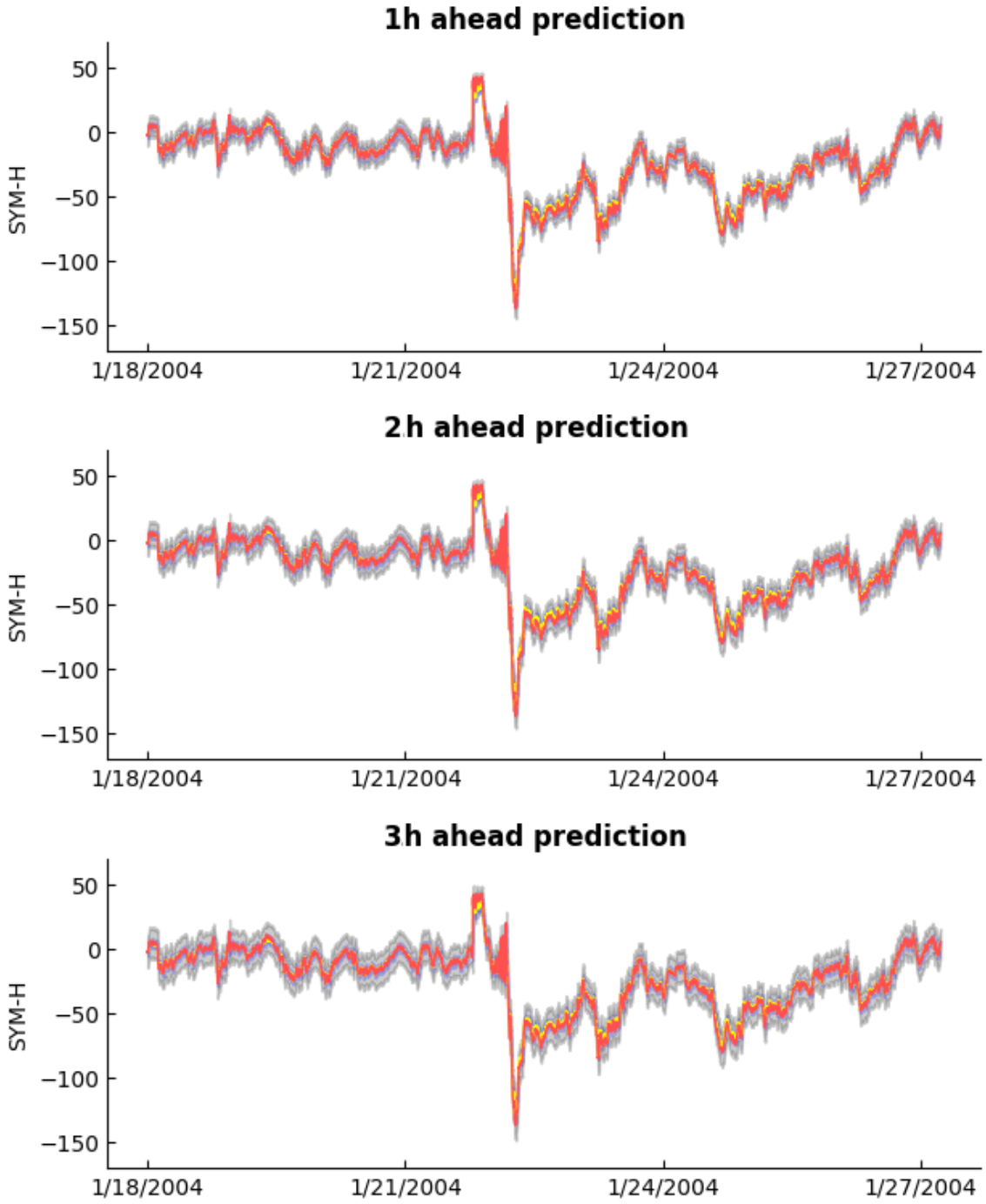


Figure 4.5 SYM-H 1-minute resolution uncertainty quantification results produced by SYMHNet for 1-3 hours ahead for the January 2004 storm. The light gray is the aleatoric (data) uncertainty, the light-blue is the epistemic (model) uncertainty, the red line is the observed SYM-H, and the yellow line is the predicted SYM-H.

LSTM+CNN method to forecast SYM-H for 1 and 2 hours ahead. [85] also forecasts SYM-H using gradient boosting machines for 1 and 2 hours ahead,we referred to as

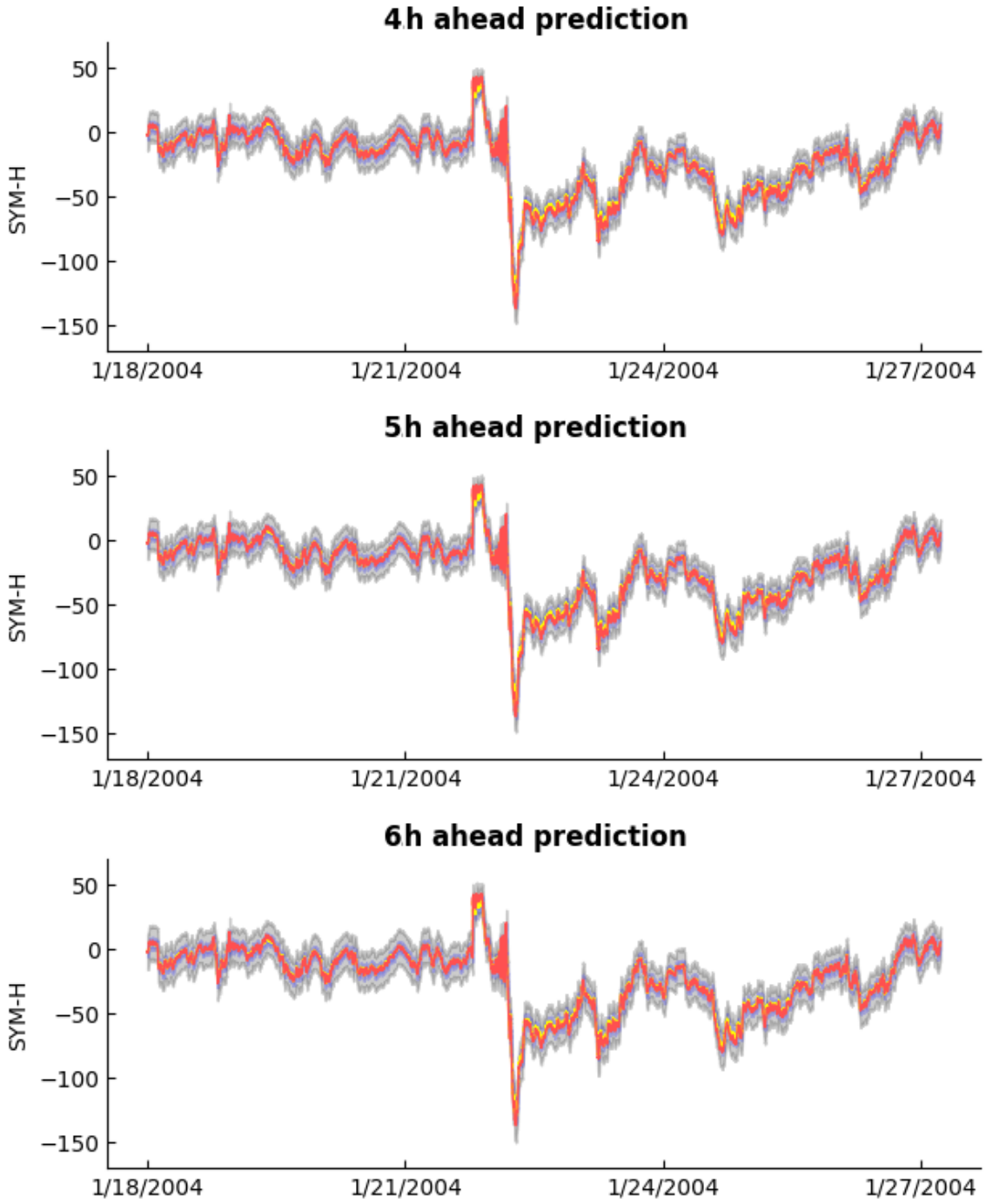


Figure 4.6 SYM-H 1-minute resolution uncertainty quantification results produced by SYMHNet for 4-6 hours ahead for the January 2004 storm. The light gray is the aleatoric (data) uncertainty, the light-blue is the epistemic (model) uncertainty, the red line is the observed SYM-H, and the yellow line is the predicted SYM-H.

the GBM method. [154] provided an SYM-H forecasting comparison between long short-term memory (LSTM) and convolutional neural networks (CNN) for 1 hour

ahead only. During our experiments, we use the results provided by each method for the available number of hours ahead. Our proposed algorithm provides up to 6 hours ahead forecasting that are not provided by other methods, therefore, there are no result to compare, but we show how our method behaves in forecasting the SYM-H index in these additional hours. Also, [85] did not provide R2 results.

In addition, The related methods did not produce uncertainty quantification results, so we also turned off the uncertainty quantification component in SYMHNet.

Tables 4.4 and 4.5 present the RMSE results and Table 4.6 shows the R2 results for the 5-minute temporal resolution comparison between our proposed method SYMHNet model and the related methods LSTM+CNN [48], GBM [85], and LSTM and CNN from [154] for the test storms number 26 to 42.

The tables show that our SYMHNet network outperforms competing models for which results are available. Also, our additional number of hours of forecasting are inline with the compared results and show that our model can perform very well in all of the number of hours in the study.

Ablation Study with 5-Minute Resolution Data In this section, we show the ablation tests we performed in the model to analyze and assess the components of SYMHNet network. In order to create different variations of the network, we modify the layers in the variants where appropriate to accept input data fed to the sub-network. We built three different variants or subnets from the original SYMHNet: SYMHNet-B, SYMHNet-G, and SYMHNet-BG. SYMHNet-B represents a sub-network without the BiLSTM components while keeping the GNN and other components. SYMHNet-G represents a sub-network without the GNN components while keeping the BiLSTM and other components. SYMHNet-BC represents the sub-network without the BiLSTM and GNN components while keeping only the dense layers to simulate a simple multi-layer perceptron network. To perform this

Table 4.4 RMSE Results for the 1h Ahead Predictions Over the Test Storms Set Obtained by the Machine Learning Methods Considered in This Study

	1h ahead prediction (RMSE)				
Storm#	SYMHNet	LSTM+CNN	GBM	LSTM	CNN
26	**4.105**	6.630	5.863	6.700	7.200
27	**7.077**	8.913	7.729	8.900	10.500
28	**3.251**	5.858	4.281	5.400	5.600
29	**5.376**	6.683	5.833	7.200	7.700
30	**4.746**	5.200	4.927	5.600	6.500
31	**7.846**	8.584	8.277	10.700	9.600
32	**5.359**	7.259	6.841	8.300	8.200
33	**12.177**	13.340	14.492	16.300	19.100
34	**5.754**	10.034	10.190	11.300	12.400
35	**6.148**	7.693	7.154	8.500	8.800
36	**6.552**	9.525	8.512	8.700	10.500
37	**13.975**	15.184	14.548	17.500	17.300
38	**3.479**	4.080	3.886	4.200	4.600
39	**5.546**	6.431	5.901	5.600	6.800
40	**4.644**	4.673	4.976	5.500	5.900
41	**7.103**	7.882	7.558	9.000	9.400
42	**4.926**	5.669	5.030	5.900	6.300

experiment, the uncertainty quantification mechanism was turned off for SYMHNet and the three sub-networks.

Figure 4.7 shows the RMSE and R^2 results for the ablation test comparing the SYMHNet model with the three sub-network variants where error bars are as defined in [9]. It can be seen from the figure that SYMHNet outperforms all the variants. The SYMGNet-B is the second best in comparison. GNN is already proven to perform well in regression problems [26]. Therefore, GNN is performing well without BiLSTM but has a degradation of 5-10% compared to the full model. SYHMNet-G is third in comparison and is not performing very well without the GNN component. This is in sync with the finding introduced by [48], and it can be seen that SYMHNet-G is also performing less than LSTM+CNN. Finally, SYMHNet-BG is lowest in performance because it loses the advantage of the GNN and BiLSTM components. It should also be noted that the performance of all variants including the full model is degraded for

Table 4.5 RMSE Results for the 2h Ahead Predictions Over the Test Storms Set Obtained by the Machine Learning Methods Considered in This Study

	2h ahead prediction (RMSE)		
Storm #	SYMHNet	LSTM+CNN	GBM
26	**4.511**	8.989	8.285
27	**8.765**	13.418	11.585
28	**5.286**	5.877	5.650
29	**6.061**	9.314	8.826
30	**5.435**	7.288	7.280
31	**12.097**	12.436	12.613
32	**8.049**	8.937	9.927
33	**17.022**	18.481	24.519
34	**6.239**	13.941	13.736
35	**6.539**	9.932	9.504
36	**6.702**	12.058	12.068
37	**16.676**	21.084	22.327
38	**3.788**	5.213	5.153
39	**6.636**	6.798	7.391
40	**5.030**	5.281	5.633
41	**11.140**	11.707	12.121
42	**6.216**	8.273	7.976

longer hours ahead prediction. This is because the models have to predict longer time that include large span of the data from the time point t to the h-hour ahead where h ¿ 2. For example, for 5-hour ahead prediction, the span of the data is 5*¡number of records per hour¿ = 5*12 = 60 data points. Similarly for 1-minute resolution, check Section 4.1.5 for details, data points = 5*60 = 300.

Case Study with 5-Minute Resolution Data In this section, we provide a case study for the January 2004 storm, storm number 36. First we show the regression fitting plots then the uncertainty quantification results.

Figure 4.8 shows the regression fitting plot between the predicted and observed values of the SYM-M with the correlation coefficient (CC) values for 5-minute resolution data, respectively. The figure shows that our model is performing very well in predicting the storm values and the predicted values are well fitted for 1 to 6

Table 4.6 R^2 Results for the 1h and 2h Ahead Predictions Over the Test Storms Set Obtained by the Machine Learning Methods Considered in This Study

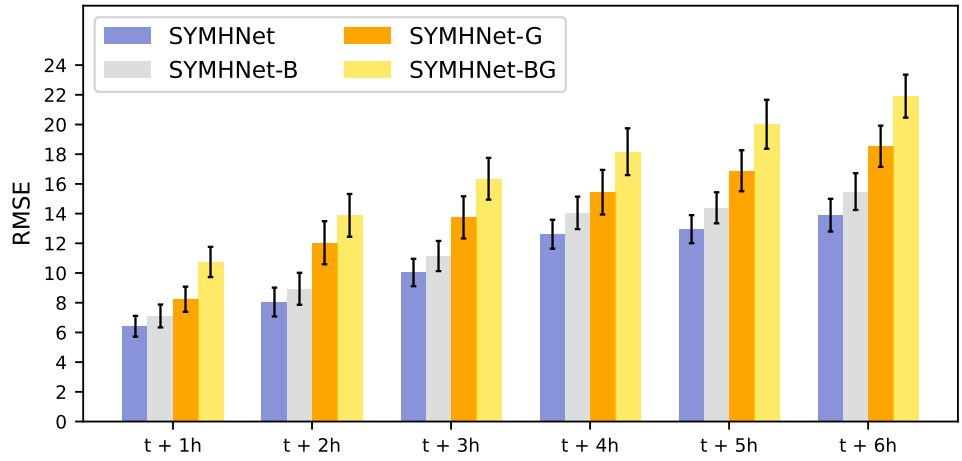
	1h ahead prediction (R^2)				2h ahead prediction (R^2)	
Storm#	SYMHNet	LSTM+CNN	LSTM	CNN	SYMHNet	LSTM+CNN
26	**0.953**	0.870	0.890	0.870	**0.943**	0.766
27	**0.996**	0.939	0.940	0.920	**0.998**	0.862
28	**0.994**	0.936	0.950	0.950	**0.996**	0.936
29	**0.993**	0.922	0.930	0.920	**0.995**	0.848
30	**0.997**	0.946	0.950	0.930	**0.998**	0.894
31	**0.997**	0.971	0.960	0.970	**0.999**	0.939
32	**0.996**	0.953	0.950	0.950	**0.997**	0.929
33	**0.997**	0.965	0.960	0.950	**0.998**	0.932
34	**0.987**	0.798	0.750	0.700	**0.989**	0.612
35	**0.995**	0.907	0.900	0.890	**0.996**	0.845
36	**0.994**	0.864	0.890	0.840	**0.995**	0.782
37	**0.973**	0.966	0.960	0.960	**0.969**	0.934
38	**0.994**	0.939	0.940	0.930	**0.995**	0.900
39	**0.992**	0.932	0.960	0.940	**0.995**	0.924
40	**0.995**	0.966	0.950	0.950	**0.997**	0.957
41	**0.996**	0.969	0.960	0.960	**0.997**	0.931
42	**0.974**	0.968	0.970	0.960	**0.998**	0.932

hours ahead. CC is calculated as follows:

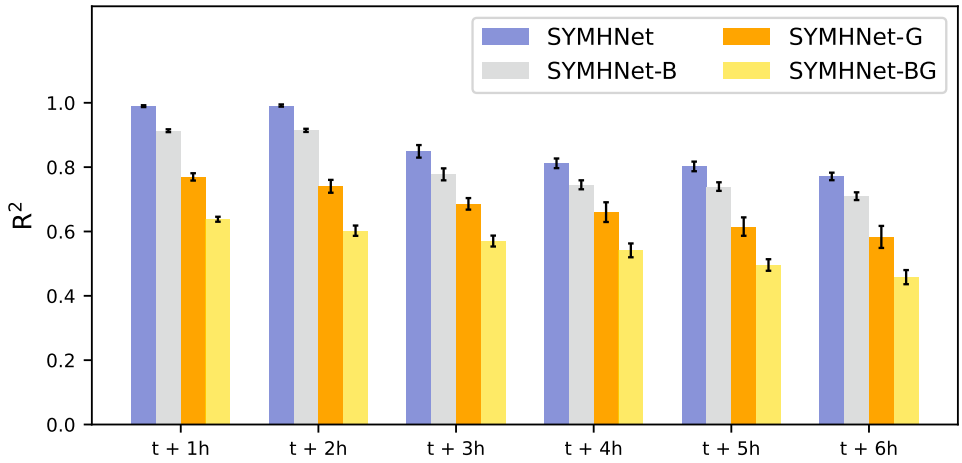
$$\mathrm{CC} = \frac{\sum_i^m (y_i - \mu(\hat{y}))(y_i - \bar{y})}{\sqrt{\sum_{i=1}^m (y_i - \mu(\hat{y}))^2} \sqrt{\sum_{i=1}^m (y_i - \bar{y})^2}} \tag{4.5}$$

where $\mu(\hat{y})$ denotes the mean of all predicted SYM-H values and $\bar{y}$ denotes the mean of all observed SYM-H values. CC is used to measure how strong the relationship between the predicted and actual SYM-H values is. CC ranges from 1 to -1, where 1 means there is a strong positive correlation, -1 means there is a strong negative correlation, and 0 means there is no correlation between the predicted and actual SYM-H values.

Figures 4.9 and 4.10 show the uncertainty quantification for 1-3 and 4-6 hours ahead, respectively. we selected a smaller region of the storm to clearly show the uncertainty shades and make them visible. The red lines in the figures represent the ground-truth observations of the SYM-H values while the prediction are shown



(a)



(b)

Figure 4.7 SYM-H 5-minute temporal resolution prediction results from the ablation experiment for 1-6 hour ahead prediction. We are comparing our proposed SYMHNet model with the three sub-network variants for the test storms number 26 to 42. (a) is the RMSE mean and standard deviation error. (b) is the R-squared mean and standard deviation error.

as yellow lines. The light gray area shows aleatoric uncertainty (data uncertainty) and the light blue area represents epistemic uncertainty (model uncertainty) for the SYM-H index. The graphs reveal that the yellow lines (predicted values) are fairly similar to the red lines (ground truth values) and indicate the good performance of SYMHNet, which is consistent with the outcomes. Moreover, the light gray region is tinier than the light blue region, implying that model uncertainty is less than data

uncertainty. This is due to the fact that the uncertainty in the predicted outcome is primarily caused by noise in the input test data. Furthermore, the figures show that the model uncertainty for 6-hour ahead prediction is slightly larger than other hours. This is expected because of the longer time the model has to predict.

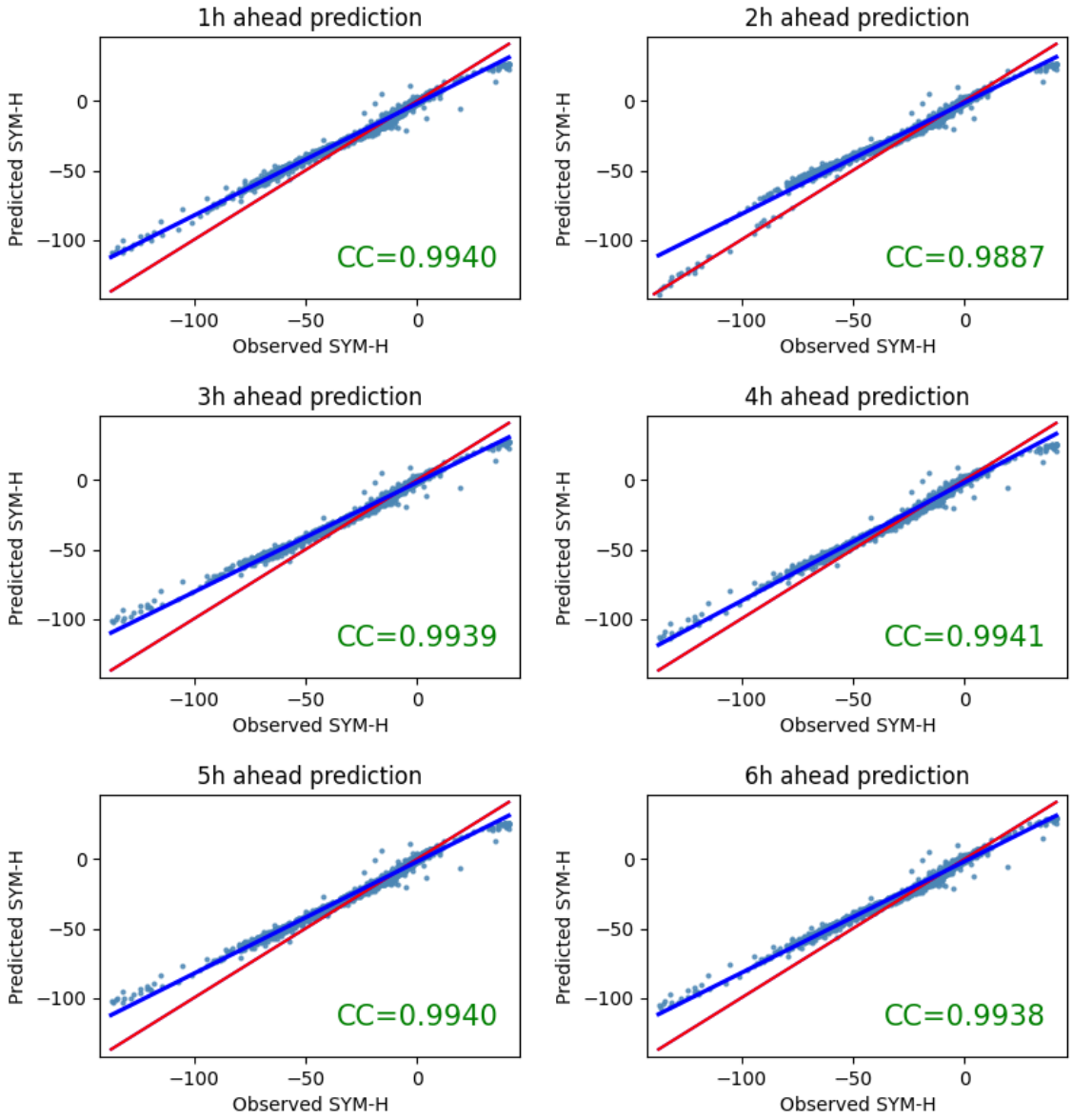


Figure 4.8 SYM-H 5-minute temporal resolution regression plot with correlation coefficient for the January 2004 storm.

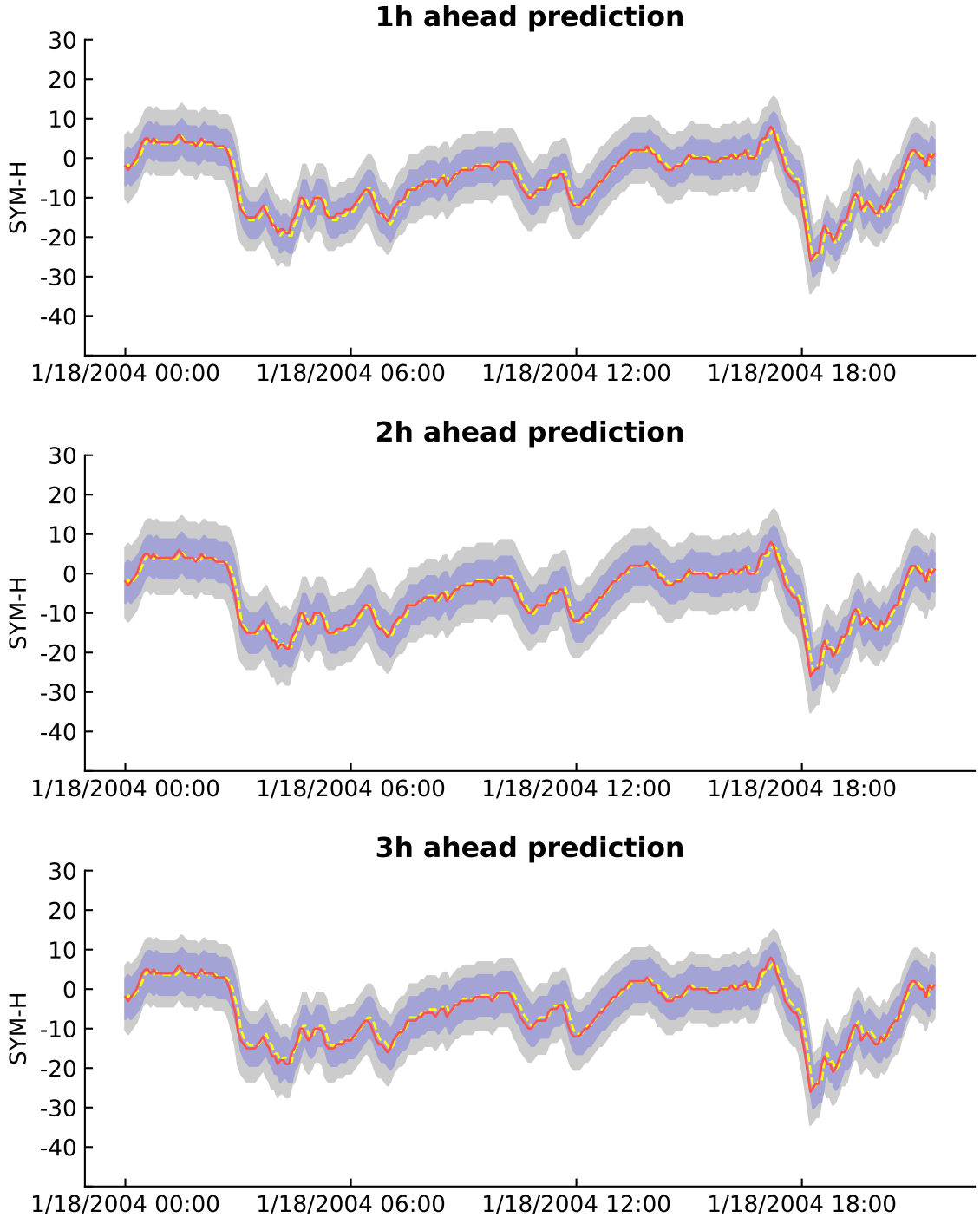


Figure 4.9 SYM-H 5-minute resolution uncertainty quantification results produced by SYMHNet for 1-3 hours ahead for the January 2004 storm. The light gray is the aleatoric (data) uncertainty, the light-blue is the epistemic (model) uncertainty, the red line is the observed SYM-H, and the yellow line is the predicted SYM-H.

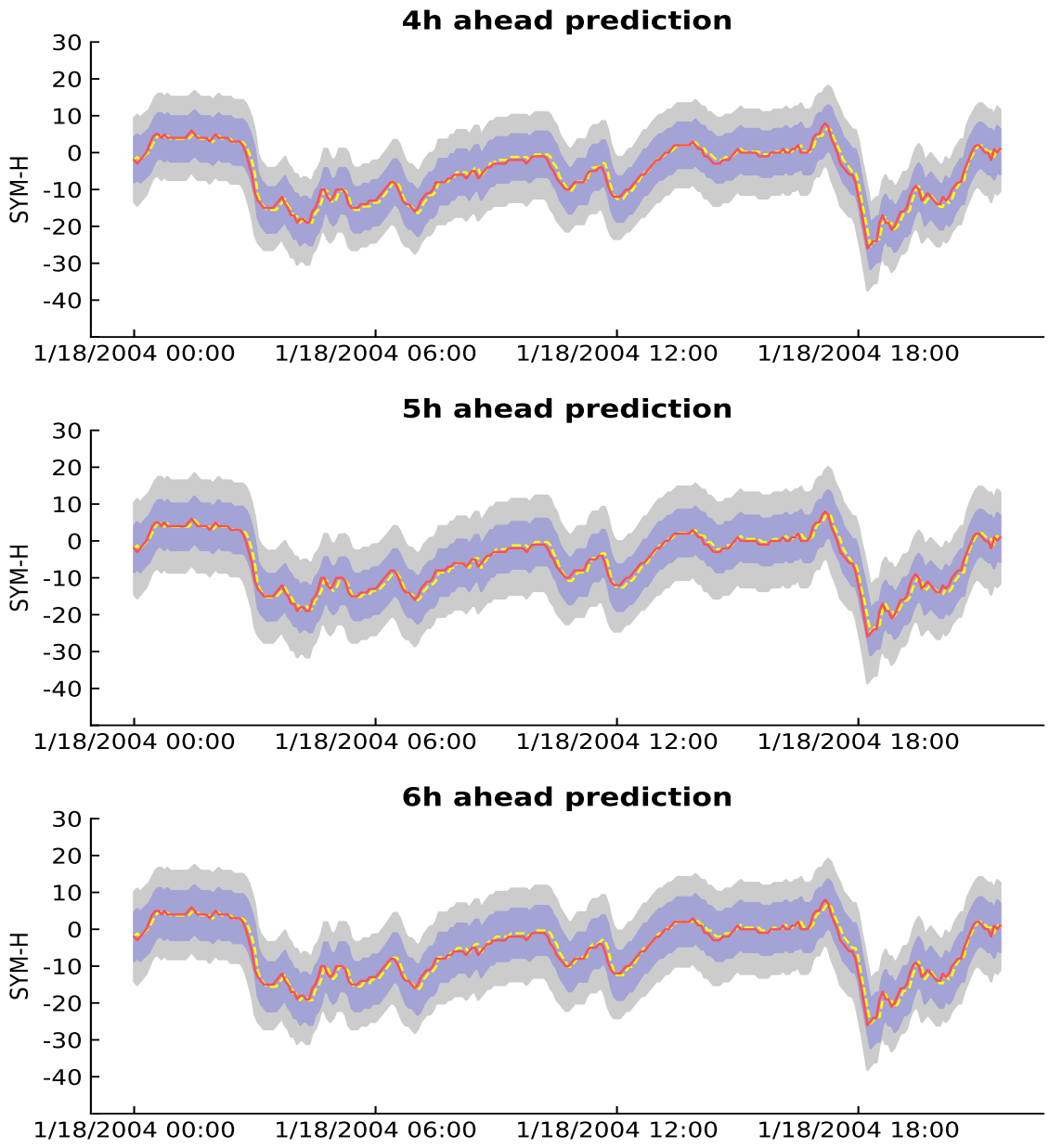


Figure 4.10 SYM-H 5-minute resolution uncertainty quantification results produced by SYMHNet for 4-6 hours ahead for the January 2004 storm. The light gray is the aleatoric (data) uncertainty, the light-blue is the epistemic (model) uncertainty, the red line is the observed SYM-H, and the yellow line is the predicted SYM-H.

4.1.7 Summary

Geomagnetic activities have a significant impact on Earth, which can cause damage on spacecraft, electrical power grids, and navigation systems. Geospace scientists

use geomagnetic indices to measure and quantify the activities. The SYM-H index can be used to explain the information about the response and behaviour of the Earth's magnetosphere during geomagnetic storms activities. Therefore, many work and researches were conducted to perform predictions of the the indices due to their importance to understand the Earth behaviour during the storms. The previous work focused on the 5-minute temporal resolution and skipped the 1-minute temporal resolution data set because of its forecasting difficulties due to the high temporal resolution which poses a more difficult challenge to forecast due to its highly oscillating character. Throughout this investigation, we introduced a novel deep learning model, SYMHNet, for 1 to 6 hour-ahead forecasting of the SYM-H index utilising solar wind parameters from the NASA Space Science Data Coordinated Archive. SYMHNet combines temporal graph neural network and bidirectional long short-term memory with Parameter estimation to quantify aleatoric (data) and epistemic (model) uncertainty when executing SYM-H index prediction tasks. Our findings demonstrate the superiority of SYMHNet over competing machine learning techniques. These results were derived from the information gathered to create two distinct databases. One database contained the 42 storms that occurred during the two solar cycles between 1996 and 2018. The other database was constructed using all data from 1996 to 2018. This dataset was used for the testing of cross validation. To eliminate bias in our results, we conducted additional experiments employing 22-fold cross validation where the data was divided into 22 partitions or folds. Each layer represents one year of testing data, while the rest layers serve as training data. The successive arrangement of data in each layer was retained. One fold was used for testing and the remaining 21 folds were used for training in each run. There were 22 folds, and consequently 22 runs, for which the overall performance metric values were calculated. The results of the 22-fold validation set matched those reported in the study. Furthermore, we extended our work to predict SYM-H using the 1-minute

temporal resolution data set. The model architecture stays the same as shown in Figure 4.2 and described in Section 4.3.2, but with slightly different internal configuration. For 1-minute data set, the model is configured with higher number of neurons in the dense layer, higher percentage in the dropout layer, and larger number of epochs during the training phase. This is because the 1-minute resolution data are known to pose a more difficult challenge to forecast SYM-H due to its highly oscillating character [154]. This behaviour could make the data more noisy to the model than it is in 5-minute resolution data. Therefore, the model requires more iteration with larger neurons in order to learn the internal features and pattern hidden in the data. We also present the result of the 1-minute temporal resolution in Section 4.1.5. The results show that by comparison the 1-minute temporal resolution results have better performance than those for the 5-minute temporal resolution due to the large number of data the model has to learn from.

Consequently, we conclude that the suggested SYMHNet is a viable machine learning method for short-term, 1 to 6 hour-ahead forecasts of the SYM-H index for both 1 and 5-minute-resolution temporal data sets.

4.2 Prediction of Kp Index

4.2.1 Background and Related Work

Geomagnetic activities have a crucial impact on Earth, which can affect spacecraft, electrical power grids, observers of the aurora, and navigation systems. They are very complex, and their scales vary in orders of magnitude. Geomagnetic activity modeling and forecasting has therefore been an important subject in space weather research.

Generally, geomagnetic indices are indicators of disturbances in the Earth's magnetic field. One important geomagnetic index, named the Kp index, is used to describe the overall level of geomagnetic activity. Kp values range from 0 to 9.

The U.S. Space Weather Prediction Center (SWPC) at the National Oceanic and Atmospheric Administration (NOAA) classifies the geomagnetic activities into four categories: (i) quiet to unsettled (Kp < 4), (ii) active (Kp $= 4$), (iii) minor storm (Kp $= 5$), and (iv) major to severe storm (Kp > 5). A storm occurs when there is an exchange of energy from the solar wind to the near-Earth space environment. At any period, if the Kp index indicates a high geomagnetic activity (e.g., a storm), SWPC issues a warning or alert for the affected parties.

Several machine learning-based methods have been developed to forecast the Kp index. Costello [51] designed and implemented a neural network (NN)-based operational system, whose prediction results are posted on the NOAA website.[2] Boberg *et al.* [28] described a real-time NN algorithm for making three-hour predictions of the Kp index. Shprits *et al.* [152] explored the relative efficiency of solar wind-based predictions and compared long and short term Kp forecasting. Ji *et al.* [86] provided a comparison between NN and SVM (support vector machine) algorithms used for Kp forecasting. Tan *et al.* [160] and later Chakraborty and Morley [41] developed deep learning models for Kp forecasting using long short-term memory (LSTM) networks. In addition to the above Kp prediction methods, several researchers investigated the importance of predictive parameters such as the magnetic component, particle density and velocity, and total magnetic field B to understand the impact of each of the parameters on Kp forecasting performance [191].

In this research, we present a new transformer-based Bayesian deep learning model, named KpNet, for Kp prediction. Our work makes several contributions to the field, listed below.

- KpNet is the first model to utilize transformers to forecast the Kp index for a short-term (*i.e.*, 1-9 hours ahead) period.

[2]Retrieved on 09/01/2022 from `https://www.noaa.gov/`

- With Bayesian inference, KpNet can quantify both aleatoric and epistemic uncertainties in producing Kp predictions. Aleatoric uncertainty, also known as data uncertainty, measures the noise inherent in data. Epistemic uncertainty, also known as model uncertainty, measures the uncertainty in the parameters of a model [93]. To our knowledge, no previous Kp prediction method can quantify both types of uncertainty.

- KpNet performs better than closely related machine learning methods for short-term Kp prediction in terms of both the root mean square error (RMSE) and R-squared (R^2).

4.2.2 Methodology

Data Collection: The Kp measurements used in this study are provided by the GFZ German Research Centre for Geosciences.[3] The Kp values in the GFZ site are collected with a 3-hour cadence where the values range from 0 to 8.33. The solar wind parameters used in this study are taken from the NASA Space Science Data Coordinated Archive.[4] We collect the data in the time period between January 1, 2010 and March 31, 2022. We select the time resolution of the hourly average for the solar wind parameters. Following [160, 191], we consider eight solar wind parameters, namely the interplanetary magnetic field (IMF) magnitude average, magnetic field Bx, By, and Bz components, plasma temperature, proton density, plasma speed, and flow pressure. Due to the difference in cadence, where Kp uses a 3-hour cadence whereas the solar wind parameters use a 1-hour cadence, we process the data by temporally matching the Kp measurements from the GFZ site with the solar wind parameters from the NASA site to create the final dataset.

Data Labeling: We divide our dataset into two parts: training set and test set. The training set contains the records from January 1, 2010 to December 31, 2021. The test set contains the records from January 1, 2022 to March 31, 2022. The training set and test set are disjoint. The records are labeled as follows. Let t be a given time

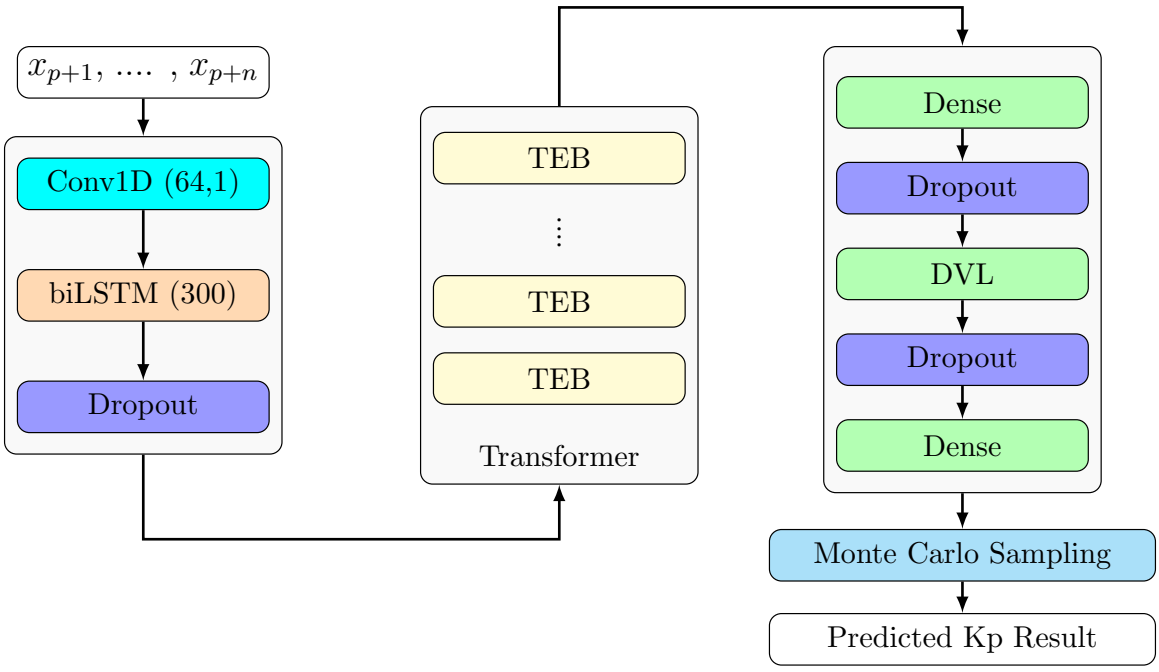
[3]Retrieved on 09/01/2022 from `https://www.gfz-potsdam.de/en/kp-index/`
[4]Retrieved on 09/01/2022 from `https://nssdc.gsfc.nasa.gov`

point and let w be the time window ahead of t, where w ranges from 1 to 9 hours for the short-term Kp forecasting studied here. The label of the record at time point t is defined as the Kp value at time point $t + w$ for w-hour-ahead forecasting. Each training record at time point t in the training set has nine values including eight solar wind parameter values at time point t and the label of the training record. Each test record in the test set contains only eight solar wind parameter values; the label of the test record in the test set will be predicted by our KpNet model.

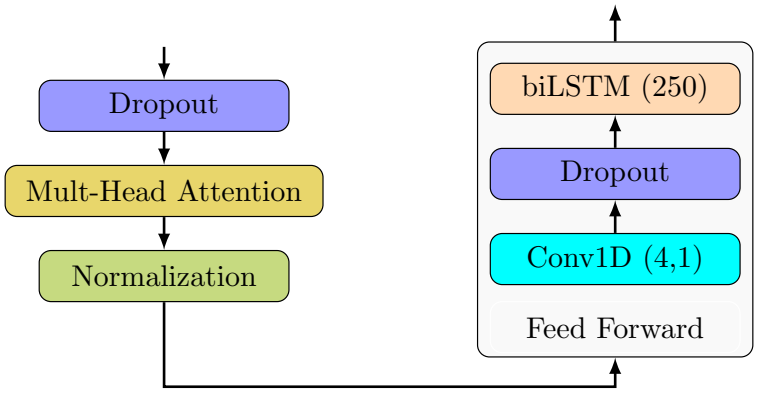
The KpNet Architecture: Figure 4.11 presents the architecture of KpNet, which is created using the keras framework from tensorflow.[5] To enhance the KpNet learning capability and its performance, we add multiple layers to the network. The input of KpNet consists of non-overlapping sequences of records x_{p+1}, x_{p+2}, ..., x_{p+n}, where n is set to 512 in our study. The sequences are passed to a one-dimensional convolution (Conv1D) layer with 64 kernels where the size of each kernel is 1. Conv1D was proven to be well suited for sequential data and was also previously used for geomagnetic index prediction [155]; it learns internal patterns from the input data sequence and passes them to a bidirectional long short-term memory (biLSTM) layer that is configured with 300 neurons. Combining Conv1D and biLSTM layers has shown substantial improvement in performance when dealing with time series as our ablation studies show later.

The biLSTM layer transfers the learned patterns down to a transformer network, which is composed of b transformer encoder blocks (TEBs).[6] Each TEB consists of a multi-head attention layer [169], a batch normalization layer, and a feed-forward network. Generally, transformers for natural language processing (NLP) use layer normalization leading to significant performance gains over batch normalization [169]. However, we use batch normalization here to avoid the effect of outliers in time series

[5]Retrieved on 09/10/2022 from `https://www.tensorflow.org`
[6]Following [169], we set b to 8 in this study.



(a)



(b)

Figure 4.11 (a) Architecture of KpNet. (b) Configuration details of a transformer encoder block (TEB) used in KpNet. Each TEB has two dropout layers, a multi-head attention layer, a batch normalization layer, followed by a feed forward network that contains a Conv1D layer, and a bidirectional LSTM (biLSTM) layer.

which do not exist in NLP word embedding [187]. The multi-head attention layer provides transformation on the sequential input values to obtain distinct metrics of size h. Here, h is the number of attention heads that is set to 4 and the size

of each attention head is also set to 4; the other parameters are left with their default values. The feed-forward network, composed of a Conv1D layer, with 4 kernels where the number of kernels equals the number of attention heads and each kernel size is 1, followed by a biLSTM layer with 250 neurons, helps reduce over-fitting. Notice that each TEB uses a transformer encoder without the decoder because we are dealing with time series instead of language processing and therefore translation is not involved [187].

Furthermore, a dense variational layer (DVL) [163] with 10 neurons is added that uses variational inference [25] to approximate the posterior distribution over the model weights. DVL is similar to a regular dense layer but requires two input functions that define the prior and posterior distributions over the model weights. DVL allows KpNet to represent the weights by a distribution instead of estimated points. In addition, KpNet includes several dense and dropout layers. Each dense layer is strongly connected with its preceding layer where every neuron in the dense layer is connected with every neuron in the preceding layer. Each dropout layer provides a mechanism to randomly drop a percentage of hidden neurons to avoid over-fitting, as explained below.

4.2.3 Experiments and Results

We conducted a series of experiments to evaluate our proposed KpNet model and compare it with closely related methods. The performance metrics used in the study are the root mean square error (RMSE) and R-squared (R^2) [5]. The smaller (larger, respectively) RMSE (R^2, respectively) a method has, the more accurate the method is.

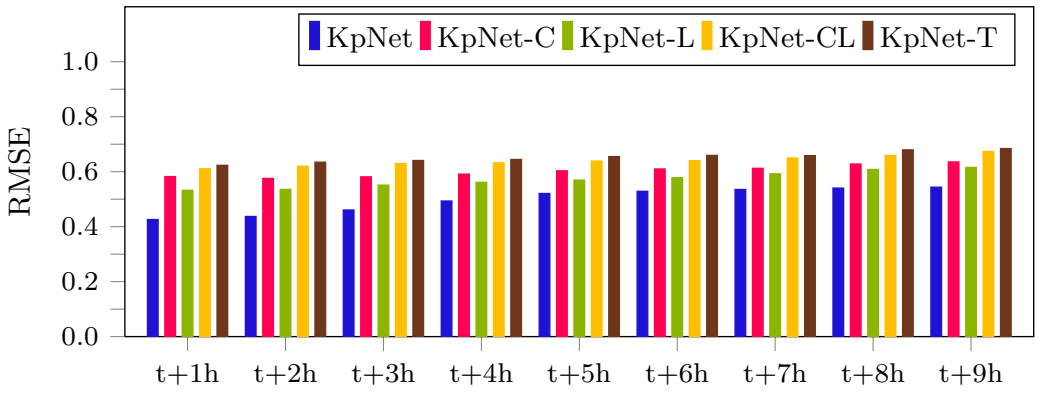
4.2.4 Ablation Studies

In this experiment, we performed ablation tests to analyze and assess the components of the proposed KpNet model. We considered four subnets derived from KpNet: KpNet-C, KpNet-L, KpNet-CL, and KpNet-T. KpNet-C (KpNet-L, KpNet-CL, KpNet-T, respectively) represents the subnet of KpNet in which we remove the Conv1D (64, 1) layer (biLSTM (300) layer, Conv1D (64, 1) and biLSTM (300) layers, transformer network, respectively) while keeping the remaining components of the KpNet framework. For comparison purposes, we turned off the uncertainty quantification mechanism in the five models.

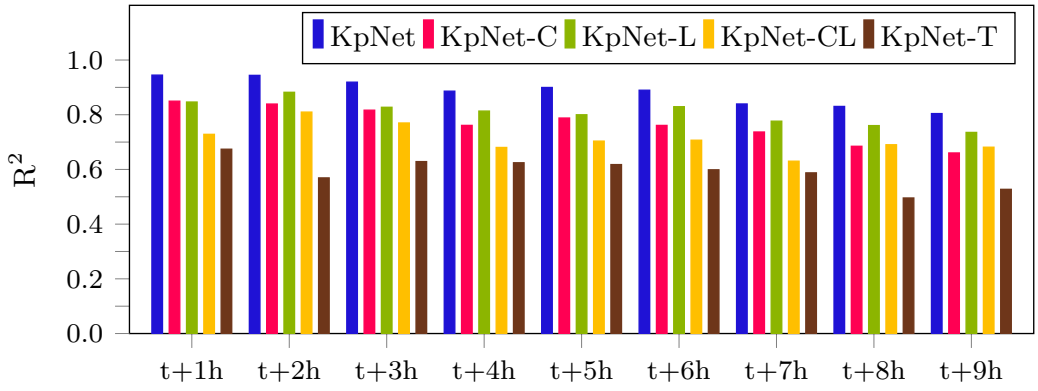
Figure 4.12 presents the ablation test results of the five models. The $t + w$h, $1 \leq w \leq 9$, on the X-axis represents the w-hour ahead prediction of the Kp index based on the test records in the test set. It can be seen from Figure 4.12 that the full network model, KpNet, achieves the best performance among the five models. KpNet-C captures the temporal correlation from the input data but does not learn additional patterns and properties to strengthen the relationship between the test records. KpNet-L captures the properties from the test records but lacks the temporal correlation information to deeply analyze the sequential information in the input data. We also see from Figure 4.12 that removing both of the two layers, Conv1D and biLSTM, decreases the performance significantly. Overall, KpNet-T is the worst model, indicating the importance of the transformer network. The results based on RMSE and R^2 are consistent. In subsequent experiments, we used the full network model, KpNet, due to its best performance among the five models.

4.2.5 Comparison with Related Methods

In this experiment, we compared KpNet with four related machine learning methods that have been used in previous Kp forecasting studies, including linear regression (LR) [160], support vector regression (SVR) [86], a long short-term memory (LSTM)



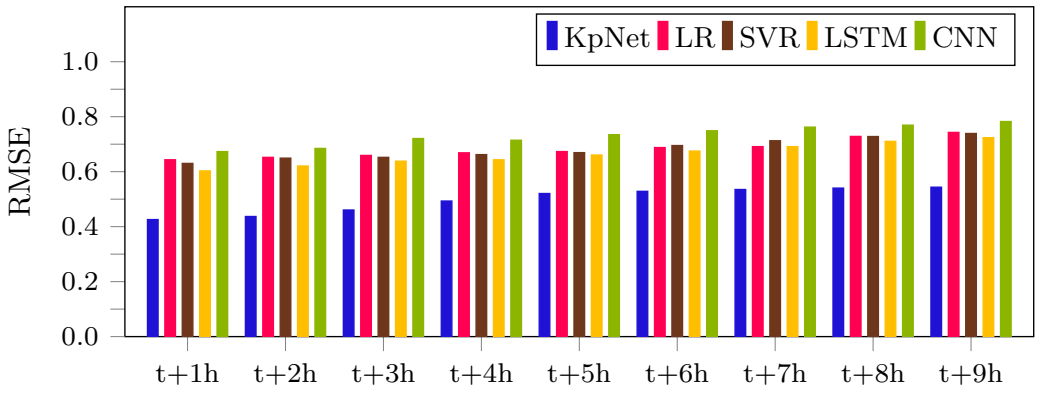
(a)



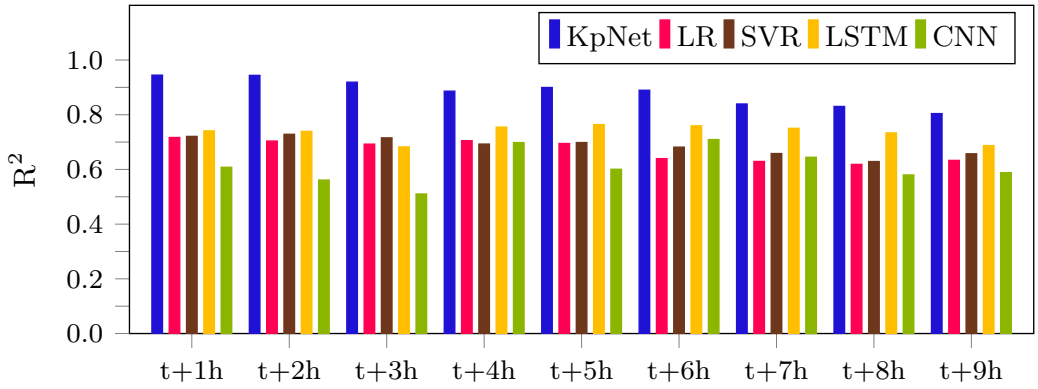
(b)

Figure 4.12 Results of the ablation tests obtained by removing components of KpNet.

network [160], and a convolutional neural network (CNN) that has also been used for geomagnetic index prediction [155]. Since the four methods do not have the ability to quantify both data and model uncertainties, we turned off the uncertainty quantification mechanism in our KpNet model when performing this experiment. Figure 4.13 presents the results. It can be seen from the figure that KpNet achieves the best performance, producing the most accurate predictions, among the five methods in terms of RMSE and R^2. This finding is consistent with those in the literature [187], which indicate that transformers often perform better than LSTM and CNN models in time series forecasting.



(a)



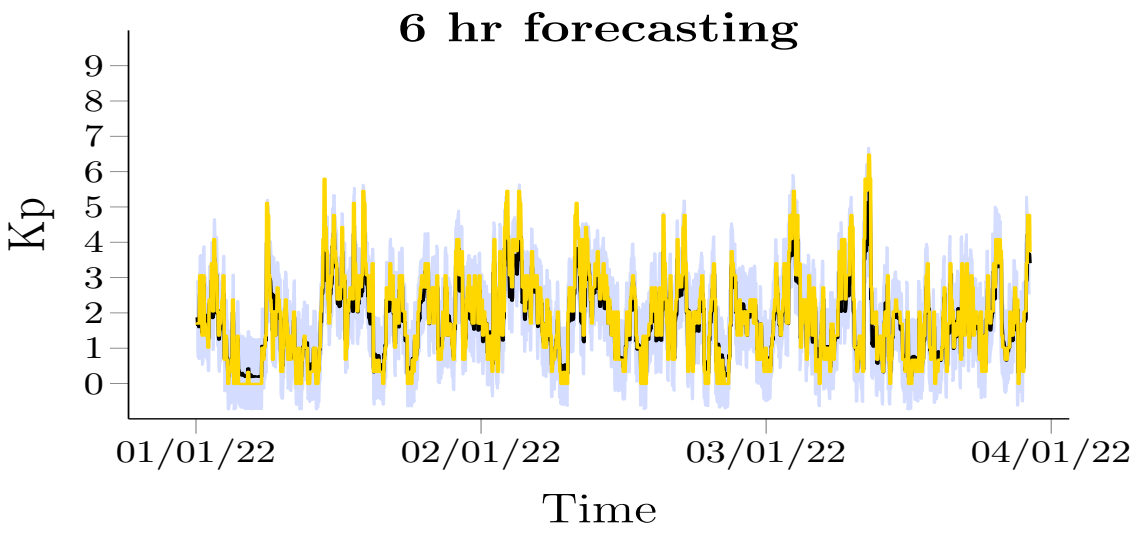
(b)

Figure 4.13 Performance comparison of five Kp prediction methods.

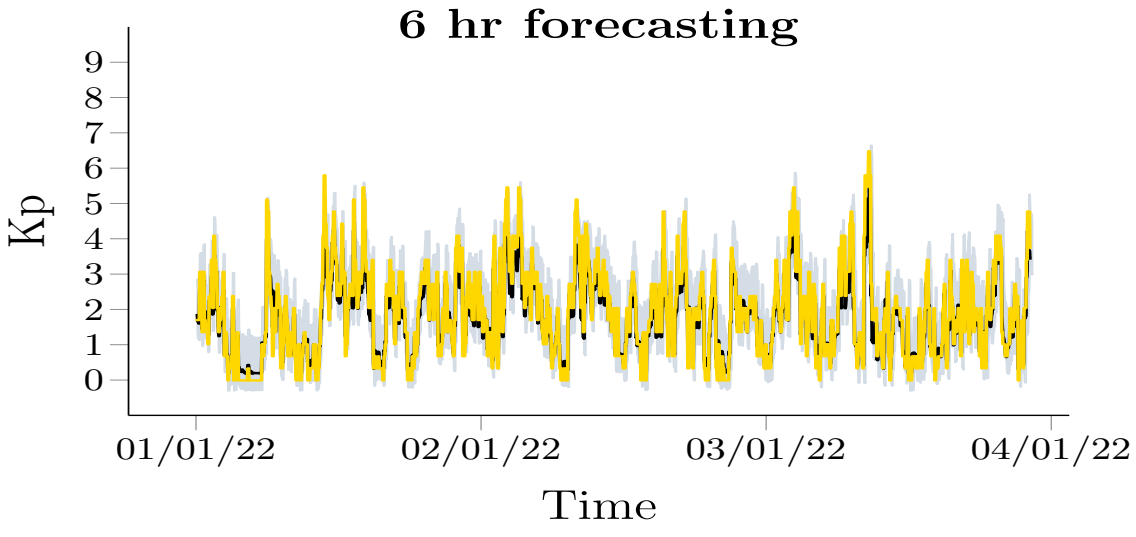
4.2.6 Uncertainty Quantification Results

Figure 4.14 shows uncertainty quantification results produced by KpNet on the test set. Due to space limitation, we only present the results obtained from 6-hour ahead predictions of the Kp index. In Figure 4.14, yellow lines represent observed Kp values (ground truths) while black lines represent predicted Kp values. The light blue region in Figure 4.14(a) represents aleatoric uncertainty (data uncertainty). The light gray region in Figure 4.14(b) represents epistemic uncertainty (model uncertainty). It can be seen from Figure 4.14 that the black lines are reasonably close to the yellow lines, demonstrating the good performance of KpNet, which is consistent with the results in Figure 4.13. Figure 4.14 also shows that the light gray region is much smaller than

the light blue region, indicating that the model uncertainties are much smaller than the data uncertainties. Thus, the uncertainty in the predicted result is mainly due to the noise in the input test data. Similar results were obtained from the other w-hour, $1 \leq w \leq 9$, $w \neq 6$, ahead predictions of the Kp index.



(a)



(b)

Figure 4.14 Uncertainty quantification results produced by KpNet. (a) 6 hr ahead forecasting results with data uncertainty (light blue region). (b) 6 hr ahead forecasting results with model uncertainty (light gray region). Yellow lines represent observed Kp values while black lines represent predicted Kp values.

4.2.7 Summary

We presented a novel deep learning model, named KpNet, to perform short-term, 1-9 hour ahead, forecasting of the Kp index based on the solar wind parameters taken from the NASA Space Science Data Coordinated Archive. KpNet combines transformer encoder blocks with Bayesian inference, which is capable of quantifying both aleatoric uncertainty and epistemic uncertainty when making Kp predictions. Our experimental results demonstrated the good performance of KpNet and its superiority over related machine learning methods. These results were based on the data collected in the period between January 1, 2010 and March 31, 2022. The training set contained hourly records from January 1, 2010 to December 31, 2021. The test set contained hourly records from January 1, 2022 to March 31, 2022. To avoid bias in our findings, we performed additional experiments using 10-fold cross validation where the data was divided into 10 approximately equal partitions or folds. The sequential order of the data in each fold was maintained. In each run, one fold was used for testing and the other nine folds together were used for training. There were 10 folds and hence 10 runs where the average performance metric values over the 10 runs were calculated. Results from the 10-fold cross validation were consistent with those reported in this work. Thus we conclude that the proposed KpNet is a feasible machine learning method for short-term, 1-9 hour, ahead predictions of the Kp index. In the future, we plan to extend KpNet to perform long-term Kp forecasting using other data sources such as solar images in addition to solar wind parameters.

4.3 Prediction of the Disturbance Storm Time Index

4.3.1 Background and Related Work

Geomagnetic activities have significant impact on Earth. They can disturb or damage telephone systems, power grid transmission systems and space satellites. Geomagnetic activity modeling and forecasting has therefore been an important subject in space

weather research. The main source of geomagnetic activity is solar activity. The solar wind, which is a stream of charged particles released from the atmosphere of the Sun, is considered as the medium through which the Sun exerts influence on Earth. As a consequence, solar wind parameters such as the interplanetary magnetic field (IMF), total electric field, solar wind speed and plasma temperature are often used to model geomagnetic activities and forecast geomagnetic indices.

The disturbance storm time (Dst) index is an important geomagnetic index. It has been used to characterize the size and intensity of a geomagnetic storm. A negative Dst value means that the Earth's magnetic field is weakened, which happens during storms. A storm is considered moderate when Dst is greater than -50 nT, intense when Dst is between -50 nT and -250 nT, or super when Dst is less than -250 nT [124, 71].

Many techniques have been developed to model and forecast the Dst index. For example, [36] adopted differential equations to model the Dst index. The authors used solar wind parameters as the source of differential equations in their model. [66] created the first Dst prediction model by employing a time-delay artificial neural network (ANN) with solar wind parameters as input. The authors performed 1-6 hour ahead predictions for the Dst index forecasting. [16] discussed another strategy by combining physical models and ANNs, along with parameters such as the solar wind velocity, IMF magnitude, and IMF clock angle. [106] employed a particle swarm optimization technique to train ANN connection weights to improve the accuracy of Dst index predictions.

The above approaches mainly focused on single point predictions. [43] extended the above approaches by considering probabilistic forecasting of the Dst index. The authors used Gaussian processes (GP) to build autoregressive models to estimate Dst 1 hour ahead based on past Dst values, as well as the solar wind velocity and the IMF Bz component. Their technique generated a predictive distribution instead

of single point predictions. However, the mean values of the forecasts are not as accurate as the forecasts produced by ANNs. To improve GP's poor point prediction performance, [71] built a Dst index prediction model by combining GP with a long short-term memory (LSTM) network.

In this work, we present a novel Bayesian deep learning approach for performing short-term, 1-6 hour ahead, predictions of the Dst index. Our approach, called the Dst Transformer and denoted by DSTT, combines a multi-head attention layer with Bayesian inference capable of handling both aleatoric uncertainty and epistemic uncertainty. Aleatoric uncertainty, also known as data uncertainty, measures the noise inherent in data. Epistemic uncertainty, also known as model uncertainty, measures the uncertainty in the parameters of a model [93]. Thus, our work extends the aforementioned GP-based probabilistic forecasts, which can only handle model uncertainty, to quantify both data and model uncertainties through Bayesian inference. It is worth noting that Bayesian deep learning has also been used to mine solar images [87]. However, Dst values are time series data, not image data, and hence the architecture of our Dst Transformer is totally different from the model architecture developed by [87].

The contributions of our work are summarized below.

- Our DSTT model is the first to utilize the Transformer network to forecast the Dst index for a short-term period (*i.e.*, 1-6 hours ahead).

- This is the first study in which both data and model uncertainties are quantified when performing Dst index forecasting.

- Our DSTT model outperforms closely comparable machine learning methods for short-term Dst index forecasting, as evidenced by performance metrics including the root mean square error (RMSE) and R-squared (R^2).

4.3.2 Methodology

Data Source: The Dst index measurements used in this study are provided by the NASA Space Science Data Coordinated Archive.[7] The data source provides other widely accessed data that are frequently used in solar wind analysis. The data source is being periodically updated with Advanced Composition Explorer (ACE).[8] We used the Dst index data in the time period between January 1, 2010 and November 15, 2021. We selected the time resolution of the hourly average for the Dst index. Following [111], we considered seven solar wind parameters, namely the interplanetary magnetic field (IMF), magnetic field Bz component, plasma temperature, proton density, plasma speed, flow pressure, and electric field. The total number of records in our dataset is 104,080. The Dst index values in the dataset range from 77 nT to -223 nT.

Data Labeling: We divided our dataset into two parts: training set and test set. The training set contains 102,976 records from January 1, 2010 to September 30, 2021. The test set contains 1104 records from October 1, 2021 to November 15, 2021. The training set and test set are disjoint. The records are labeled as follows. Let t be a time point of interest and let w be the time window ahead of t, where w ranges from 1 to 6 hours for the short-term Dst forecasting studied here. The label of the record at time point t is defined as the Dst index value at time point $t + w$ for w-hour-ahead forecasting. Each record in the training set has eight values including the seven solar wind parameter values and the label of the training record. Each record in the test set contains only the seven solar wind parameter values; the label of each testing record in the test set will be predicted by our DSTT model.

[7]Retrieved on 03/15/2022 from `https://nssdc.gsfc.nasa.gov`
[8]Retrieved on 03/15/2022 from `https://omniweb.gsfc.nasa.gov`

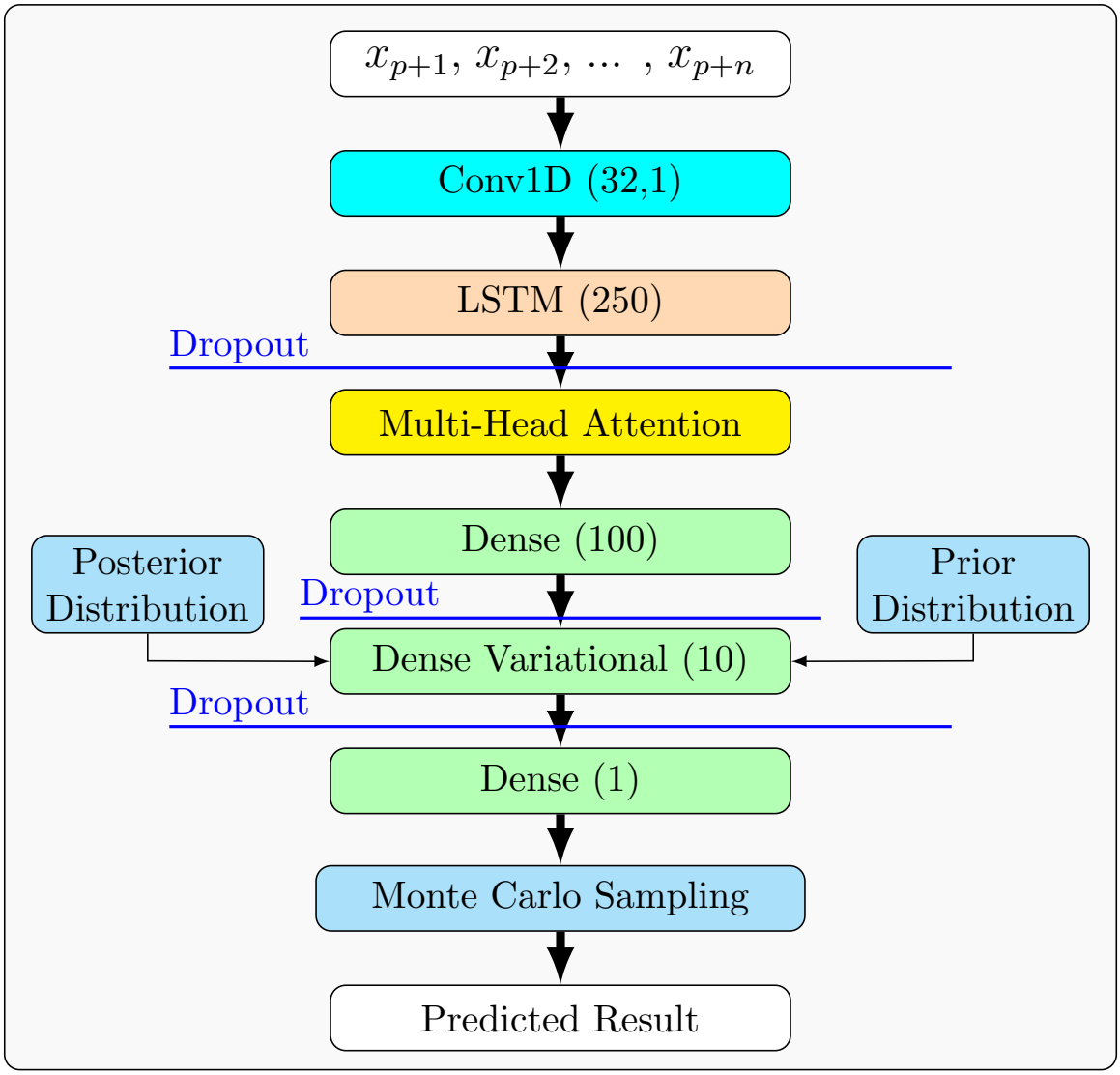


Figure 4.15 Architecture of our Dst Transformer (DSTT).

Architecture of the DSTT: Figure 4.15 presents the architecture of our DSTT model. DSTT is created using the tensorflow keras framework.[9] We add multiple layers to DSTT to enhance its performance and improve its learning capability. The model accepts as input non-overlapping sequences of records $x_{p+1}, x_{p+2}, \ldots, x_{p+n}$, where n is set to 1024 in our study. Each sequence is passed to a one-dimensional convolution neural network (Conv1D) with 32 kernels where the size of each kernel is 1. Conv1D is well suited for sequential data; it learns patterns from the input

[9]Retrieved on 04/01/2022 from `https://www.tensorflow.org`

data sequence and passes them to a long short-term memory (LSTM) layer that is configured with 250 LSTM units. Combining Conv1D and LSTM layers has shown significant improvement in performance when dealing with sequential data such as time series [5, 59]. LSTM hands the learned patterns down to a multi-head attention layer [168]. The multi-head attention layer provides transformation on the sequential input of values to obtain distinct metrics of size h. Here, h is the number of attention heads that is set to 3 and the size of each attention head is also set to 3 because a number greater than 3 caused overhead and less than 3 caused performance degradation. The other parameters are left with their default values.

Furthermore, we add custom attention to instruct the layers to focus and pay more attention to critical information of the input data sequence and capture the correlation between the input and output by computing the weighted sum of the data sequence. In addition, we add a dense variational layer (DVL) [163] with 10 neurons that uses variational inference [25] to approximate the posterior distribution over the model weights. DVL is similar to a regular dense layer, but requires two input functions that define the prior and posterior distributions over the model weights. DVL allows our DSTT model to represent the weights by a distribution instead of estimated points.

The model, DSTT, also includes multiple dense and dropout layers. Each dense layer is strongly connected with its preceding layer where every neuron in the dense layer is connected with every neuron in the preceding layer. Each dropout layer instructs the DSTT model to randomly drop a percentage of its hidden neurons throughout the training phase to avoid over-fitting of training data.

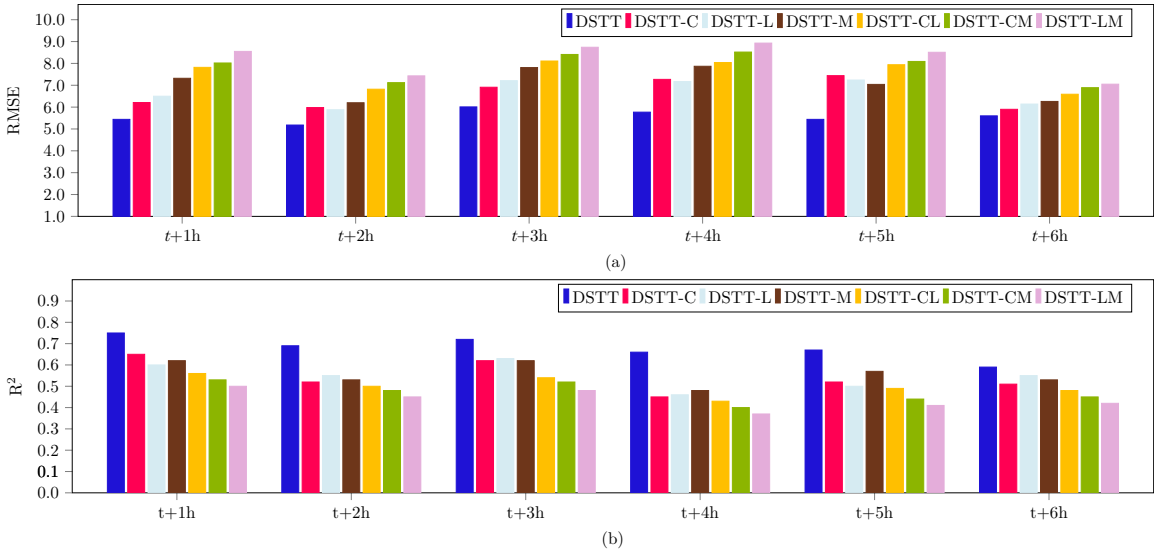


Figure 4.16 Results of the ablation study.

4.3.3 Performance Metrics

We conducted a series of experiments to evaluate our proposed DSTT model and compare it with closely related methods. The performance metrics used in our study are the root mean square error (RMSE) [5] and R-squared (R^2).

RMSE is calculated as follows:

$$\text{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}, \tag{4.6}$$

where m is the total number of testing records in the test set, $\hat{y}_i$ (y_i, respectively) represents the predicted Dst index value (observed Dst index value, respectively) at time point i. The smaller the RMSE, the more accurate a method is.

R^2 is calculated as follows:

$$R^2 = 1 - \frac{\sum_i^m (y_i - \hat{y}_i)^2}{\sum_i^m (y_i - \bar{y})^2}, \tag{4.7}$$

where $\bar{y}$ is the mean of the observed Dst index values. The larger the R^2, the more accurate a method is.

4.3.4 Ablation Study

In this experiment, we performed ablation tests to analyze and evaluate the components of our DSTT model. We considered six subnets derived from DSTT: DSTT-C, DSTT-L, DSTT-M, DSTT-CL, DSTT-CM and DSTT-LM. DSTT-C (DSTT-L, DSTT-M, DSTT-CL, DSTT-CM, DSTT-LM, respectively) represents the subnet of DSTT in which we remove the Conv1D layer (LSTM layer, multi-head attention layer, Conv1D and LSTM layers, Conv1D and multi-head attention layers, LSTM and multi-head attention layers, respectively) while keeping the remaining components of the DSTT network. For comparison purposes, we turned off the uncertainty quantification mechanism in the seven models.

Figure 4.16 presents the RMSE and R^2 results of the seven models. The $t+w$h, $1 \leq w \leq 6$, on the X-axis corresponds to the w-hour ahead predictions of the Dst index based on the testing records in the test set. It can be seen from Figure 4.16 that our proposed full network model, DSTT, achieves the best performance among the seven models. DSTT-C captures the temporal correlation from the input data but it does not learn additional patterns and properties to strengthen the relationship between data records. DSTT-L captures the properties from the data records but it lacks the temporal correlation information to deeply analyze the sequential information in the input data. DSTT-M captures both the temporal correlation and additional properties, but it does not provide transformation on the sequential inputs to obtain distinct metrics to further strengthen the correlation between the predicted and observed Dst values. Similarity, DSTT-CL, DSTT-CM, and DSTT-LM do not capture the combined patterns due to the removed layers. As a consequence, the six subnets achieve worse performance than DSTT. It can be seen from Figure 4.16 that removing two layers yields worse results than removing one layer only. DSTT-LM yields the worst results, indicating the importance of including the LSTM and multi-head attention layers. The results based on RMSE and R^2 are consistent.

In subsequent experiments, we used DSTT due to its best performance among the seven models.

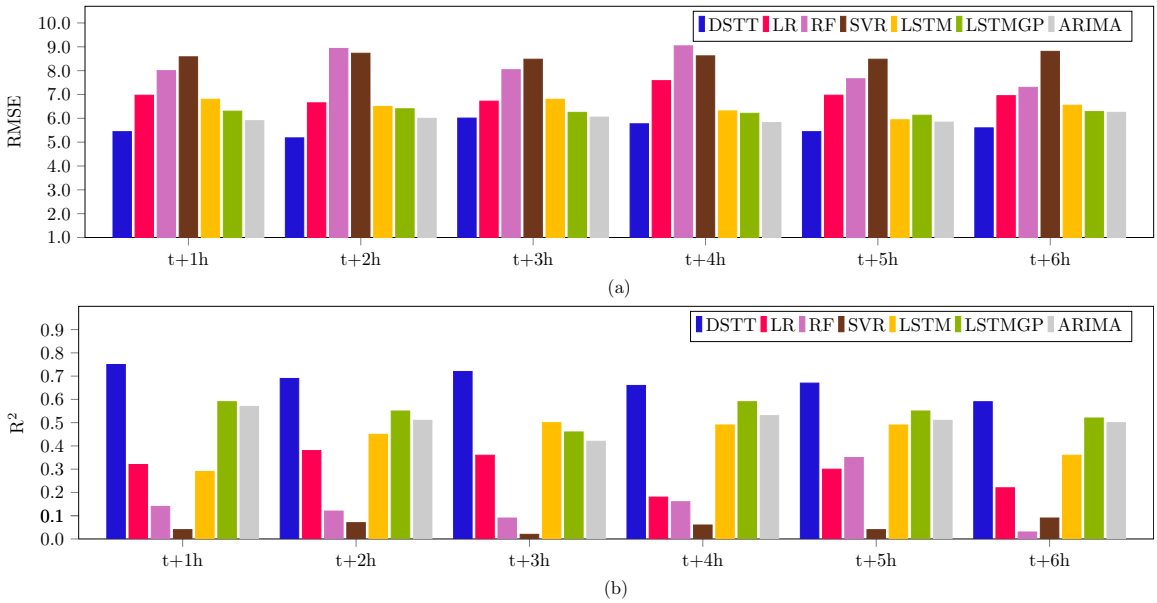


Figure 4.17 Performance comparison of seven Dst forecasting methods.

4.3.5 Comparison with Related Methods

In this experiment, we compared the Dst Transformer (DSTT) with six closely related machine learning methods including linear regression (LR), random forests (RF), support vector regression (SVR), auto regressive integrated moving average (ARIMA) [164], long short-term memory (LSTM), and the method developed by [71], which combines LSTM with Gaussian processes (GP) and is denoted by LSTMGP. Because the six related methods do not have the ability to quantify both data and model uncertainties, we turned off the uncertainty quantification mechanism in our DSTT model when performing this experiment.

Figure 4.17 presents the RMSE and R^2 results of the seven methods: DSTT, LR, RF, SVR, ARIMA, LSTM and LSTMGP. It can be seen from the figure that DSTT achieves the best performance, producing the most accurate predictions, among the seven methods in terms of both RMSE and R^2. The deep learning methods including

DSTT, LSTM and LSTMGP as well as ARIMA mostly perform better than the traditional machine learning algorithms including RF, LR and SVR.

4.3.6 Uncertainty Quantification Results

Figure 4.18 shows uncertainty quantification results produced by our DSTT model on the test set. Due to space limitation, we only present the results obtained from 4-hour ahead predictions of the Dst index. In Figure 4.18, the orange lines represent observed values of the Dst index (ground truth) while the blue lines represent the predicted values of the Dst index. The light blue region in Figure 4.18(a) represents aleatoric uncertainty (data uncertainty). The light gray region in Figure 4.18(b) represents epistemic uncertainty (model uncertainty).

It can be seen from Figure 4.18 that the blue lines are reasonably close to the orange lines, indicating the good performance of our DSTT model, which is consistent with the results in Figure 4.17. Figure 4.18 also shows that the light gray region is much smaller than the light blue region, indicating that the model uncertainties are much smaller than the data uncertainties. Thus, the uncertainty in the predicted result is mainly due to the noise in the input test data. Similar results were obtained from other w-hour, $1 \leq w \leq 6$, $w \neq 4$, ahead predictions of the Dst index. We note that when w is longer than 4, the prediction performance starts to degrade. This is understandable given that we are trying to predict a Dst index value that is farther away from the input test record.

4.3.7 Summary

The disturbance storm time (Dst) index is an important and useful measurement in space weather research, which is used to understand the severity of a geomagnetic storm. The Dst index is also known as the measure of the decrease in the Earth's magnetic field. In this work, we present a novel deep learning model, called the

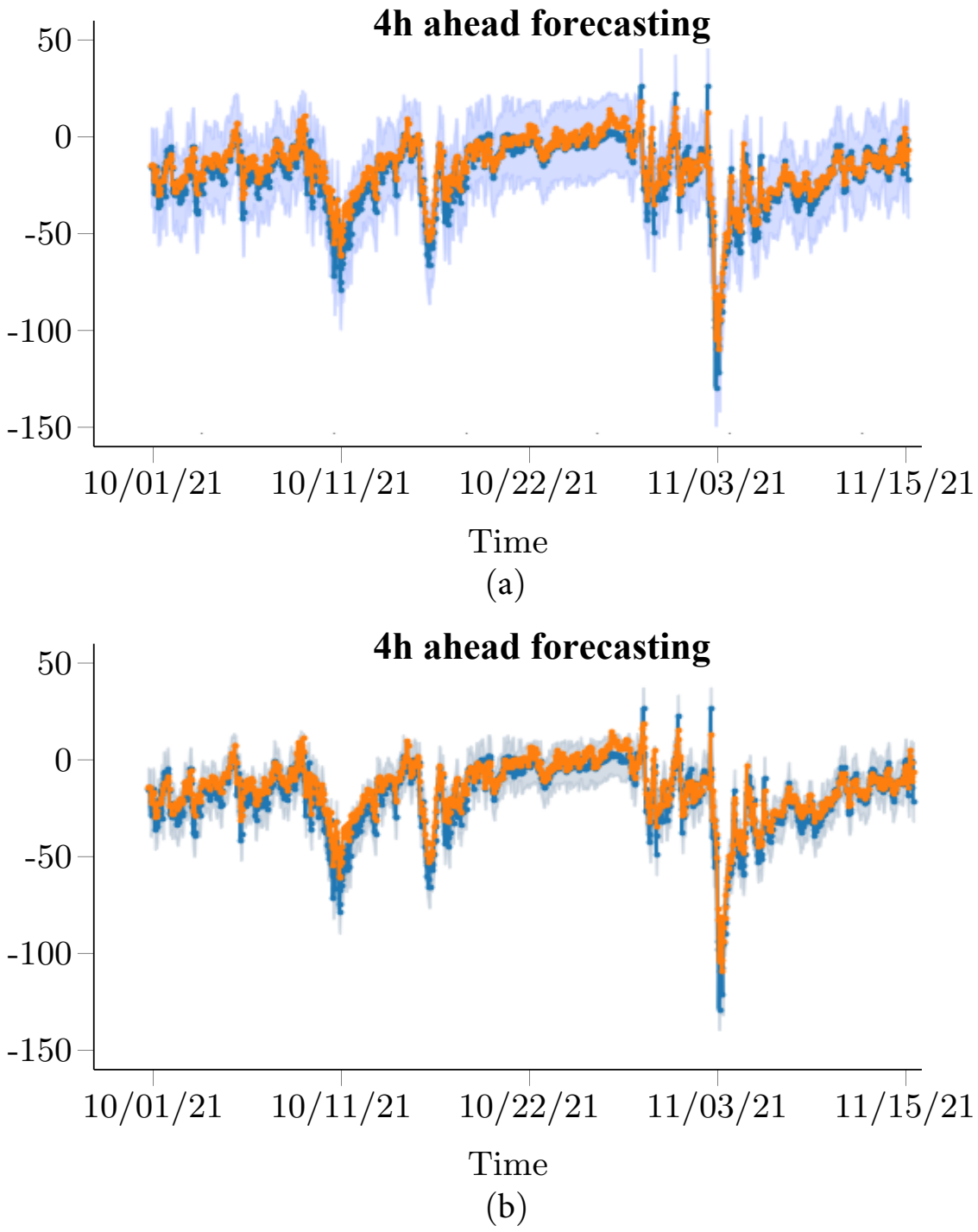


Figure 4.18 Uncertainty quantification results produced by our Dst Transformer (DSTT).

Dst Transformer or DSTT, to perform short-term, 1-6 hour ahead predictions of the Dst index. Our empirical study demonstrated the good performance of the Dst Transformer and its superiority over related methods.

Our experiments were based on the data collected in the period between January 1, 2010 and November 15, 2021. The training set contained hourly records from January 1, 2010 to September 30, 2021. The test set contained hourly records from October 1, 2021 to November 15, 2021. To avoid bias in our findings, we performed additional experiments using 10-fold cross validation (CV). For the CV tests, we used the original data set described above and another data set ranging from November 28, 1963 to March 1, 2022 that has 510,696 records. In addition, we generated synthetic data with up to 1.2 million records to further assess the performance and stability of our DSTT model. With the 10-fold CV tests, the data was divided into 10 approximately equal partitions or folds. The sequential order of the data in each fold was maintained. In each run, one fold was used for testing and the other nine folds together were used for training. There were 10 folds and hence 10 runs. We computed the performance metrics including RMSE and R^2 for each method studied in this research in each run. The means and standard deviations of the metric values over the 10 runs were calculated and recorded. Results from the 10-fold CV tests were consistent with those reported in the research. Thus we conclude that the proposed Dst Transformer (DSTT) is a feasible machine learning method for short-term, 1-6 hour ahead predictions of the Dst index. Furthermore, our DST Transformer can quantify both data and model uncertainties in making the predictions, which can not be done by the related methods.

Our work focuses on short-term predictions of the Dst index by utilizing solar wind parameters. These solar wind parameters are collected by instruments near Earth and are suited for short-term predictions of the geomagnetic storms near Earth [71, 111]. When using the solar wind parameters to perform long-term (e.g., 3-day ahead) predictions of the Dst index, the accuracy is low. In the future work, we plan to perform long-term predictions of the Dst index by utilizing solar data collected by instruments near the Sun. The solar data reflects solar activity, which is the source of

geomagnetic activity. We plan to extend the Bayesian deep learning method described here to mine the solar data for performing long-term Dst index forecasts.

CHAPTER 5

RECONSTRUCTION OF SOLAR IRRADIANCE

5.1 Background and Related Work

Deep learning has drawn significant interest in recent years. It has been used extensively in biomedical applications for Parkinson's disease assessment [72], drug-disease interaction learning [44], drug-drug interaction prediction [116], clinical event prediction [138], breast cancer subtype classification [143], medical image segmentation [189] and so on. It finds many applications in other domains as well. For example, deep learning has been used for power load forecasting [45] and traffic forecasting [184].

More recently, [188] employed a periodic long short-term memory (LSTM) network for parking behavior prediction.

[88] used a U-shaped convolutional neural network (CNN) to track solar magnetic flux elements. In this research, we propose a new deep learning method for reconstructing total solar irradiance.

Solar irradiance is the primary source of energy for our Earth [102], and is a key input for climate models and changes [17]. It is described in terms of total solar irradiance (TSI) when all of the radiation is measured. Irradiance is defined as the amount of light energy from an object that is hitting a square meter of another object each second. Solar irradiance is the amount of light energy from the Sun's entire disk measured at the Earth, and it is known to vary over different temporal scales, in a manner that is strongly wavelength dependent [100]. TSI variability affects the Earth's atmosphere and climate in many ways [69]. To understand the effect of solar radiation on our Earth's climate changes, solar irradiance records for long periods of time are required. Since systematic measurements of irradiance started only in

the late seventies, many models were introduced to provide irradiance records dating back to times ranging from century to millennia. All such models are based on the empirical evidence that irradiance variability is modulated by surface magnetism [58], while the approaches adopted in the different models are mostly driven by the type of proxies of the magnetic field available at the temporal scales considered.

Most of the published models aim to reconstruct irradiance variability up to a few centuries into the past [107, 103]. Such models are intended to address the impact of solar variability on Earth's increase of temperatures registered from the pre-industrial era, and mostly make use of sunspot, or sunspot-group number, as a proxy of the surface magnetic activity. A few models have instead been proposed in the literature aiming at reconstructing irradiance variations at longer temporal scales. Because at those times telescopic observations were not available, such reconstructions necessarily make use of indirect proxies. These mostly consist of radioisotopes like ^{14}C, ^{10}Be and nitrate-related species [166], which are generated by the interaction of energetic particles with the Earth's atmosphere, whose flux, in turn, is regulated by the heliospheric magnetic field.

Some of the historical irradiance reconstruction models used linear regression relationships between the irradiance measured at modern times and input proxies. More complex techniques make use of geomagnetic models to estimate from radioisotopes the *open* and *closed* components of the solar magnetic field, from which the distribution of magnetic features over the Sun's disk is recovered.

The most recent state-of-the-art model of this kind was developed by [179] , who reconstructed TSI for the previous 9 millennia, making use of two different cosmogenic isotopes, ^{10}Be and ^{14}C, derived from various datasets [180]. All of the published models reconstructed solar irradiance based on physics properties [179].

In this research, we present the first deep learning model, called TSInet, to reconstruct total solar irradiance for more than 9,000 years. Our main contributions are outlined as follows:

1. We use TSInet to reconstruct TSI for the entire 9,000 years already covered in the recent reconstruction by [179] and for additional 1,000 years when physical data are not available.

2. Our deep learning model does not rely on proxies; hence our model is not affected by uncertainties in the proxies including errors in their measurements and estimates.

3. Our TSInet method can be extended back at times when proxies are not available. When physical data are available, TSInet agrees well with the state-of-the-art physics-based reconstruction models on the available data.

5.2 Data

In this work, we use measurements of the TSI provided by the Total Irradiance Monitor aboard the SOlar Radiation and Climate Experiment (SORCE)[146][1]. This dataset, used as our training set, contains daily TSI measurements carried out from 2003 to the present. Figure 5.1 illustrates the SORCE time series dataset showing the total solar irradiance over time.

Our testing set contains measurements from TCTE Total Solar Irradiance daily averages[2]. Total Solar Irradiance Calibration Transfer Experiment (TCTE)[3] measurements are made by the LASP TCTE Total Irradiance Monitor (TIM) instrument aboard the U.S. Air Force's STPSat-3 spacecraft. This TIM has been measuring total solar irradiance since late 2013.

[1]Retrieved on 04/10/2021 from `http://lasp.colorado.edu/home/sorce/data/tsi-data/`
[2]Retrieved on 04/10/2021 from `http://lasp.colorado.edu/lisird/data/tcte_tsi_24hr/`
[3]Retrieved on 04/10/2021 from `http://lasp.colorado.edu/home/missions-projects/quick-facts-tcte/`

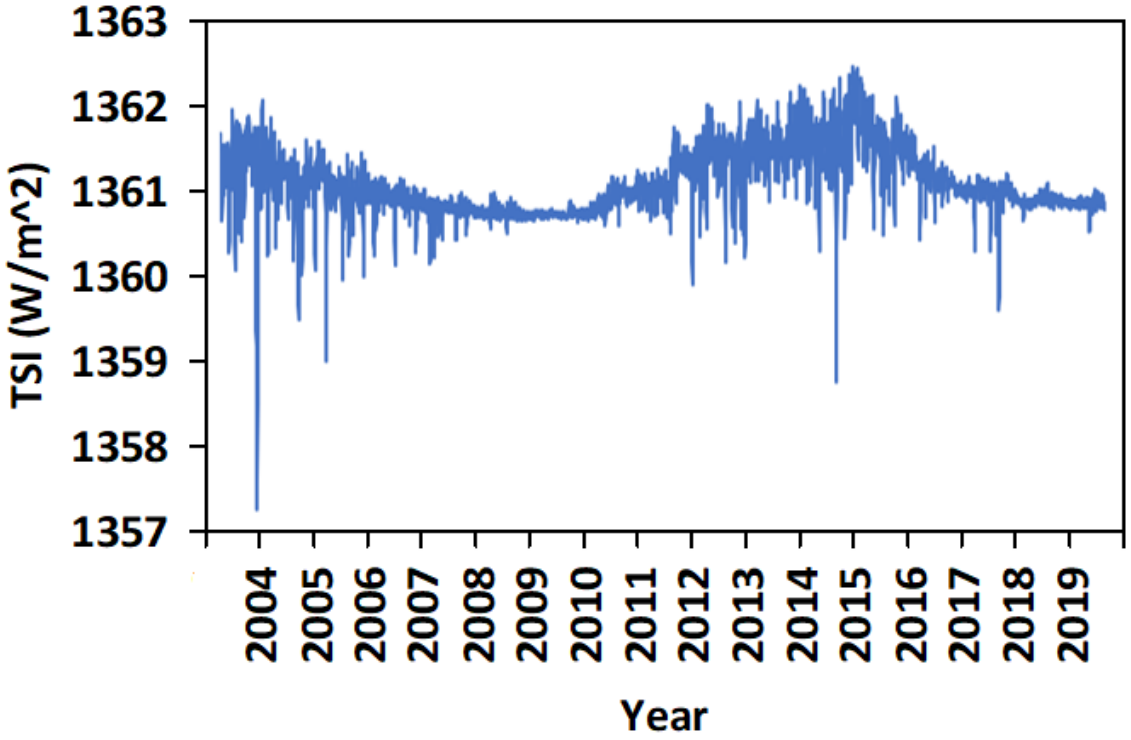


Figure 5.1 SORCE total solar irradiance (TSI) data from 2003 to the present.

In addition, we adopt the following publicly available datasets obtained, over different temporal ranges, by different physics-based models, which will be used as testing sets in our work.

- **NRLTSI2 Daily Averages**[4] is the daily climate record of total solar irradiance from 1882 to the present. It is constructed using version 2 of the Naval Research Laboratory's (NRL) solar variability model (NRLTSI2). The NRLTSI2 model computes TSI based on the changes of the quiet Sun conditions arising from bright faculae and dark sunspots on the solar disk. It uses linear regression between solar sunspots and facular features, as well as irradiance observations from SORCE.

- **SATIRE-S** (Spectral And Total Irradiance REconstruction model - Space era)[5] provides daily reconstruction of solar irradiance in the period of 1974 – 2013. Irradiance is reconstructed by combining the area coverage of magnetic and quiet features as derived by full-disk magnetograms and continuum images of

[4]Retrieved on 04/10/2021 from `http://lasp.colorado.edu/lisird/`
[5]Retrieved on 04/10/2021 from `http://www2.mps.mpg.de/projects/sun-climate/data.html`

the Sun, together with spectral syntheses obtained by one-dimensional, static, atmosphere models [183].

- **SATIRE-M** (Spectral And Total Irradiance REconstruction model - Millennia)[6] is similar to SATIRE-S, but the area coverage of magnetic structures is estimated by making use of a model which relies on the sunspot number [180]. This model provides decennial averages and reconstructs the solar irradiance over the last 9,000 years. The model is used to reconstruct decadal total TSI.

The total solar irradiance values range from 1356.656 to 1363.525. We use a feature scaling technique, also known as data normalization, to normalize the range of data to increase the cohesion of the TSI values. Specifically, we use the min-max normalization that is calculated as follows:

$$\hat{v_i} = \frac{v_i - \min(S)}{\max(S) - \min(S)} \tag{5.1}$$

where $\hat{v_i}$ (v_i, respectively) is the normalized value (actual value, respectively) at time point i, and S represents the input data set.

The normalized TSI values range from 0 to 1.

5.3 Proposed Method

5.3.1 Architecture and Training of TSInet

Figure 5.2 presents the architecture of our TSInet. Let t be the latest time point. Data sample x_t contains w values $v_t, v_{t-1}, \ldots, v_{t-w+1}$ and the label v_{t-w} where v_t is the TSI value at time point t. (In the study presented here, the time window, w, is set to 7.) We train TSInet with multiple batches. In the first batch, we use the n training data samples, $x_t, x_{t-1}, \ldots, x_{t-n+1}$, to train TSInet. (In the study presented here, the number of input data samples, n, is set to 10.) The label of the n training data samples is determined by the label of the last data sample (*i.e.*, x_{t-n+1}). In the second batch, we use the next n training data samples, $x_{t-n}, x_{t-n-1}, \ldots, x_{t-2n+1}$,

[6]Retrieved on 04/10/2021 from `http://www2.mps.mpg.de/projects/sun-climate/data.html`

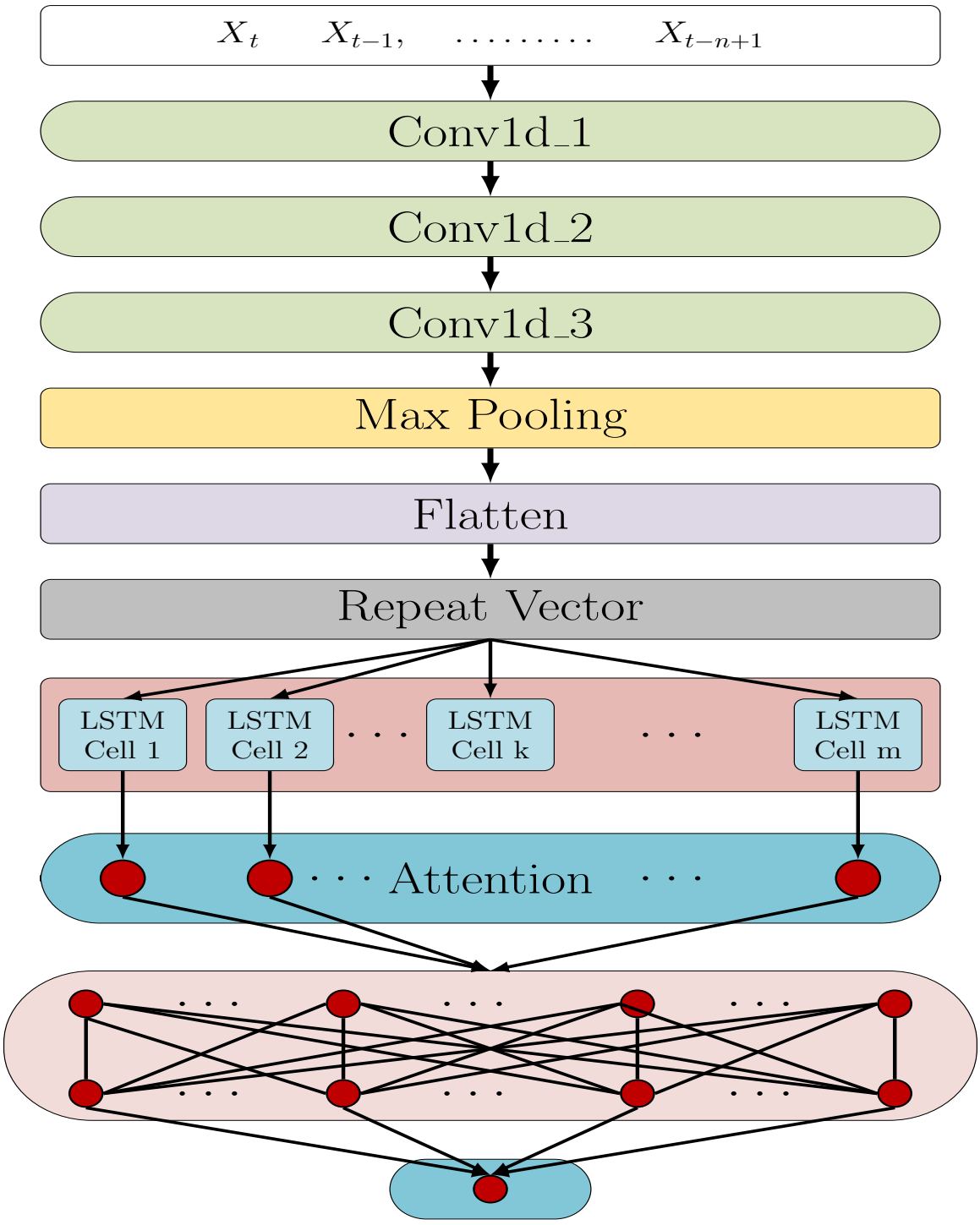


Figure 5.2 Architecture of TSInet.

to train TSInet. The label of the n training data samples is determined by the label of x_{t-2n+1}. In the third batch, we use the following n training data samples, $x_{t-2n}, x_{t-2n-1}, \ldots, x_{t-3n+1}$, to train TSInet. The label of the n training data samples

is determined by the label of x_{t-3n+1}. We continue this training process until all TSI values in the training set are used. For every two adjacent data samples x_i, x_{i-1}, they overlap on $w-1$ TSI values, namely $v_{i-1}, v_{i-2}, \ldots, v_{i-w+1}$.

TSInet consists of three convolutional layers (Conv1d_1, Conv1d_2, and Conv1d_3) where the kernel slides along 1 dimension on the time series, a max pooling layer, a flatten layer, a repeat vector layer, an LSTM (long short-term memory) layer, an attention layer, two fully connected layers, and an output layer. The output from the three convolutional layers is flattened by the flatten layer and transformed into a sequence, also known as a feature vector. The repeat vector layer repeats the feature vector to reshape and prepare it as the input to the LSTM layer. The LSTM layer in our architecture contains m LSTM cells (in this study, m is set to 10). The attention layer with m neurons is used to focus on the relevant information in each time step [14]. Each of the two fully connected layers has 200 neurons. The activation function used in our model is ReLU (rectified linear unit). TSInet produces as output a predicted TSI value.

The proposed TSInet is implemented in Python, Keras, and Tensorflow. We use adaptive moment estimation (Adam) [108, 67] as the network optimizer, which is a stochastic gradient descent algorithm that can update network weights based on training data. Adam is configured with a learning rate of 0.003 and a weight decay of 0.000005 to regularize the weights and minimize the test error during training in each epoch. Other Adam parameters (β_1, β_2, respectively) are set to default values (0.9, 0.999, respectively). To achieve faster back-propagation convergence, we adopt the mini-batch strategy described in [108, 67]. The number of epochs is set to 10 by default.

5.3.2 Reconstruction of Total Solar Irradiance

After describing the architecture and training procedure of TSInet, we now turn to the algorithms for reconstructing total solar irradiance (TSI) in a testing set. We develop two reconstruction algorithms: (i) single-step or 1-step reconstruction; (ii) multi-step or k-step, $k > 1$, reconstruction.

Let t be the latest time point. With single-step reconstruction, we begin by considering the n testing data samples $x_t, x_{t-1}, \ldots, x_{t-n+1}$ where x_t contains the $w + 1$ TSI values $v_t, v_{t-1}, \ldots, v_{t-w+1}, v_{t-w}$ in the testing set. Our TSInet model, which is trained as described in Section "Architecture and Training of TSInet," takes as input the n testing data samples and predicts the label of the last testing data sample (*i.e.*, x_{t-n+1}), which is treated as the label of the n testing data samples. We then use the n testing data samples together with the predicted label to re-fit or re-train TSInet. The re-trained TSInet then takes as input the next n testing data samples $x_{t-1}, x_{t-2}, \ldots, x_{t-n}$ and predicts the label of the last testing data sample (*i.e.*, x_{t-n}), which is treated as the label of the n testing data samples. We again use the n testing data samples together with the predicted label to re-fit or re-train TSInet. The re-trained TSInet then takes as input the following n testing data samples $x_{t-2}, x_{t-3}, \ldots, x_{t-n-1}$ and predicts the label of the last testing data sample (*i.e.*, x_{t-n-1}), which is treated as the label of the n testing data samples. We then use the n testing data samples together with the predicted label to re-train TSInet. We continue this predicting-retraining process until all TSI values (labels) in the testing set have been predicted, at which point we have reconstructed all the TSI values in the testing set.

With multi-step reconstruction, we begin by considering the first batch containing the n testing data samples $x_t, x_{t-1}, \ldots, x_{t-n+1}$ in the testing set. Our trained TSInet takes as input these n testing data samples and predicts the label of the last testing data sample (*i.e.*, x_{t-n+1}), which is treated as the label of the n

testing data samples. Then, the same TSInet model takes as input the second batch containing the next n testing data samples $x_{t-1}, x_{t-2}, \ldots, x_{t-n}$ and predicts the label of the last testing data sample (*i.e.*, x_{t-n}), which is treated as the label of the n testing data samples. We keep on using the same TSInet model until the model takes as input the kth batch containing the n testing data samples $x_{t-k+1}, x_{t-k}, \ldots, x_{t-n-k+2}$ and predicts the label of the last testing data sample (*i.e.*, $x_{t-n-k+2}$), which is treated as the label of the n testing data samples. We then use the k batches, where each batch contains n testing data samples together with their predicted label, to retrain our TSInet as shown in Figure 5.2. Then we use the re-trained model to predict the labels for the next $k \times n$ testing data samples.

The difference between single-step reconstruction and multi-step reconstruction is that the former retrains TSInet once using one batch containing n testing data sample in every one step while the latter retrains TSInet once using k batches containing $k \times n$ testing data samples in every k steps.

5.4 Experiments and Results

5.4.1 Performance Metrics

We conducted a series of experiments to evaluate the performance of the proposed TSInet and compare it with related methods. The performance metrics used here are the root mean square error (RMSE) and Pearson correlation coefficient (CORR). RMSE is calculated as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{5.2}$$

where $\hat{y}_i$ (y_i, respectively) represents the predicted TSI value (actual TSI value, respectively) at time point i. RMSE measures the differences between the actual TSI values and the predicted TSI values by a method. The lower the RMSE value,

the more accurate the method is. CORR is calculated as follows:

$$\text{CORR} = \frac{\sum_i (\hat{y}_i - \mu(\hat{y}))(y_i - \mu(y))}{\sqrt{\sum_i (\hat{y}_i - \mu(\hat{y}))^2}\sqrt{\sum_i (y_i - \mu(y))^2}} \tag{5.3}$$

where $\mu(\hat{y})$ denotes the mean of all predicted TSI values and $\mu(y)$ denotes the mean of all actual TSI values. CORR is used to measure how strong the relationship between the predicted and actual TSI values is. CORR ranges from 1 to -1, where 1 means there is a strong positive correlation, -1 means there is a strong negative correlation, and 0 means there is no correlation between the predicted and actual TSI values.

5.4.2 Single-Step vs. Multi-Step Reconstruction Algorithms

In this experiment, we compared the single-step (*i.e.*, 1-step) and multi-step (*i.e.*, k-step) reconstruction algorithms

described in Section "Reconstruction of Total Solar Irradiance." We used the SORCE training set to train TSInet and reconstructed the TSI values in the TCTE testing set for varying k, $k = 1, \ldots, 10$. For each k, we computed the performance metrics and recorded the runtime used by the algorithms. Figure 5.3 shows the performance metrics, RMSE and CORR, for varying k. it can be seen from the figure that the performance of TSInet degrades as k increases. This happens because the TSInet model is refitted more often, and hence is more accurate when k is smaller. On the other hand, smaller k requires more runtime, as shown in Figure 5.3. In subsequent experiments, we fixed $k = 5$ as it achieved good performance while requiring reasonable runtime.

5.4.3 Ablation Tests

In this experiment, we performed ablation tests to analyze and evaluate the components of our TSInet framework by considering two models based on TSInet: CNN and LSTM. The CNN model is a subnet of TSInet, keeping the three

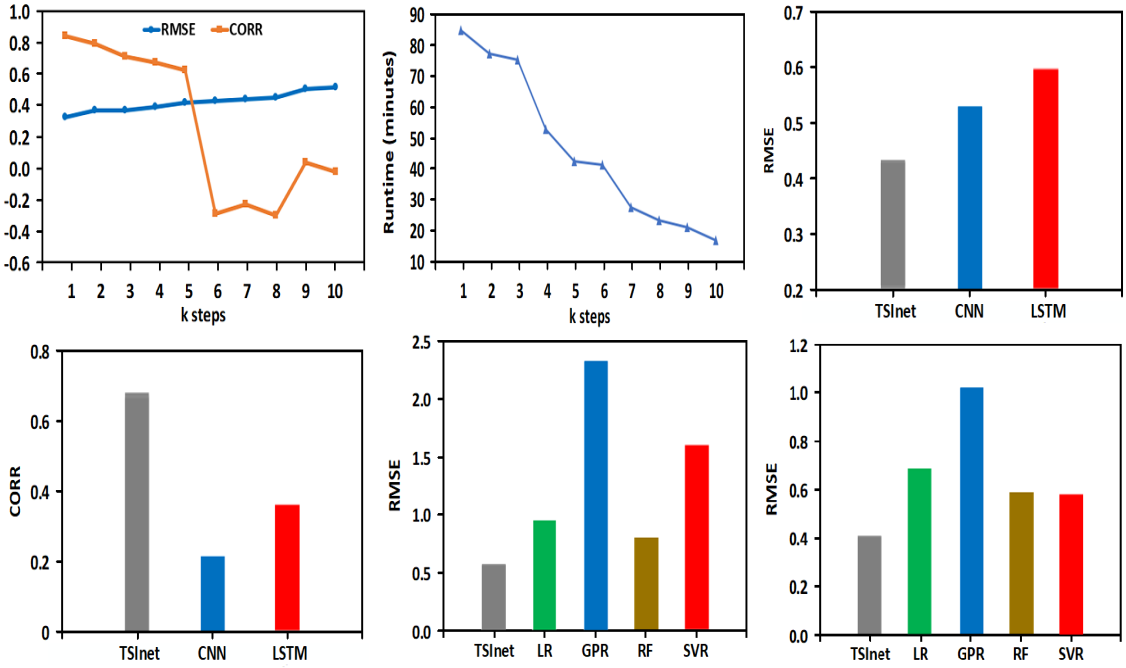


Figure 5.3 (i) RMSE and CORR values obtained by TSInet for k-step, $k = 1, 2, \ldots, 10$, reconstruction of TSI on the TCTE dataset. (ii) Runtime needed by TSInet for k-step, $k = 1, 2, \ldots, 10$, reconstruction of TSI on the TCTE dataset. (iii) Comparison of RMSE values of TSInet, CNN and LSTM on the TCTE dataset. (iv) Comparison of CORR values of TSInet, CNN and LSTM on the TCTE dataset. (v) Comparison of RMSE values of five TSI reconstruction methods on the TCTE dataset. (vi) Comparison of RMSE values of five TSI reconstruction methods on the NRLTSI2 dataset. Results from the SATIRE-S dataset are similar and omitted.

convolutional layers, max pooling layer, flatten layer, repeat vector layer, attention layer, two fully connected layers, and output layer, but removing the LSTM layer. The LSTM model is also a subnet of TSInet, keeping the LSTM layer, attention layer, two fully connected layers, and output layer, but removing the three convolutional layers, max pooling layer, flatten layer, and repeat vector layer.

Figure 5.3 presents the RMSE and CORR results from TSInet, CNN, and LSTM. It can be seen from the figures that TSInet yields the best accuracy and correlation among the three methods. This happens because CNN learns characteristics from the input data but it lacks temporal components to deeply analyze the time series information in the data. On the other hand, LSTM captures the

temporal correlation in the input data, but it works on the raw input data without learning additional characteristics to strengthen the correlation between the data entries. TSInet combines the characteristics it learns in the CNN network and temporal correlation it learns in the LSTM network. Therefore, TSInet achieves the best performance.

5.4.4 Comparison with Related Methods

In this experiment, we compared TSInet with four closely related machine learning algorithms including linear regression (LR), Gaussian process regression (GPR) [151, 8], random forest (RF), and support vector regression (SVR). Figures 5.3 presents the RMSE results for the five methods on the TCTE and NRLTSI2 datasets. The figures show that TSInet achieves the best performance among the five methods in terms of RMSE. The CORR results are similar and omitted here.

To assess whether the results obtained by our TSInet agree with entries in the testing datasets, we performed the Wilcoxon signed-rank test [136]. According to the test, the difference between the TSInet results and entries in TCTE (NRLTSI2, SATIRE-S, respectively) is not statistically significant with $p = 0.01552 < 0.05$ ($0.0001 < 0.05$, $0.0001 < 0.05$, respectively).

5.4.5 Reconstruction of TSI on the SATIRE-M Dataset

SATIRE-M contains decennial averages and is comprised of solar irradiance over the last 9,000 years. Each entry in the SATIRE-M dataset represents an average of 10 years. However, our TSInet reconstructs daily TSI. To reconstruct solar irradiance on the SATIRE-M dataset, we employ the following technique. Recall that the SATIRE-S dataset provides daily reconstruction of solar irradiance in the period of 1974 – 2013. We first use TSInet to reconstruct total solar irradiance beyond 1974 on SATIRE-S. Then we take 10-year averages on the reconstructed TSI values.

Figure 5.4 compares the 10-year averages obtained by TSInet with the entries in SATIRE-M. TSInet's results agree mostly with entries in SATIRE-M. According to the Wilcoxon signed-rank test [136], the difference between TSInet's results and SATIRE-M entries is not statistically significant ($p = 0.000297 < 0.05$). Figure 5.4 also shows that our TSInet model is capable of reconstructing total solar irradiance beyond 9,000 years. We reconstructed total solar irradiance for additional 1,000 years beyond the SATIRE-M data.

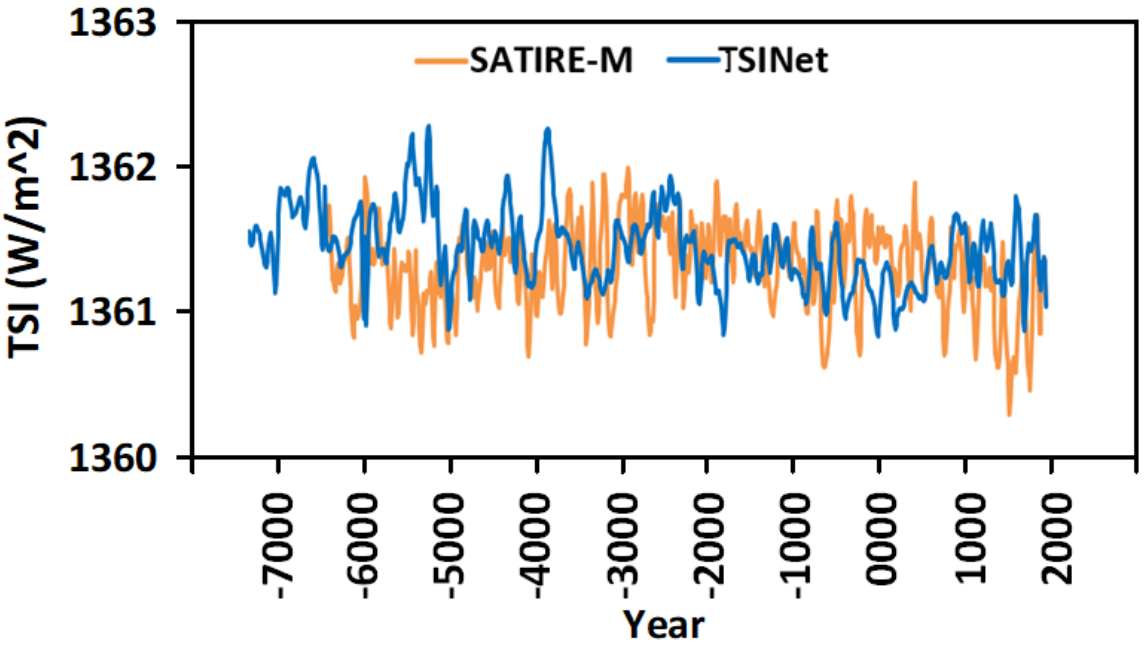


Figure 5.4 SATIRE-M reconstruction using TSInet plus reconstruction of solar irradiance for additional1,000 years.

5.5 Summary

The Earth's primary source of energy is the radiant energy from the Sun. This energy is known as solar irradiance, or total solar irradiance (TSI) when all of the radiation is measured. The changes in solar irradiance have a significant impact on Earths' atmosphere and climate. Therefore, studying and reconstructing solar irradiance is crucial in solar physics. Existing methods for solar irradiance reconstruction

are all based on physics-based models [179]. In this research, we presented the first deep learning method (TSInet) for reconstructing total solar irradiance (TSI). Experimental results showed that results from our TSInet agree well with those from the physics-based models. When compared to closely related machine learning methods, TSInet achieves the best performance among the methods. TSInet does not depend on physics properties such as proxies, and hence it can be extended back at times when proxies were not available. We demonstrated here that TSInet is able to reconstruct TSI for more than 9 millennia.

CHAPTER 6

SOFTWARE AND DATABASES

This chapter presents the implementation of some of deep learning tools using Jupyter notebooks and with Github for the all presented models and frames in the dissertation including TSInet, DeepSun MLaaS, SEP biLSTM, DSTT, and KpNet. The tools are Binder enabled and have Zenodo archive to download.

6.1 Community Coordinated Software Center

6.1.1 MLaaS: FlareML Framework

This section presents the FlareML framework. FlareML is the backend of an the machine-learning-as-a-service (MLaaS) presented in Section 2.1. Same as MLaaS, FlareML system employs four machine learning methods: (i) ensemble (ENS), (ii) random forests (RF), (iii) multilayer perceptrons (MLP), and (iv) extreme learning machines (ELM). FlareML is implemented in a Jupyter notebook using Python programming language and its related Tensorflow packages. In addition, it can run in Binder and has Zenodo enabled archive. Specific requirements to run the tool are detailed with examples in the GitHub repository. The repository includes the links to run the Jupyter notebook, launch a Binder docker, and download Zenodo. Figure 6.1 shows the FlareML GitHub repository main page that includes all the source code required to download and run the tool.

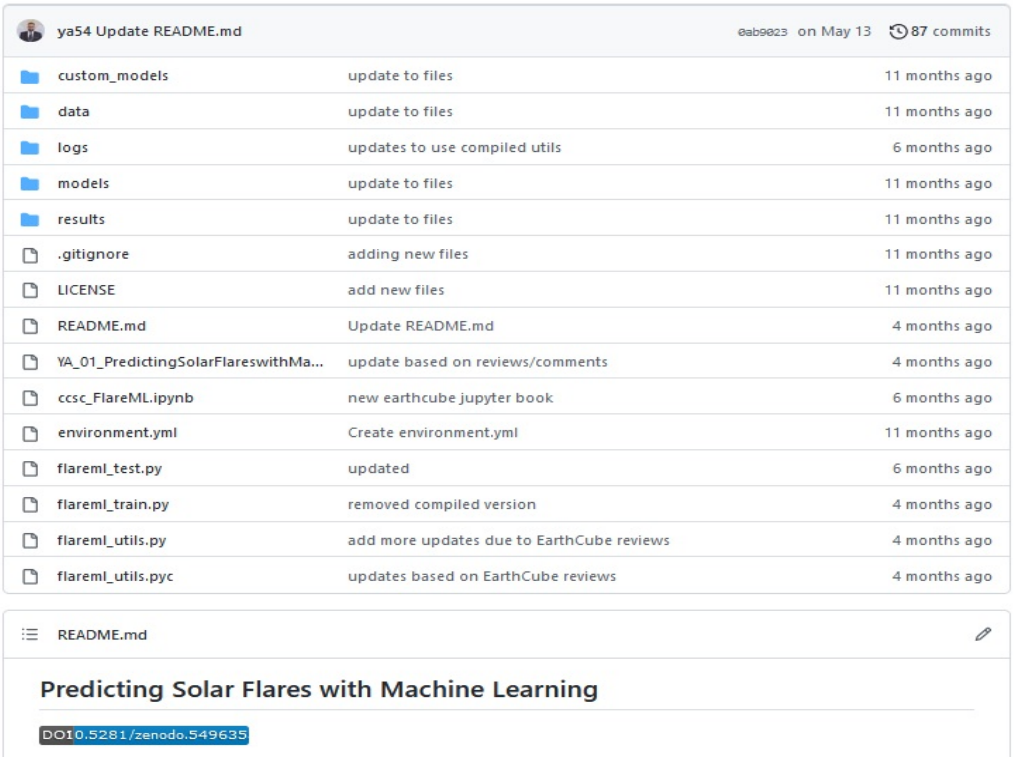


Figure 6.1 FlareML Github repository

6.1.2 BiLSTM: SEP Prediction Framework

This section presents the SEP BiLSTM framework. BiLSTM demonstrates how to predict SEP using deep learning (DL) and SDO/HMI vector magnetic data products (SHARP parameters) as specified in Chapter 3. BiLSTM is implemented in a Jupyter notebook using Python programming language and its related Tensorflow packages, it can run in Binder, and has Zenodo enabled archive. Specific requirements to run the tool are detailed with examples in the GitHub repository. The repository includes the links to run the Jupyter notebook, launch a Binder docker, and download Zenodo. Figure 6.2 shows the BiLSTM GitHub repository main page that includes all the source code required to download and run the tool.

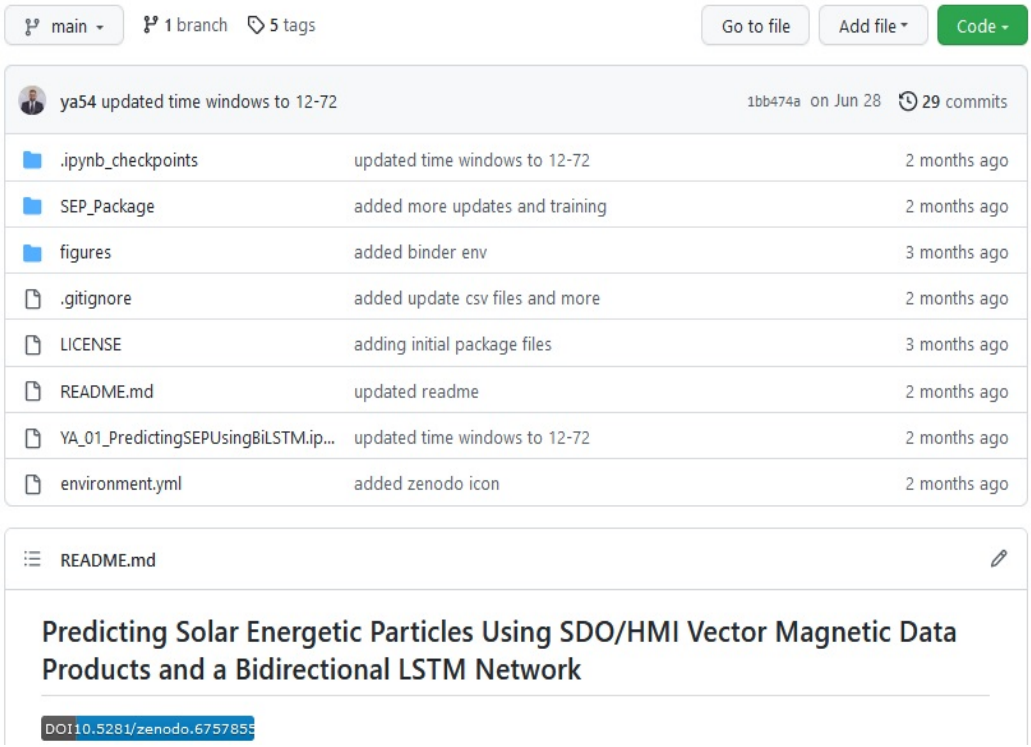


Figure 6.2 BiLSTM Github repository

6.1.3 DSTT Framework

This section presents the DSTT framework. DSTT to demonstrate how to forecast Dst index using deep learning (DL) and solar wind parameters and provide uncertainty quantification with Bayesian network as detailed in Chapter 4. DSTT is implemented in a Jupyter notebook using Python programming language and its related Tensorflow packages, it can run in Binder, and has Zenodo enabled archive. Specific requirements to run the tool are detailed with examples in the GitHub repository. The repository includes the links to run the Jupyter notebook, launch a Binder docker, and download Zenodo. Figure 6.3 shows the DSTT GitHub repository main page that includes all the source code required to download and run the tool.

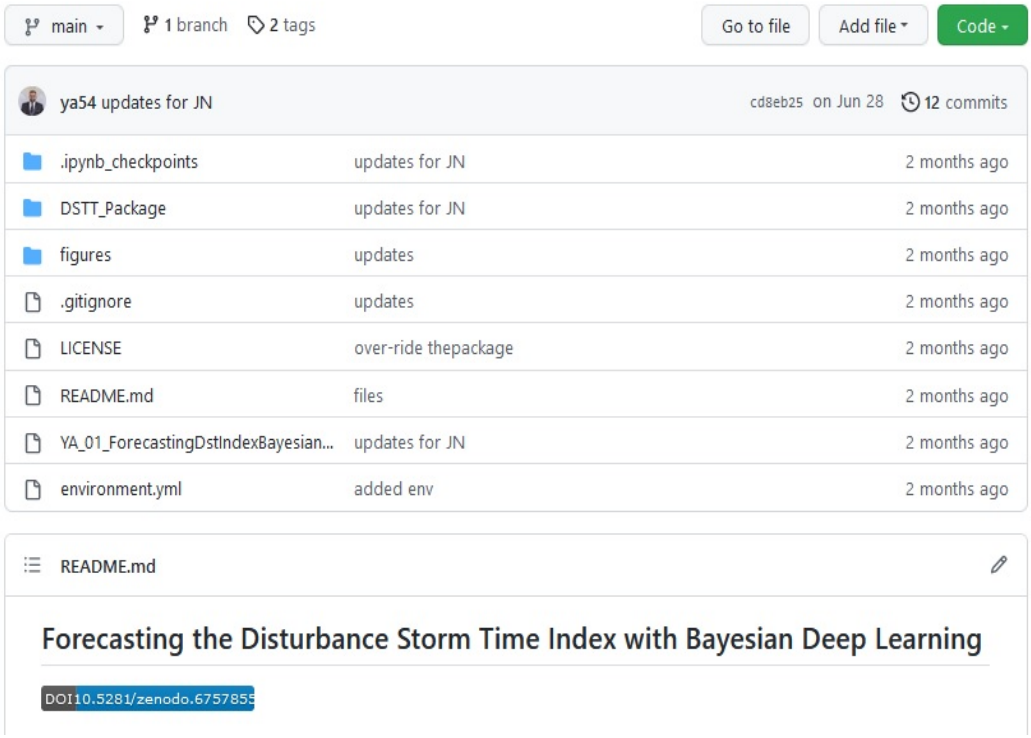


Figure 6.3 DSTT Github repository

6.1.4 KpNet Framework

This section presents the KpNet framework. KpNet to demonstrate how to forecast Kp index using deep learning (DL) and solar wind parameters and provide uncertainty quantification with Bayesian network as described in Chapter 4. KpNet is implemented in a Jupyter notebook using Python programming language and its related Tensorflow packages, it can run in Binder, and has Zenodo enabled archive. Specific requirements to run the tool are detailed with examples in the GitHub repository. The repository includes the links to run the Jupyter notebook, launch a Binder docker, and download Zenodo. Figure 6.4 shows the GitHub KpNet repository main page that includes all the source code required to download and run the tool.

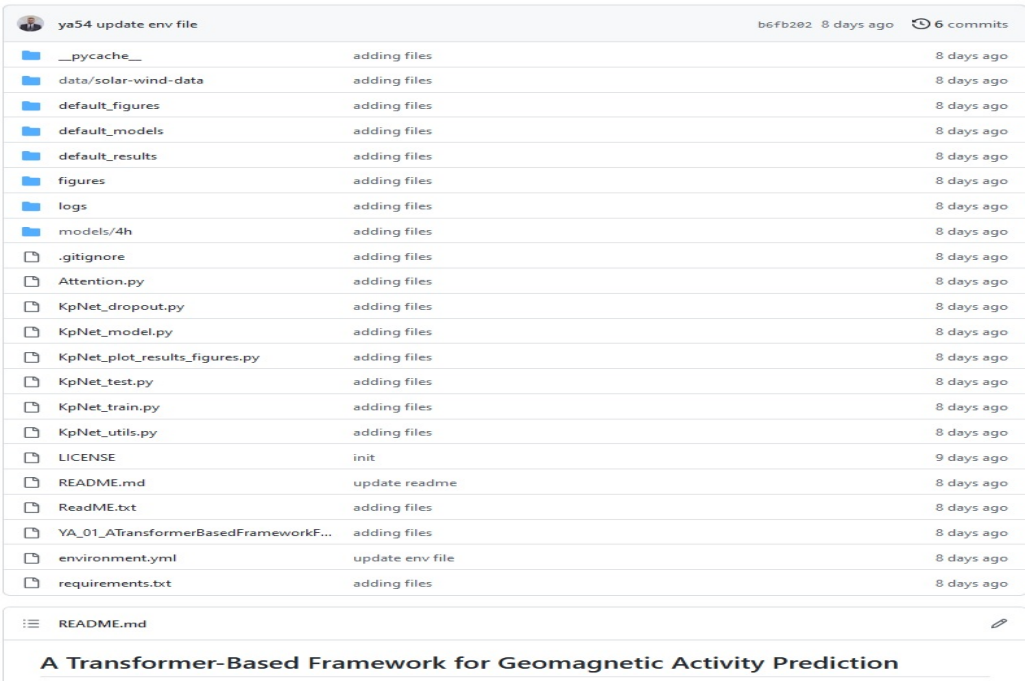


Figure 6.4 KpNet Github repository

6.1.5 TSInet Framework

This section describes and presents the TSInet framework model. TSInet is for reconstructing total solar irradiance. The tool is implemented in a Jupyter notebook using Python programming language and its related Tensorflow packages. In addition, The tool can run in Binder. Figure 6.5 shows the TSInet GitHub repository main page that includes all the source code required to download and run the tool. The repository includes the links to run the Jupyter notebook, launch a Binder docker, and download Zenodo.

6.2 Cyberinfrastructure Database

The Cyberinfrastructure contains solar images of significant M- and X-class flares occurring in solar cycle 24. These solar images, obtained from NASA's Solar Dynamics Observatory, allow scientists to observe and understand the precursors and aftermath of each flare stored in SolarDB. The web-based online cyberinfrastructure also contains digitized solar images of some historic observations from the Big

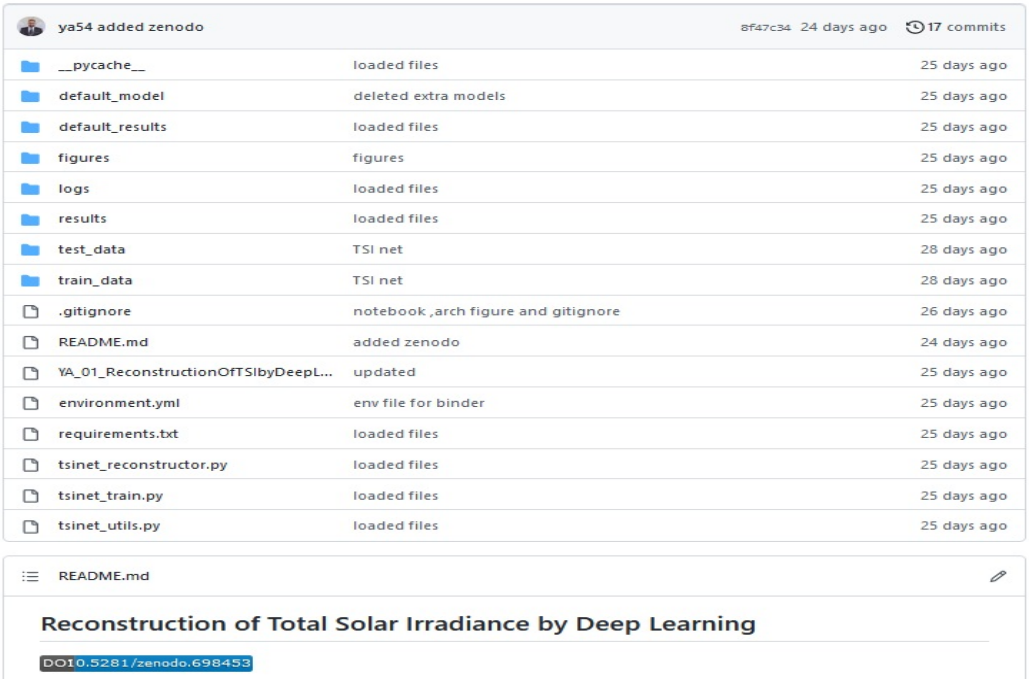


Figure 6.5 TSInet Github repository

Bear Solar Observatory (BBSO) and H-alpha observations collected from eight observatories worldwide.

6.2.1 SolarDB

This repository is composed of three databases, namely a flare database, a global Hα database and a Big Bear Solar Observatory archive, as well as our operational near real-time flare forecasting system. The figure below shows the home page of the cyberinfrastructure. Figure 6.6 shows the landing page of the Cyberinfrastructure website.

6.2.2 Flare Database

The Flare Database contains 129 significant flares of class M5.0 or larger from 60 active regions. It provides several filters to facilitate searching the database. It also provides options to allow the user to watch a quick look movie for a selected flare

Figure 6.6 Cyberinfrastructure: SolarDB database website main page.

event or download data files related to the flare event. Figure 6.7 shows a search example in the database.

6.2.3 Global H-alpha Network Database

This database site consists of multiple links to show H-alpha artifacts as follows:

1. **Latest images** include a list of latest images from the following observatories: Big Bear solar observatory, Observatório de Coimbra, Observatory de Paris Meudon, Uccle Solar Equatorial Table, Kanzelhöhe Solar Observatory, Catania Astrophysical Observatory, Observatoire Pic Du Midi, and Huairou Solar Observatory.

2. **Synoptic images** include a list of the most recent and historical Synoptic map of Carrington rotation numbers.

3. **Stations** list the most update information about the each observatory station including physical location address, contact numbers, website address, and geographical coordinates.

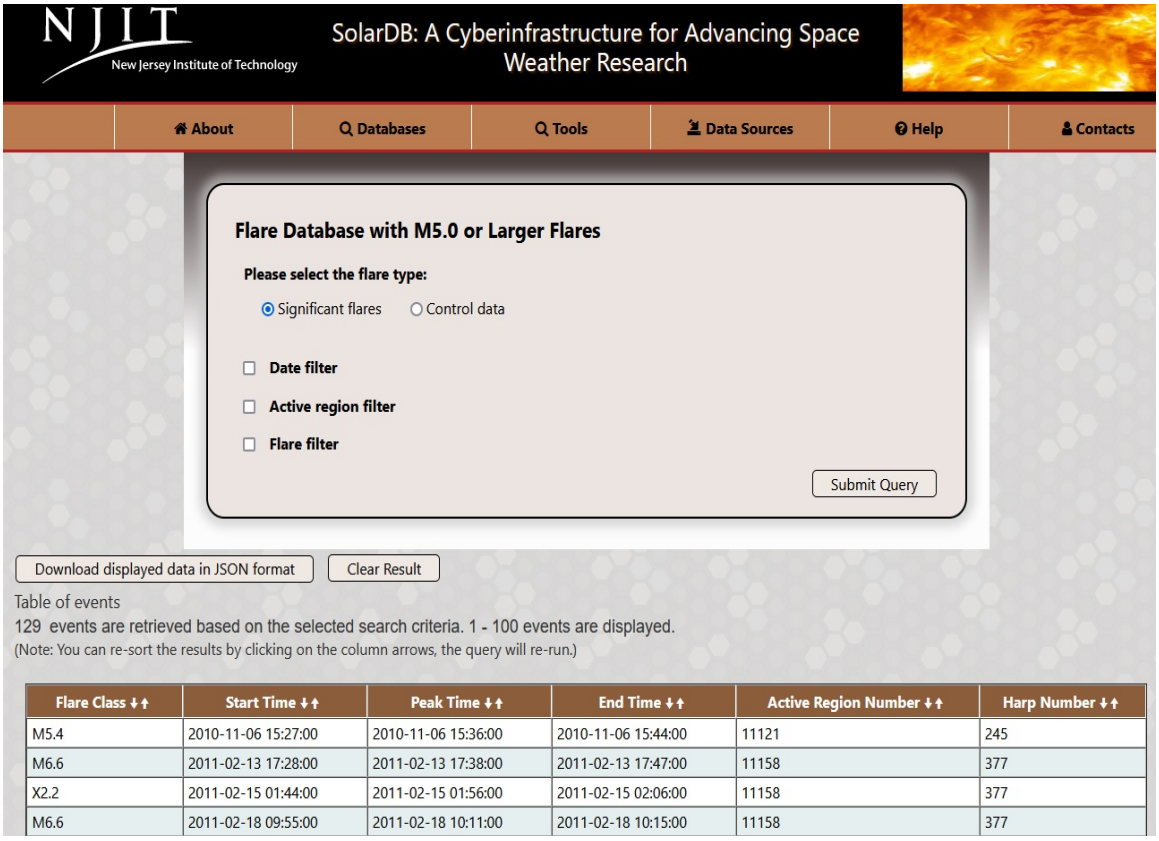


Figure 6.7 Cyberinfrastructure: Flare Database.

4. **Instruments** list the instruments information used by the observatories including Telescope Aperture, Filter Bandpass, Tunable Filter Range, Detector, Detector Size, and Dynamic Range.

5. **Daily images** includes historical daily images from the observatories.

   Figure 6.8 shows an example of the daily images from all observatories.

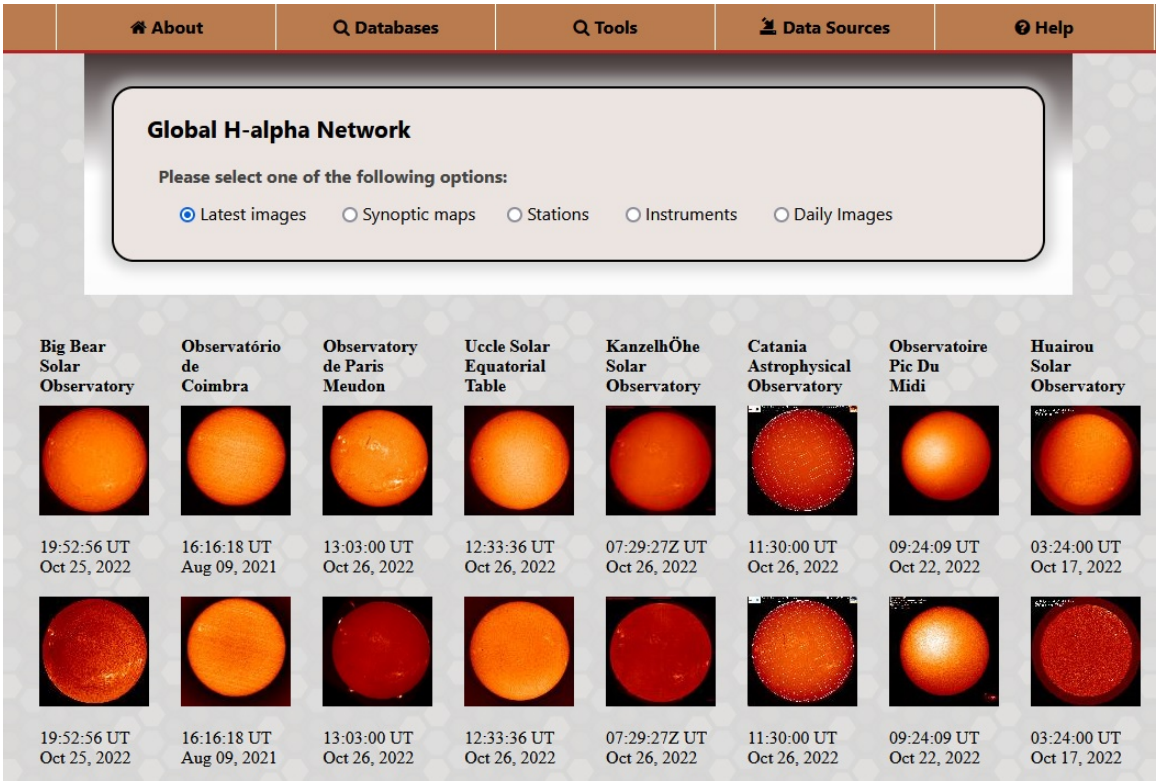


Figure 6.8 Cyberinfrastructure: Global H-alpha Network Database.

6.2.4 Big Bear Solar Observatory (Historic Observations)

This database contains 1147 records of archived active regions from the Big Bear solar observatory. It provides several filters to facilitate searching the database. It also provides options to allow the user to watch a quick look movie for a selected active region or download data files related to the active region. Figure 6.9 shows a search example in the database.

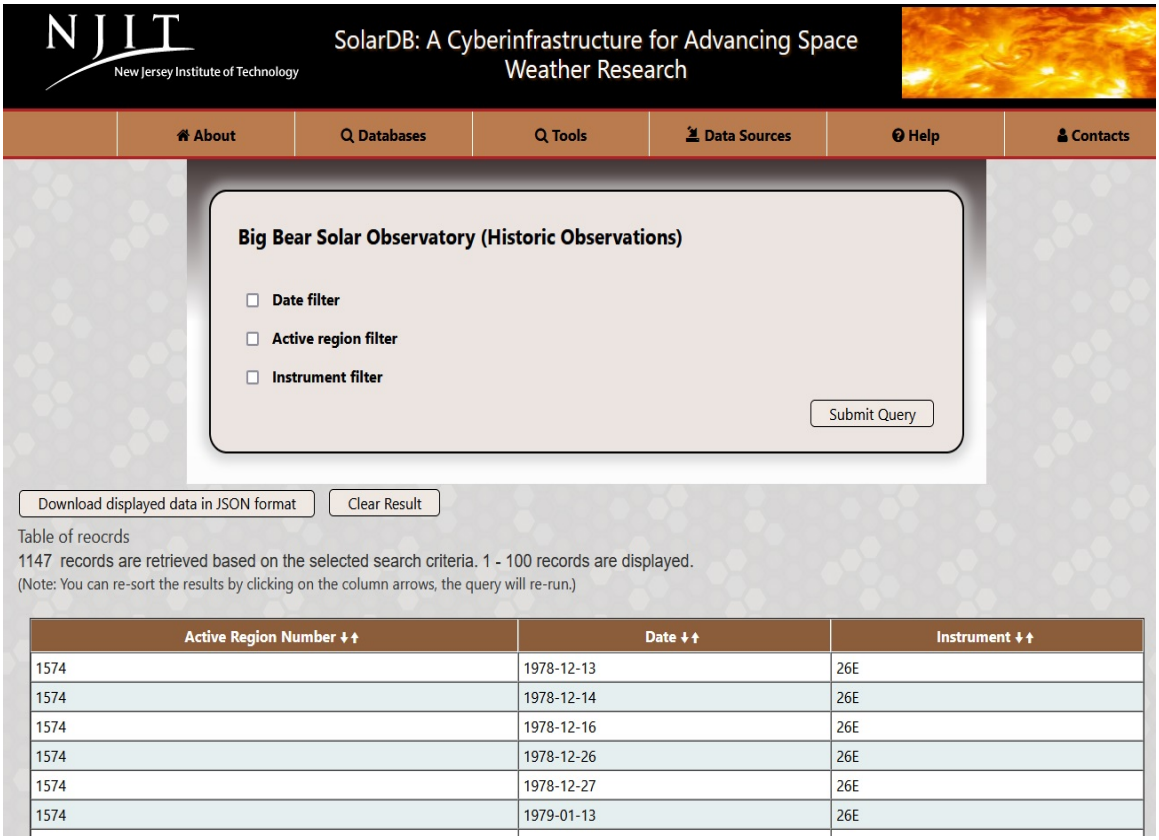


Figure 6.9 Cyberinfrastructure: Big Bear Solar Observatory (Historic Observations).

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

This dissertation addresses multiple crucial problems in space weather and space climate domains. Basically, in space weather, it tackles flares prediction, solar energetic particles and geomagnetic activities: disturbance storm time (Dst) index and the planetary index (Kp). In space climate, it addresses the total solar irradiance construction. In addition, the dissertation provides software development tools that can be used in space weather and space climate research and operational systems.

For space weather, the dissertation addresses the following problems. First, the dissertation presents the first web based machine-learning-as-a-service (MLaaS) framework, called DeepSun, that is capable of predicting solar flares through the internet as well as a hybrid-transformer algorithm that forecasts the occurrences of solar flares within the next 24 hours. Second, the dissertation presents a deep learning method, specifically a bidirectional long short-term memory (biLSTM) network, to predict if an active region (AR) would produce an SEP event given that (i) the AR will produce an M- or X-class flare and a coronal mass ejection (CME) associated with the flare, or (ii) the AR will produce an M- or X-class flare regardless of whether or not the flare is associated with a CME. Third, the dissertation presents multiple algorithms to forecast geomagnetic indices. Geomagnetic indices are used by Geospace scientists to measure space storms and their activities. The first index is the SYM-H index and the algorithm used is graph neural network combined with bidirectional long short-term memory. The second index is the interplanetary index, known as the Kp index using transformer based model. The third index is the disturbance storm time (Dst) using mult-head attention layer combined with convolutional neural network and long short-term memory algorithms. The algorithms have an added add-on

functionality that integrates the Bayesian inference into the learning framework, that is able to quantify both aleatoric (data) uncertainty and epistemic (model) uncertainty when predicting future indices.

For space climate, the dissertation presents the first deep learning method, called TSInet, to reconstruct total solar irradiance (TSI). The TSInet reconstructs total solar irradiance by deep learning for short and long periods of time that span beyond the physical models' data availability. It can be used as an add-on to physical models to reconstruct TSI for more than 9,000 years.

In addition, the dissertation presents the implementation of the deep learning tools are developed for the algorithms presented in the dissertation. The tools are implemented using Jupyter notebooks with Github and publicly available to download and use. The tools include the TSInet for reconstructing total solar irradiance, DeepSun MLaaS for predicting solar flares, SEP biLSTM for predicting solar energetic particles, DSTT for predicting and forecasting disturbance storm time, and KpNet for forecasting the Kp index. The tools are Binder enabled and also have Zendo archive to download.

REFERENCES

[1] A. Viljanen, R. Pirjola, E. Prácser, J. Katkalov, and M. Wik. Geomagnetically induced currents in Europe - Modelled occurrence in a continent-wide power grid. *Space Weather and in Space Climate*, 4:A09, 2014.

[2] Y. Abduallah, V. K. Jordanova, H. Liu, Q. Li, J. T. L. Wang, and H. Wang. Predicting solar energetic particles using SDO/HMI vector magnetic data products and a bidirectional LSTM network. *The Astrophysical Journal Supplement Series*, 260(1):16, May 2022.

[3] Y. Abduallah, J. T. L. Wang, P. Bose, G. Zhang, F. Gerges, and H. Wang. Forecasting the disturbance storm time index with bayesian deep learning. In *Proceedings of the 35th International Florida Artificial Intelligence Research Society Conference*, Hutchinson Island, Jensen Beach, Florida, May 2022.

[4] Y. Abduallah, J. T. L. Wang, Y. Nie, C. Liu, and H. Wang. DeepSun: Machine-learning-as-a-service for solar flare prediction. *Research in Astronomy and Astrophysics*, 21(7):160, Aug 2021.

[5] Y. Abduallah, J. T. L. Wang, Y. Shen, K. A. Alobaid, S. Criscuoli, and H. Wang. Reconstruction of total solar irradiance by deep learning. In *Proceedings of the 34th International Florida Artificial Intelligence Research Society Conference*, North Miami Beach, Florida, May 2021.

[6] Y. Abduallah, J. T. L. Wang, C. Xu, and H. Wang. A transformer-based framework for geomagnetic activity prediction. In *Proceedings of the Foundations of Intelligent Systems: 26th International Symposium on Methodologies for Intelligent Systems, Cosenza, Italy*, Heidelberg, Berlin: Springer-Verlag, Oct. 2022.

[7] O. W. Ahmed, R. Qahwaji, T. Colak, P. A. Higgins, P. T. Gallagher, and D. S. Bloomfield. Solar Flare Prediction Using Advanced Feature Extraction, Machine Learning, and Feature Selection. *Solar Physics*, 283(1):157–175, Mar. 2013.

[8] M. Al-Shedivat, A. G. Wilson, Y. Saatchi, Z. Hu, and E. P. Xing. Learning scalable deep kernels with recurrent structure. *Journal of Machine Learning Research*, 18(1), 2017.

[9] K. A. Alobaid, Y. Abduallah, J. T. L. Wang, H. Wang, H. Jiang, Y. Xu, V. Yurchyshyn, H. Zhang, H. Cavus, and J. Jing. Predicting CME arrival time through data integration and ensemble learning. *Frontiers in Astronomy and Space Sciences*, 9:1013345, Oct. 2022.

[10] E. Alpaydin. *Machine Learning: The New AI, The MIT Press Essential Knowledge series*. MIT Press, Cambridge, MA, USA, 2016.

[11] K. A. Althelaya, E.-S. M. El-Alfy, and S. Mohammed. Evaluation of bidirectional lstm for short-and long-term stock market prediction. In *Proceedings of the 9th International Conference on Information and Communication Systems*, 2018.

[12] F. Arias del Campo, M. C. Guevara Neri, O. O. Vergara Villegas, V. G. Cruz Sánchez, H. de Jesús Ochoa Domínguez, and V. García Jiménez. Auto-adaptive multilayer perceptron for univariate time series classification. *Expert Systems with Applications*, 181:115147, 2021.

[13] J. R. Ayala Solares, H.-L. Wei, R. J. Boynton, S. N. Walker, and S. A. Billings. Modeling and prediction of global magnetic disturbance in near-Earth space: A case study for Kp index using NARX models. *Space Weather*, 14(10):899–916, 2016.

[14] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *Computing Research Repository*, abs/1409.0473, 2014.

[15] R. Bala and P. Reiff. Improvements in short-term forecasting of geomagnetic activity. *Space Weather*, 10(6), 2012.

[16] R. Bala and P. Reiff. Improvements in short-term forecasting of geomagnetic activity. *Space Weather*, 10(6), 2012.

[17] W. T. Ball, Y. C. Unruh, N. A. Krivova, S. Solanki, and J. W. Harder. Solar irradiance variability: A six-year comparison between SORCE observations and the SATIRE model. *Astronomy & Astrophysics*, 530:A71, 2011.

[18] G. Barnes, K. D. Leka, C. J. Schrijver, T. Colak, R. Qahwaji, O. W. Ashamari, Y. Yuan, J. Zhang, R. T. J. McAteer, D. S. Bloomfield, P. A. Higgins, P. T. Gallagher, D. A. Falconer, M. K. Georgoulis, M. S. Wheatland, C. Balch, T. Dunn, and E. L. Wagner. A comparison of flare forecasting methods. I. Results from the "all-clear" workshop. *The Astrophysical Journal*, 829:89, Oct. 2016.

[19] G. Barnes, K. D. Leka, E. A. Schumer, and D. J. Della-Rose. Probabilistic forecasting of solar flares from vector magnetogram data. *Space Weather*, 5:S09002, Sept. 2007.

[20] M. Bekkar, H. K. Djemaa, and T. A. Alitouche. Evaluation measures for models assessment over imbalanced data sets. *Journal of Information Engineering and Applications*, 3:27–38, 2013.

[21] T. Z. Benmerar, T. Megherbi, M. Kachouane, R. Deriche, and F. O. Boumghar. Browser-level parallelism and interactive rendering APIs for scalable computation-intensive SaaS: Application to brain diffusion MRI. *IEEE Transactions on Services Computing*, 2018.

[22] A. O. Benz. Flare Observations. *Living Reviews in Solar Physics*, 5(1):1, Feb. 2008.

[23] S. Berkebile-Stoiser, A. M. Veronig, B. M. Bein, and M. Temmer. Relation between the Coronal Mass Ejection Acceleration and the Non-thermal Flare Characteristics. *Astrophysical Journal*, 753(1):88, July 2012.

[24] A. Bhaskar and G. Vichare. Forecasting of SYMH and ASYH indices for geomagnetic storms of solar cycle 24 including St. Patrick's day, 2015 storm using NARX neural network. *Journal of Space Weather and Space Climate*, 9:A12, 2019.

[25] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112:859 – 877, 2016.

[26] S. Bloemheuvel, J. van den Hoogen, D. Jozinović, A. Michelini, and M. Atzmueller. Graph neural networks for multivariate time series regression with application to seismic data. *International Journal of Data Science and Analytics*, 2022.

[27] D. S. Bloomfield, P. A. Higgins, R. T. J. McAteer, and P. T. Gallagher. Toward reliable benchmarking of solar flare forecasting methods. *The Astrophysical Journal*, 747:L41, Mar. 2012.

[28] F. Boberg, P. Wintoft, and H. Lundstedt. Real time Kp predictions from solar wind data using neural networks. *Physics and Chemistry of The Earth Part C-Solar-Terrestrial and Planetary Science*, 25:275–280, 2000.

[29] M. G. Bobra and S. Couvidat. Solar flare prediction using SDO/HMI vector magnetic field data with a machine-learning algorithm. *The Astrophysical Journal*, 798:135, Jan 2015.

[30] M. G. Bobra and S. Ilonidis. Predicting coronal mass ejections using machine learning methods. *The Astrophysical Journal*, 821(2):127, Apr. 2016.

[31] M. G. Bobra, X. Sun, J. T. Hoeksema, M. Turmon, Y. Liu, K. Hayashi, G. Barnes, and K. D. Leka. The Helioseismic and Magnetic Imager (HMI) vector magnetic field pipeline: SHARPs - Space-weather HMI Active Region Patches. *Solar Physics*, 289:3549–3578, Sept. 2014.

[32] P. J. Braspenning, F. Thuijsman, and A. J. M. M. Weijters, editors. *Artificial Neural Networks: An Introduction to ANN Theory and Practice*, volume 931 of *Lecture Notes in Computer Science*. Springer, 1995.

[33] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.

[34] T. V. Brito, J. Woodroffe, V. K. Jordanova, M. Henderson, and J. Birn. Particle tracing modeling of ion fluxes at geosynchronous orbit. *Journal of Atmospheric and Solar-Terrestrial Physics*, 177:131–140, Oct. 2018.

[35] R. K. Burton, R. L. McPherron, and C. T. Russell. An empirical relationship between interplanetary conditions and Dst. *Journal of Geophysical Research (1896-1977)*, 80(31):4204–4214, 1975.

[36] R. K. Burton, R. L. McPherron, and C. T. Russell. An empirical relationship between interplanetary conditions and Dst. *Journal of Geophysical Research (1896-1977)*, 80(31):4204–4214, 1975.

[37] B. Butcher and B. J. Smith. Feature engineering and selection: A practical approach for predictive models. *The American Statistician*, 74(3):308–309, 2020.

[38] L. Cai, S. Y. Ma, and Y. L. Zhou. Prediction of SYM-H index during large storms by NARX neural network from IMF and solar wind data. *Annales Geophysicae*, 28(2):381–393, 2010.

[39] E. Camporeale. The challenge of machine learning in space weather: Nowcasting and forecasting. *Space Weather*, 17(8):1166–1207, 2019.

[40] B. A. Carter, E. Yizengaw, R. Pradipta, J. M. Weygand, M. Piersanti, A. Pulkkinen, M. B. Moldwin, R. Norman, and K. Zhang. Geomagnetically induced currents around the world during the 17 March 2015 storm. *Journal of Geophysical Research: Space Physics*, 121(10):10,496–10,507, 2016.

[41] S. Chakraborty and S. K. Morley. Probabilistic prediction of geomagnetic storms and the Kp index. *Space Weather and in Space Climate*, 10:36, 2020.

[42] M. Chandorkar, E. Camporeale, and S. Wing. Probabilistic forecasting of the disturbance storm time index: An autoregressive Gaussian process approach. *Space Weather*, 15(8):1004–1019, 2017.

[43] M. Chandorkar, E. Camporeale, and S. Wing. Probabilistic forecasting of the disturbance storm time index: An autoregressive Gaussian process approach. *Space Weather*, 15(8):1004–1019, 2017.

[44] H. Chen and J. Li. Learning data-driven drug-target-disease interaction via neural tensor network. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, pages 3452–3458, 2020.

[45] P. Chen, S. Liu, C. Shi, B. Hooi, B. Wang, and X. Cheng. Neucast: Seasonal neural forecast of power grid time series. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, pages 3315–3321, 2018.

[46] P. F. Chen. Coronal Mass Ejections: Models and Their Observational Basis. *Living Reviews in Solar Physics*, 8(1):1, Apr. 2011.

[47] Y. Chen, W. B. Manchester, A. O. Hero, G. Toth, B. DuFumier, T. Zhou, X. Wang, H. Zhu, Z. Sun, and T. I. Gombosi. Identifying solar flare precursors using time series of SDO/HMI images and SHARP parameters. *Space Weather*, 17(10):1404–1426, Oct. 2019.

[48] A. Collado-Villaverde, P. Muñoz, and C. Cid. Deep neural networks with convolutional and LSTM layers for SYM-H and ASY-H forecasting. *Space Weather*, 19(6):e02748, 2021.

[49] G. Consolini and T. S. Chang. Magnetic field topology and criticality in geotail dynamics: Relevance to substorm phenomena. *Space Science Reviews*, 95:309–321, 2001.

[50] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[51] K. A. Costello. *Moving the Rice MSFM into a real-time forecast mode using solar wind driven forecast modules*. PhD Dissertation, Rice University, Houston, TX, 1998.

[52] N. Cristianini and E. Ricci. *Support Vector Machines*, pages 928–932. Springer US, Boston, MA, USA, 2008.

[53] I. Daglis, D. Baker, J. Kappenman, M. Panasyuk, and E. Daly. Effects of space weather on technology infrastructure. *Space Weather*, 2:S02004, 2004.

[54] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 29, Red Hook, NY: Curran Associates, Inc., 2016.

[55] Y. Deng, L. Lu, L. Aponte, A. M. Angelidi, V. Novak, G. E. Karniadakis, and C. S. Mantzoros. Deep transfer learning and data augmentation improve glucose levels prediction in type 2 diabetes patients. *Nature Portfolio Journal Digital Medicine*, 4:1013345, July 2021.

[56] J. S. Denker and Y. LeCun. Transforming neural-net output levels to probability distributions. In *Proceedings of the 3rd International Conference on Neural Information Processing Systems*, NIPS'90, San Francisco, CA, USA, 1990. Burlington, MA: Morgan Kaufmann Publishers Inc.

[57] M. H. Denton, M. G. Henderson, V. K. Jordanova, M. F. Thomsen, J. E. Borovsky, J. Woodroffe, D. P. Hartley, and D. Pitchford. An improved empirical model of electron and ion fluxes at geosynchronous orbit based on upstream solar wind conditions. *Space Weather*, 14(7):511–523, 2016.

[58] V. Domingo, I. Ermolli, P. Fox, C. Frohlich, M. Haberreiter, N. Krivova, G. Kopp, W. Schmutz, S. K. Solanki, H. C. Spruit, Y. Unruh, and A. Vogler. Solar surface magnetism and irradiance on time scales from days to the 11-year cycle. *Space Science Reviews*, 145:337–380, 2009.

[59] U. Faghihi and C. Kalantarpour. Using deep learning algorithms to detect the success or failure of the Electroconvulsive Therapy (ECT) sessions. *The International FLAIRS Conference Proceedings*, 34, Apr. 2021.

[60] G. Fisher, D. Bercik, B. Welsch, and H. Hudson. Global Forces in Eruptive Solar Flares: The Lorentz Force Acting on the Solar Atmosphere and the Solar Interior. *Solar Physics*, 277:59–76, Mar. 2012.

[61] K. Florios, I. Kontogiannis, S.-H. Park, J. A. Guerra, F. Benvenuto, D. S. Bloomfield, and M. K. Georgoulis. Forecasting Solar Flares Using Magnetogram-based Predictors and Machine Learning. *Solar Physics*, 293(2):28, Feb. 2018.

[62] Y. Gal and Z. Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

[63] P. T. Gallagher, Y.-J. Moon, and H. Wang. Active-region monitoring and flare forecasting - I. Data processing and first results. *Solar Physics*, 209:171–183, Sept. 2002.

[64] V. García, R. A. Mollineda, and J. S. Sánchez. Index of balanced accuracy: A performance measure for skewed class distributions. In H. Araujo, A. M. Mendonça, A. J. Pinho, and M. I. Torres, editors, *Proceedings of the Pattern Recognition and Image Analysis*, Heidelberg, Berlin: Springer, 2009.

[65] C. T. Gaunt and G. Coetzee. Transformer failures in regions incorrectly considered to have low GIC-risk. In *Proceedings of the IEEE Lausanne Power Tech*, 2007.

[66] H. Gleisner, H. Lundstedt, and P. Wintoft. Predicting geomagnetic storms from solar-wind data using time-delay neural networks. *Annales Geophysicae*, 14:679–686, 1996.

[67] I. J. Goodfellow, Y. Bengio, and A. C. Courville. *Deep Learning*. MIT Press, 2016.

[68] A. Graves. Practical variational inference for neural networks. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Proceedings of the Advances in Neural Information Processing Systems*, volume 24, 2011.

[69] L. J. Gray, J. Beer, M. Geller, J. D. Haigh, M. Lockwood, K. Matthes, U. Cubasch, D. Fleitmann, G. Harrison, L. Hood, J. Luterbacher, G. A. Meehl, D. Shindell, B. van Geel, and W. White. Solar influences on climate. *Reviews of Geophysics*, 48:RG4001, 2010.

[70] M. A. Gruet, M. Chandorkar, A. Sicard, and E. Camporeale. Multiple-hour-ahead forecast of the dst index using a combination of long short-term memory neural network and gaussian process. *Space Weather*, 16(11):1882–1896, 2018.

[71] M. A. Gruet, M. Chandorkar, A. Sicard, and E. Camporeale. Multiple-hour-ahead forecast of the Dst index using a combination of long short-term memory neural network and Gaussian process. *Space Weather*, 16(11):1882–1896, Nov. 2018.

[72] N. Y. Hammerla, J. Fisher, P. Andras, L. Rochester, R. Walker, and T. Ploetz. PD disease state assessment in naturalistic environments using deep learning. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, pages 1742–1748, 2015.

[73] R. A. Harrison. The nature of solar flares associated with coronal mass ejection. *Astronomy and Astrophysics*, 304:585, Dec. 1995.

[74] P. Heidke. Berechnung des erfolges und der güte der windstärkevorhersagen im sturmwarnungsdienst. *Geografiska Annaler*, 8(4):301–349, 1926.

[75] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[76] J. T. Hoeksema, Y. Liu, K. Hayashi, X. Sun, J. Schou, S. Couvidat, A. Norton, M. Bobra, R. Centeno, K. D. Leka, G. Barnes, and M. Turmon. The Helioseismic and Magnetic Imager (HMI) Vector Magnetic Field Pipeline: Overview and Performance. *Solar Physics*, 289(9):3483–3530, Sept. 2014.

[77] Z. Hu, T. Turki, N. Phan, and J. T. L. Wang. A 3D atrous convolutional long short-term memory network for background subtraction. *IEEE Access*, 6:43450–43459, 2018.

[78] Z. Hu, T. Turki, and J. T. L. Wang. Generative adversarial networks for stochastic video prediction with action control. *IEEE Access*, 8:63336–63348, 2020.

[79] G. Huang and L. Chen. Convex incremental extreme learning machine. *Neurocomputing*, 70(16-18):3056–3062, 2007.

[80] G.-B. Huang and L. Chen. Enhanced random search based incremental extreme learning machine. *Neurocomputing*, 71(16-18):3460–3468, 2008.

[81] N. Huang, Y. Xu, and H. Wang. Relationship between Intensity of White-light Flares and Proton Flux of Solar Energetic Particles. *Research Notes of the American Astronomical Society*, 2(1):7, Jan. 2018.

[82] X. Huang, H. Wang, L. Xu, J. Liu, R. Li, and X. Dai. Deep Learning Based Solar Flare Forecasting Model. I. Results for Line-of-sight Magnetograms. *Astrophysical Journal*, 856(1):7, Mar. 2018.

[83] C.-C. Hung, Y.-J. Chen, S. J. Guo, and F.-C. Hsu. Predicting the price movement from candlestick charts: a CNN-based approach. *International Journal of Ad Hoc and Ubiquitous Computing*, 34(2):111–120, 2020.

[84] F. Inceoglu, J. H. Jeppesen, P. Kongstad, N. J. H. Marcano, R. H. Jacobsen, and C. Karoff. Using machine learning methods to forecast if solar flares will be associated with CMEs and SEPs. *Astrophysical Journal*, 861(2):128, jul 2018.

[85] D. Iong, Y. Chen, G. Toth, S. Zou, T. I. Pulkkinen, J. Ren, E. Camporeale, and T. I. I. Gombosi. New findings from explainable SYM-H forecasting using gradient boosting machines. *Earth and Space Science Open Archive*, page 33, 2021.

[86] E.-Y. Ji, Y.-J. Moon, J. Park, J.-Y. Lee, and D.-H. Lee. Comparison of neural network and support vector machine methods for Kp forecasting. *Journal of Geophysical Research: Space Physics*, 118(8):5109–5117, 2013.

[87] H. Jiang, J. Jing, J. Wang, C. Liu, Q. Li, Y. Xu, J. T. L. Wang, and H. Wang. Tracing Hα fibrils through Bayesian deep learning. *The Astrophysical Journal Supplement Series*, 256(1):20, sep 2021.

[88] H. Jiang, J. Wang, C. Liu, J. Jing, H. Liu, J. T. L. Wang, and H. Wang. Identifying and tracking solar magnetic flux elements with deep learning. *The Astrophysical Journal Supplement Series*, 250(1):5, 2020.

[89] E. Jonas, M. Bobra, V. Shankar, J. Todd Hoeksema, and B. Recht. Flare Prediction Using Photospheric and Coronal Image Data. *Solar Physics*, 293(3):48, Mar. 2018.

[90] V. K. Jordanova, G. L. Delzanno, M. G. Henderson, H. C. Godinez, C. A. Jeffery, E. C. Lawrence, S. K. Morley, J. D. Moulton, L. J. Vernon, J. R. Woodroffe, T. V. Brito, M. A. Engel, C. S. Meierbachtol, D. Svyatsky, Y. Yu, G. Tóth, D. T. Welling, Y. Chen, J. Haiducek, S. Markidis, J. M. Albert, J. Birn, M. H. Denton, and R. B. Horne. Specification of the near-Earth space environment with SHIELDS. *Journal of Atmospheric and Solar-Terrestrial Physics*, 177:148–159, Oct. 2018.

[91] H. Kang, S.-H. Yang, J. Huang, and J. Oh. Time series prediction of wastewater flow rate by bidirectional lstm deep learning. *International Journal of Control Automation and Systems*, 18:3023–3030, 2020.

[92] Y. Kawabata, Y. Iida, T. Doi, S. Akiyama, S. Yashiro, and T. Shimizu. Statistical Relation between Solar Flares and Coronal Mass Ejections with Respect to Sigmoidal Structures in Active Regions. *Astrophysical Journal*, 869(2):99, Dec. 2018.

[93] A. Kendall and Y. Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Proceedings of the Advances in Neural Information Processing Systems*, volume 30, Red Hook, NY: Curran Associates, Inc., 2017.

[94] E. Kilpua, H. E. J. Koskinen, and T. I. Pulkkinen. Coronal mass ejections and their sheath regions in interplanetary space. *Living Reviews in Solar Physics*, 14(1):5, Nov. 2017.

[95] J. Kim, G. Englebienne, K. P. Truong, and V. Evers. Deep temporal models using identity skip-connections for speech emotion recognition. In *Proceedings of the 2017 Association for Computing Machinery on Multimedia Conference*, 2017.

[96] J. H. King and N. E. Papitashvili. Solar wind spatial scales in and comparisons of hourly Wind and ACE plasma and magnetic field data. *Journal of Geophysical Research: Space Physics*, 110(A2), 2005.

[97] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[98] A. J. Klimas, D. Vassiliadis, D. N. Baker, and D. A. Roberts. The organized nonlinear dynamics of the magnetosphere. *Journal of Geophysical Research: Space Physics*, 101(A6):13089–13113, 1996.

[99] M. Kline and L. Berardi. Revisiting squared-error and cross-entropy functions for training neural network classifiers. *Neural Computing & Applications*, 14(4):310–318, 2005.

[100] G. Kopp. Magnitudes and timescales of total solar irradiance variability. *Journal of Space Weather and Space Climate*, 6:A30, 2016.

[101] M. Kravchik and A. Shabtai. Detecting cyber attacks in industrial control systems using convolutional neural networks. In *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*, New York, NY: Association for Computing Machinery, 2018.

[102] A. C. Kren, P. Pilewskie, and O. Coddington. Where does Earth's atmosphere get its energy? *Journal of Space Weather and Space Climate*, 7:A10, 2017.

[103] N. A. Krivova, L. Balmaceda, and S. K. Solanki. Reconstruction of solar total irradiance since 1700 from the surface magnetic flux. *Astronomy & Astrophysics*, 467(1):335–346, 2007.

[104] J. B. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29(2):115–129, 1964.

[105] J. A. Lazzús, P. Vega, P. Rojas, and I. Salfate. Forecasting the Dst index using a swarm-optimized neural network. *Space Weather*, 15(8):1068–1089, 2017.

[106] J. A. Lazzús, P. Vega, P. Rojas, and I. Salfate. Forecasting the Dst index using a swarm-optimized neural network. *Space Weather*, 15(8):1068–1089, 2017.

[107] J. Lean. Evolution of the Sun's spectral irradiance since the Maunder Minimum. *Geophysical Research Letters*, 27(16):2425–2428, 2000.

[108] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[109] K. D. Leka and G. Barnes. Photospheric Magnetic Field Properties of Flaring versus Flare-quiet Active Regions. I. Data, General Approach, and Sample Results. *Astrophysical Journal*, 595(2):1277–1295, Oct. 2003.

[110] K. D. Leka and G. Barnes. Photospheric magnetic field properties of flaring versus flare-quiet active regions. II. Discriminant analysis. *The Astrophysical Journal*, 595:1296–1306, Oct. 2003.

[111] A. Lethy, M. A. El-Eraki, A. Samy, and H. A. Deebes. Prediction of the Dst index and analysis of its dependence on solar wind parameters using neural network. *Space Weather*, 16(9):1277–1290, 2018.

[112] J. Li, Z. Xing, and A. Kabir. Leveraging official content and social context to recommend software documentation. *IEEE Transactions on Services Computing*, 2018.

[113] K. Li, J. Daniels, C. Liu, P. Herrero, and P. Georgiou. Convolutional recurrent neural networks for glucose prediction. *IEEE Journal of Biomedical and Health Informatics*, 24(2):603–613, 2020.

[114] M. W. Liemohn, J. P. McCollough, V. K. Jordanova, C. M. Ngwira, S. K. Morley, C. Cid, W. K. Tobiska, P. Wintoft, N. Y. Ganushkina, D. T. Welling, S. Bingham, M. A. Balikhin, H. J. Opgenoorth, M. A. Engel, R. S. Weigel, H. J. Singer, D. Buresova, S. Bruinsma, I. S. Zhelavskaya, Y. Y. Shprits, and R. Vasile. Model evaluation guidelines for geomagnetic index predictions. *Space Weather*, 16(12):2079–2102, 2018.

[115] J. Lin and T. G. Forbes. Effects of reconnection on the coronal mass ejection process. *Journal of Geophysics*, 105(A2):2375–2392, Feb. 2000.

[116] X. Lin, Z. Quan, Z. Wang, T. Ma, and X. Zeng. KGNN: knowledge graph neural network for drug-drug interaction prediction. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, pages 2739–2745, 2020.

[117] C. Liu, N. Deng, J. T. L. Wang, and H. Wang. Predicting solar flares using SDO/HMI vector magnetic data products and the random forest algorithm. *The Astrophysical Journal*, 843:104, 2017.

[118] H. Liu. *Ph.D. Dissertation*. Ph.D. Dissertation, New Jersey Institute of Technology, Newark, NJ, 2020.

[119] H. Liu, C. Liu, J. T. L. Wang, and H. Wang. Predicting solar flares using a long short-term memory network. *Astrophysical Journal*, 877(2):121, jun 2019.

[120] H. Liu, C. Liu, J. T. L. Wang, and H. Wang. Predicting solar flares using a long short-term memory network. *The Astrophysical Journal*, 877:121, jun 2019.

[121] H. Liu, C. Liu, J. T. L. Wang, and H. Wang. Predicting coronal mass ejections using SDO/HMI vector magnetic data products and recurrent neural networks. *The Astrophysical Journal*, 890:12, feb 2020.

[122] H. Liu, C. Liu, J. T. L. Wang, and H. Wang. Predicting coronal mass ejections using SDO/HMI vector magnetic data products and recurrent neural networks. *Astrophysical Journal*, 890(1):12, feb 2020.

[123] H. Liu, Y. Xu, J. Wang, J. Jing, C. Liu, J. T. L. Wang, and H. Wang. Inferring vector magnetic fields from Stokes profiles of GST/NIRIS using a convolutional neural network. *The Astrophysical Journal*, 894:70, 2020.

[124] C. A. Loewe and G. W. Prölss. Classification and mean behavior of magnetic storms. *Journal of Geophysical Research: Space Physics*, 102(A7):14209–14213, 1997.

[125] C. Marzban. The ROC Curve and the Area under It as Performance Measures. *Weather and Forecasting*, 19(6):1106, Dec. 2004.

[126] P. N. Mayaud. *What is a Geomagnetic Index?*, chapter 2, pages 2–4. American Geophysical Union (AGU), 1980.

[127] N. Meinshausen and P. Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

[128] J. Mickens, J. Elson, and J. Howell. Mugshot: Deterministic capture and replay for JavaScript applications. In *Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation*, San Jose, CA: USENIX Association, April 2010.

[129] M. B. Moldwin and J. S. Tsu. *Stormtime Equatorial Electrojet Ground-Induced Currents*, chapter 3, pages 33–40. American Geophysical Union (AGU), 2016.

[130] R. L. Moore, D. A. Falconer, and A. C. Sterling. The Limit of Magnetic-shear Energy in Solar Active Regions. *Astrophysical Journal*, 750(1):24, May 2012.

[131] P. T. Newell, T. Sotirelis, K. Liou, C.-I. Meng, and F. J. Rich. A nearly universal solar wind-magnetosphere coupling function inferred from 10 magnetospheric state variables. *Journal of Geophysical Research: Space Physics*, 112(A1), 2007.

[132] N. Nishizuka, K. Sugiura, Y. Kubo, M. Den, and M. Ishii. Deep Flare Net (DeFN) Model for Solar Flare Prediction. *Astrophysical Journal*, 858(2):113, May 2018.

[133] N. Nishizuka, K. Sugiura, Y. Kubo, M. Den, S. Watari, and M. Ishii. Solar flare prediction model with three machine-learning algorithms using ultraviolet brightening and vector magnetograms. *The Astrophysical Journal*, 835:156, Feb. 2017.

[134] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[135] W. D. Pesnell, B. J. Thompson, and P. C. Chamberlin. *The Solar Dynamics Observatory (SDO)*, pages 3–15. Springer US, New York, NY, USA, 2012.

[136] J. W. Pratt. Remarks on zeros and ties in the Wilcoxon signed rank procedures. *Journal of the American Statistical Association*, 54(287):655–667, 1959.

[137] E. Priest and T. Forbes. The magnetic nature of solar flares. *The Astronomy and Astrophysics Review*, 10:313–377, 2002.

[138] Z. Qiao, S. Zhao, C. Xiao, X. Li, Y. Qin, and F. Wang. Pairwise-ranking based collaborative recurrent neural networks for clinical event prediction. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, 2018.

[139] H. Ramchoun, M. A. J. Idrissi, Y. Ghanou, and M. Ettaouil. Multilayer perceptron: Architecture optimization and training with mixed activation functions. In *Proceedings of the 2nd international Conference on Big Data, Cloud and Applications, BDCA 2017, Tetouan, Morocco, March 29-30, 2017*, 2017.

[140] G. K. Rangarajan. Indices of geomagnetic activity. *Geomatik*, 3:323–384, 1989.

[141] L. Rastätter, M. M. Kuznetsova, A. Glocer, D. Welling, X. Meng, J. Raeder, M. Wiltberger, V. K. Jordanova, Y. Yu, S. Zaharia, R. S. Weigel, S. Sazykin, R. Boynton, H. Wei, V. Eccles, W. Horton, M. L. Mays, and J. Gannon. Geospace environment modeling 2008-2009 challenge: D_{st} index. *Space Weather*, 11(4):187–205, 2013.

[142] D. V. Reames, C. K. Ng, and A. J. Tylka. Spatial Distribution of Solar Energetic Particles in the Inner Heliosphere. *Solar Physics*, 285(1-2):233–250, July 2013.

[143] S. Rhee, S. Seo, and S. Kim. Hybrid approach of relation network and localized graph convolutional filtering for breast cancer subtype classification. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, pages 3527–3534, 2018.

[144] J. L. Roeder and V. K. Jordanova. Chapter 8 - space weather effects and prediction. In V. K. Jordanova, R. Ilie, and M. W. Chen, editors, *Ring Current Investigations*, pages 245–269. Elsevier, 2020.

[145] F. Rosenblatt. *The Perceptron: A Theory of Statistical Separability in Cognitive Systems*, page 10. Cornell Aeronautical Laboratory, Inc., Rep. No. VG-1196-G-1. U.S, Buffalo, NY, USA, 1958.

[146] G. Rottman. The SORCE mission. *Solar Physics*, 230:7–25, 2005.

[147] T. W. Sager and R. A. Thisted. Maximum Likelihood Estimation of Isotonic Modal Regression. *The Annals of Statistics*, 10(3):690 – 707, 1982.

[148] J. Schou, J. M. Borrero, A. A. Norton, S. Tomczyk, D. Elmore, and G. L. Card. Polarization Calibration of the Helioseismic and Magnetic Imager (HMI) onboard the Solar Dynamics Observatory (SDO). *Solar Physics*, 275(1-2):327–355, Jan. 2012.

[149] C. J. Schrijver. A characteristic magnetic field pattern associated with all major solar flares and its use in flare forecasting. *Astrophysical Journall*, 655(2):L117–L120, Feb. 2007.

[150] M. Schuster and K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[151] R. Senanayake, S. O'Callaghan, and F. Ramos. Predicting spatio-temporal propagation of seasonal influenza using variational Gaussian process regression. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, pages 3901–3907, 2016.

[152] Y. Y. Shprits, R. Vasile, and I. S. Zhelavskaya. Nowcasting and predicting the Kp index using historical values and real-time observations. *Space Weather*, 17(8):1219–1229, 2019.

[153] S. Siami-Namini, N. Tavakoli, and A. S. Namin. The performance of lstm and bilstm in forecasting time series. In *Proceedings of the IEEE International Conference on Big Data*, 2019.

[154] F. Siciliano, G. Consolini, R. Tozzi, M. Gentili, F. Giannattasio, and P. De Michelis. Forecasting SYM-H index: A comparison between long short-term memory and convolutional neural networks. *Space Weather*, 19(2):e2020SW002589, 2021.

[155] F. Siciliano, G. Consolini, R. Tozzi, M. Gentili, F. Giannattasio, and P. De Michelis. Forecasting SYM-H index: A comparison between long short-term memory and convolutional neural networks. *Space Weather*, 19(2), 2021.

[156] H. Song, C. Tan, J. Jing, H. Wang, V. Yurchyshyn, and V. Abramenko. Statistical assessment of photospheric magnetic features in imminent solar flare predictions. *Solar Physics*, 254:101–125, Jan. 2009.

[157] Y. Song, A. Sailer, and H. Shaikh. Hierarchical online problem classification for IT support services. *IEEE Transactions on Services Computing*, 5(3):345–357, 2012.

[158] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[159] T. SunPy Community, S. J. Mumford, S. Christe, D. Pérez-Suárez, J. Ireland, A. Y. Shih, A. R. Inglis, S. Liedtke, R. J. Hewett, F. Mayer, K. Hughitt,

N. Freij, T. Meszaros, S. M. Bennett, M. Malocha, J. Evans, A. Agrawal, A. J. Leonard, T. P. Robitaille, B. Mampaey, J. Iván Campos-Rozo, and M. S. Kirk. SunPy—Python for solar physics. *Computational Science and Discovery*, 8(1):014009, Jan. 2015.

[160] Y. Tan, Q. Hu, Z. Wang, and Q. Zhong. Geomagnetic index Kp forecasting with LSTM. *Space Weather*, 16(4):406–416, 2018.

[161] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[162] R. Tozzi, P. De Michelis, I. Coco, and F. Giannattasio. A preliminary risk assessment of geomagnetically induced currents over the Italian territory. *Space Weather*, 17(1):46–58, 2019.

[163] D. Tran, M. W. Dusenberry, M. van der Wilk, and D. Hafner. *Bayesian Layers: A Module for Neural Network Uncertainty*, page 14660–14672. Curran Associates Inc., Red Hook, NY, USA, 2019.

[164] I. Tyass, A. Bellat, A. Raihani, K. Mansouri, and T. Khalili. Wind speed prediction based on seasonal ARIMA model. In *Proceedings of the Environment, Energy and Earth Sciences Web of Conferences*, volume 336, Dec. 2022.

[165] T. T. Um, F. M. J. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek, and D. Kulić. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In *Proceedings of the 19th Association for Computing Machinery International Conference on Multimodal Interaction*, New York, NY: Association for Computing Machinery, 2017.

[166] I. G. Usoskin. A history of solar activity over millennia. *Living Reviews in Solar Physics*, 14:3, 2017.

[167] L. van Driel-Gesztelyi and L. M. Green. Evolution of Active Regions. *Living Reviews in Solar Physics*, 12(1):1, Sept. 2015.

[168] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[169] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the Annual Conference on Neural Information Processing Systems*, 2017.

[170] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

[171] G. Vichare, N. Thomas, K. Shiokawa, A. Bhaskar, and A. K. Sinha. Spatial gradients in geomagnetic storm time currents observed by Swarm multispacecraft mission. *Journal of Geophysical Research (Space Physics)*, 124(2):982–995, 2019.

[172] V. Volodina and P. Challenor. The importance of uncertainty quantification in model reproducibility. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2197):20200071, 2021.

[173] C. B. Wang, J. K. Chao, and C.-H. Lin. Influence of the solar wind dynamic pressure on the decay and injection of the ring current. *Journal of Geophysical Research: Space Physics*, 108(A9), 2003.

[174] X. Wang, Y. Chen, G. Toth, W. B. Manchester, T. I. Gombosi, A. O. Hero, Z. Jiao, H. Sun, M. Jin, and Y. Liu. Predicting solar flares with machine learning: Investigating solar cycle dependence. *Astrophysical Journal*, 895(1):3, May 2020.

[175] J. A. Wanliss and K. M. Showalter. High-resolution global storm index: Dst versus SYM-H. *Journal of Geophysical Research (Space Physics)*, 111(A2):A02202, 2006.

[176] D. F. Webb and T. A. Howard. Coronal Mass Ejections: Observations. *Living Reviews in Solar Physics*, 9(1):3, June 2012.

[177] D. S. Wilks. Sampling distributions of the Brier score and Brier skill score under serial dependence. *Quarterly Journal of the Royal Meteorological Society*, 136(653):2109–2118, Oct. 2010.

[178] J. R. Woodroffe, S. K. Morley, V. K. Jordanova, M. G. Henderson, M. M. Cowee, and J. G. Gjerloev. The latitudinal variation of geoelectromagnetic disturbances during large (Dst$\leq$-100 nT) geomagnetic storms. *Space Weather*, 14(9):668–681, 2016.

[179] C. J. Wu, N. A. Krivova, S. K. Solanki, and I. G. Usoskin. Solar total and spectral irradiance reconstruction over the last 9000 years. *Astronomy & Astrophysics*, 620:A120, 2018.

[180] C. J. Wu, I. G. Usoskin, N. Krivova, G. A. Kovaltsov, M. Baroni, E. Bard, and S. K. Solanki. Solar activity over nine millennia: A consistent multi-proxy reconstruction. *Astronomy & Astrophysics*, 615:A93, 2018.

[181] G. Wu, M. He, W. Chen, J. Wei, and H. Zhong. X-Check: Improving effectiveness and efficiency of cross-browser issues detection for JavaScript-based Web applications. *IEEE Transactions on Services Computing*, 2018.

[182] S. Yashiro and N. Gopalswamy. Statistical relationship between solar flares and coronal mass ejections. *Proceedings of the International Astronomical Union*, 4(S257):233–243, 2008.

[183] K. L. Yeo, N. A. Krivova, S. K. Solanki, and K. H. Glassmeier. Reconstruction of total and spectral solar irradiance from 1974 to 2013 based on KPVT, SoHO/MDI, and SDO/HMI observations. *Astronomy & Astrophysics*, 570:A85, 2014.

[184] B. Yu, H. Yin, and Z. Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, pages 3634–3640, 2018.

[185] Y. Yuan, F. Shih, J. Jing, and H. Wang. Automated flare forecasting using a statistical learning technique. *Research in Astronomy and Astrophysics*, 10:785–796, Aug. 2010.

[186] V. B. Yurchyshyn, H. Wang, P. R. Goode, and Y. Deng. Orientation of the magnetic fields in interplanetary flux ropes and solar filaments. *The Astrophysical Journal*, 563(1):381–388, 2001.

[187] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th Association for Computing Machinery and Special Interest Group on Knowledge Discovery in Data Conference on Knowledge Discovery & Data Mining*, 2021.

[188] F. Zhang, N. Feng, Y. Liu, C. Yang, J. Zhai, S. Zhang, B. He, J. Lin, and X. Du. PewLSTM: Periodic LSTM with weather-aware gating mechanism for parking behavior prediction. In *Proceedings of the International Joint Conferences on Artificial Intelligence 2020*, pages 4424–4430, 2020.

[189] Q. Zhang, L. Liu, K. Ma, C. Zhuo, and Y. Zheng. Cross-denoising network against corrupted labels in medical image segmentation with domain shift. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, pages 1047–1053, 2020.

[190] Y. Zhang, S. Pal, M. Coates, and D. Üstebay. Bayesian graph convolutional neural networks for semi-supervised classification. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, 2019.

[191] I. S. Zhelavskaya, R. Vasile, Y. Y. Shprits, C. Stolle, and J. Matzka. Systematic analysis of machine learning and feature selection techniques for prediction of the Kp index. *Space Weather*, 17(10):1461–1486, 2019.