Solar Phys (2019) 294:50
https://doi.org/10.1007/s11207-019-1434-6

Forecasting Sunspot Time Series Using Deep Learning Methods

Zeydin Pala[1] · **Ramazan Atici**[2]

Received: 21 January 2019 / Accepted: 27 March 2019 / Published online: 2 May 2019

Abstract To predict Solar Cycle 25, we used the values of sunspot number (SSN), which have been measured regularly from 1749 to the present. In this study, we converted the SSN dataset, which consists of SSNs between 1749 – 2018, into a time series, and made the ten-year forecast with the help of deep-learning (DL) algorithms. Our results show that algorithms such as long-short-term memory (LSTM) and neural network autoregression (NNAR), which are DL algorithms, perform better than many algorithms such as ARIMA, Naive, Seasonal Naive, Mean and Drift, which are expressed as classical algorithms in a large time-series estimation process. Using the R programming language, it was also predicted that the maximum amplitude of Solar Cycle (SC) 25 will be reached between 2022 and 2023.

Keywords Sunspot number · Statistics · Solar cycle

1. Introduction

Solar activity has significant impacts on human beings, other living beings, and vital technologies in the world. Due to the changes in solar activity with a period of about 11 years, deviations in the near-Earth space in the interplanetary environment and the magnetosphere affect the ionospheric plasma density with this period. In addition, changes in solar activity influence aero-navigation, space-navigation, spaceflights, radars, high-frequency radio communications, and ground electrical lines. These changes also potentially affect climate and living organisms in the world, including humans (Aguirre, Letellier, and Maquet, 2008; Pesnell, 2008; Pishkalo, 2008; Petrovay, 2010). For this reason, the predictions of solar activity are important both for improving the ionosphere model and for other applications and basic understanding of the behavior of the Sun.

✉ R. Atici
r.atici@alparslan.edu.tr

[1] Department of Computer Engineering, Faculty of Engineering, Mus Alparslan University, Muş, Turkey

[2] Faculty of Education, Mus Alparslan University, Muş, Turkey

The sunspot number (SSN), which is one of the leading indices used to estimate the sunspot cycle (SC), is an important parameter in many areas, including ionospheric investigations (Sagir *et al.*, 2015; Atici, 2018; Kim, Kim, and Chang, 2018; Sagir, Atici, and Ozcan, 2018; Sagir and Yesil, 2018). Due to the estimated difficulty, importance, and potential applications of the time series based on this parameter, it is still an active field. Several models have been proposed to estimate SC parameters (Layden *et al.*, 1991; Thompson, 1993; Dikpati, De Toma, and Gilman, 2006; Kilcik *et al.*, 2009; Pesnell, 2008; Petrovay, 2010; Pishkalo, 2008, 2014). In these models, SSN time-series values are used, which generally have available the data of solar activity from 1749 to 2018.

Modeling time series and predicting the future, as a result, is of great importance for many areas in practical life. Recently, considerable work has been done on this subject. Numerous important models have been introduced in the literature to model and predict the time series accurately and efficiently. The importance of time series appears to see the future, especially by looking at the past. Plans for the future are vital for many areas (Adhikari and Agrawal, 2013; Raissi and Karniadakis, 2018; Sirignano and Spiliopoulos, 2018). Therefore, time series are used extensively in the fields of economics, science, business, and finance. Today, different algorithms are used with the help of powerful computers to estimate time series. In addition to classical algorithms, machine learning (ML) and deep-learning (DL) algorithms are used to predict time series.

The most important element for ML is undoubtedly the data. By using large datasets, which are the raw material of ML, it is possible to grasp the more complex phenomena much better than before. The history of ML dates back to the 1980s and the history of DL, which emerged as a subfield of ML, dates back to the 2010s. However, state-of-the art DL algorithms have taken the prediction process one-step further.

The aim of this study is to make an estimation of SSN for the next ten years with the help of DL algorithms, using the monthly observation data of the sunspot time series between 1749 and 2018, The main contributions of this study are as follows:

i) Studies have been conducted based on SSN data between the years 1749 and 2013. The purpose of this study is to estimate the results of the sunspot data based on the monthly data between the years 1749 and 2018 by using DL algorithms and to show that the success of DL algorithms on such a large time series is better than traditional algorithms.
ii) To compare the performance of long-short-term memory (LSTM), neural network autoregression (NNAR), ARIMA, Naive, and Seasonal Naive algorithms using the R programming language and libraries.
iii) To make the training, testing, and validation of the sunspot dataset used as a univariate time series more realistic with the help of rsample, one of the popular libraries of the R language.

2. Background

In this section, we briefly summarize the technical information as regards the general structure of the time series and the algorithms used especially in the study. If we express a case that is measured at time t (where t can be annual, quarterly, monthly, weekly, daily, and hourly data) and forms a time series, we can briefly show it in y_t form. In the beginning, $t = 0$, an ordered event measured at the moment, at $t = t$, the initial and future values can be expressed as follows:

$$y_0, \ldots, y_{t-3}, y_{t-2}, y_{t-1}, y_t, y_{t+1}, y_{t+2}, y_{t+3}, \ldots \quad (1)$$

Each y_t-value in the time series results in an autoregressive model when regressed in the same time series, depending on previous values such as y_{t-1}. We can mathematically express such a model:

$$y_t = \beta_0 + \beta_1 y_{t-1} + \epsilon_t \tag{2}$$

In such a model, a response variable for the previous time is the predictor variable for the next step. Errors $[\epsilon_t]$ have the usual assumptions as regards errors in a simple linear regression model.

The degree of an autoregression (AR) model depends on the number of previous values in the time period used. In this case, the degree of the model in Equation 2 is AR (1).

If we want to find the prediction of the sunspot time series by considering the previous two years, *i.e.* y_{t-1} and y_{t-2}, the second order AR (2) model can be expressed as:

$$y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \epsilon_t \tag{3}$$

Considering the previous two models, in the most general form, $t-1, t-2, t-3, \ldots, t-k$, k-order can be expressed as an autoregression model AR (k).

The correlation between the coefficients of the two values in the time series is called the autocorrelation function (ACF), which is a method that measures the relationships between the values observed at time t and the previous values. The ACF can be shown as (Shumway and Stoffer, 2011)

$$\rho(y_t, y_{t-k}) = Corr(y_t, y_{t-k}) = \frac{\gamma(y_t, y_{t-k})}{\sqrt{\gamma(y_t, y_t)\gamma(y_{t-k}, y_{t-k})}} \tag{4}$$

where ACF measures the linear predictability of the time series at time $t-k$ using only the value y_t.

In addition, the partial autocorrelation function (PACF) method is used to determine the order of an autoregressive model.

2.1. Long-Short-Term Memory (LSTM)

LSTM is a special type of Recurrent Neural Network (RNN), one of the deep-learning (DL) algorithms. It outperforms the most advanced deep neural networks (DNNs) in numerous tasks, especially in speech and handwriting recognition (Zazo *et al.*, 2016). This algorithm consists of sequential special cells with structural inputs and outputs. Data progressing through LSTM cells is obtained as the output data for a cell, while it is used as input data for the next cell. With this aspect, LSTM can memorize data sequences longer than RNNs. Data in each cell can be destroyed, filtered, or collected for subsequent cells using special gates (by using three multiplicative gate units: the input gate, the output gate, and the forget gate). Therefore, doors based on the sigmoidal neural network allow the respective cell to pass or destroy the data (Lewis, 2016).

2.2. NNAR Method in R

One of the most important tools used in machine learning, inspired by the structure of the brain, artificial neural networks (ANNs) are the answer to making computers more human-like. They allow the modeling of complex nonlinear relationships between input regressors and the response variable. In a time-series model, the delayed values of the time series can be used as input to a neural network, as we used delayed values in a linear AR model.

This is called a neural network autoregression (NNAR) model. The NNAR model is a parametric and nonlinear estimation model (Sena and Nagwani, 2016). Such a model is used in the R environment with the parameters p and k (NNETAR (p, k) as feed-forward neural networks with a lagged input for forecasting univariate time series and single hidden layer). Here, p shows the lagged input, while k indicates the number of nodes in the hidden layer. For example, NNETAR (120, 11) used the input to model the last 120 observations $(y_{t-2}, y_{t-2}, y_{t-3}, \ldots, y_{t-120})$ with 11 neurons in the hidden layer forecasting the output y_t.

2.3. Performance Metrics for Evaluation

To examine the performance of the models by using sunspot time-series numeric values, root mean squared error (RMSE) is used (Adhikari and Agrawal, 2013). It measures the differences or residuals between actual and predicted values.

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{5}$$

where N is the total number of observations. y_i is the actual value; whereas $\hat{y}_i$ is the predicted value. The main advantage of using RMSE is that it penalizes extreme errors that occur while forecasting.

3. Experimental Study and Analysis

All analyses performed in this study were performed in the environment of R software version 3.5.1 (2 July 2018), which is an open-source language, free and extremely powerful. Note that all experiments reported here used a PC Intel i5 at 3.2 GHz CPU and 8 GB of RAM on a Windows 10 Pro operating system installed on a SATA drive. The sunspot number used in this study was obtained by the V2.0 series from the SILSO website (www.sidc.be/silso/datafiles) as monthly mean measurements (Sunspot data from the World Data Center SILSO, Royal Observatory of Belgium, Brussels).

The data in the sunspot dataset downloaded from the SILSO website in CSV format are composed of monthly data between 1749 and 2018. The data used belong to a period of 270 years and consist of a total of 3240 months of observation. Before the estimation process, in the data-processing phase, the non-measured observations, which were zero in the dataset, were kept the same as the previous robust observations, thus trying to ensure the integrity of the dataset. To improve forecasts and to help understand the time series better, it is also useful to try to split a time series into several components by using a decomposition method. The original time-series data and decomposition graphs of the data obtained in the measurements are shown in Figures 1 and 2, respectively. Based on the data structure, time-series data often have cyclic patterns in which observations rise and fall for long periods of time.

By using the decomposition method in R, the three components of the sunspot time series are shown in Figure 2. The remainder component shown at the bottom of the graph is the residual when the seasonal and trend-cycle components have been subtracted from the time-series data. As shown in Figure 2, SSN characteristics composed of the sine-wave trend and changing variance. As seen in Figures 1 and 2, the high number of observations makes the estimation process difficult due to the short-term variability in the dataset as well as the long-term irregularities.

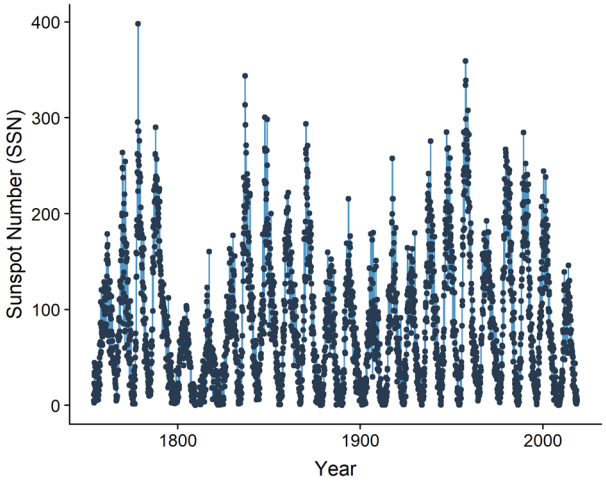

Figure 1 Sunspot monthly time series between 1749 and 2018.

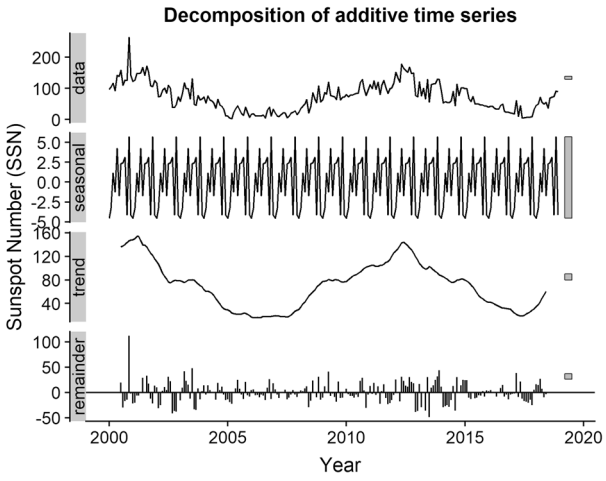

Figure 2 The sunspot data series between 2000 and 2018 (*top*) and its three additive components.

Table 1 Cross-validation plan on sunspot dataset.

Slice number of dataset	Slice length	Rolling-slice length	Training length	Validation length	Testing length	Skip-span length
6	1800	240	100 [year] × 12 = 1200	1200/3 = 600	50 [year] × 12 = 600	20 [year] × 12 = 240
12	1080	180	70 [year] × 12 = 840	840/3 = 280	20 [year] × 12 = 240	15 [year] × 12 = 180

The cross-validation (CV) method is used to make the prediction more realistic. With this method, the data of 3240 observations were analyzed with the help of the RStudio rsample library, which uses the rolling-origin-forecast resampling method. Table 1 shows the 6-slices and 12-slices CV estimation plan of the sunspot dataset, which consists of a total of 3240 months of observation.

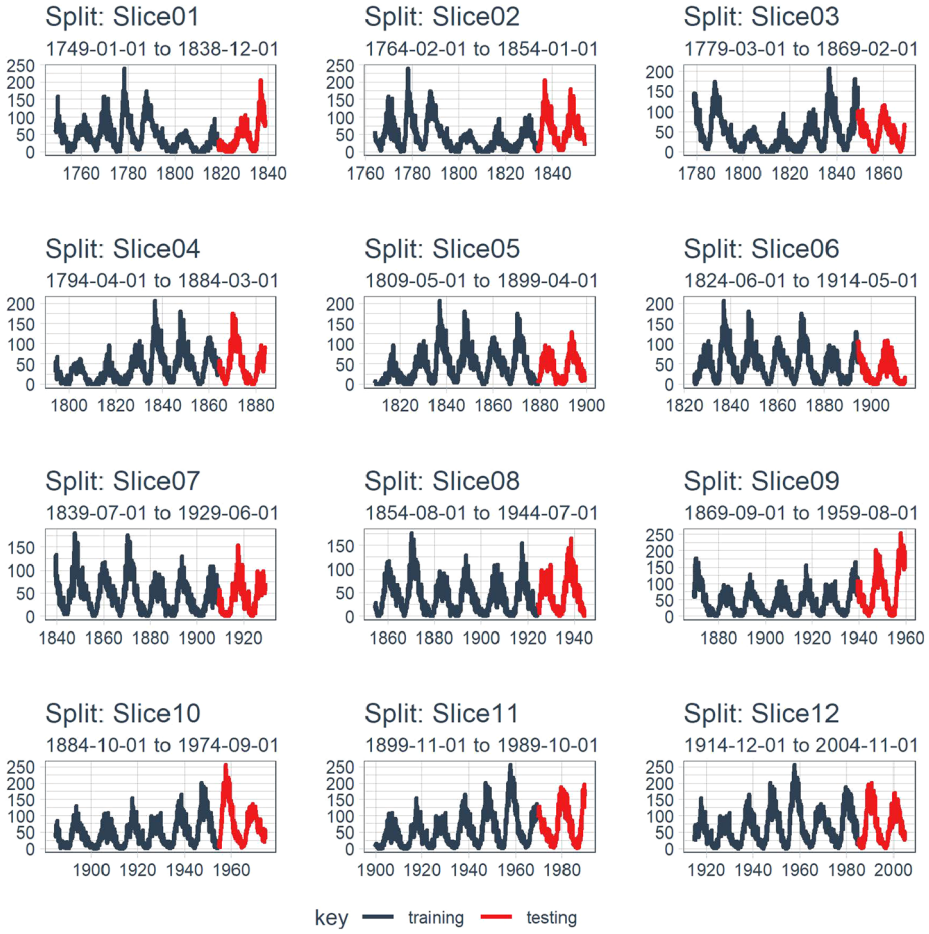

Figure 3 Sunspot dataset divided into 12 parts, as training and testing.

Each part of the sunspot dataset, which was set to 12 slice-based before the LSTM model was used, was divided into two parts as training and testing, as shown in Table 1. In addition, one third of the data allocated for training was reserved for the validation process. After this stage, each part of the dataset consists of training, validation, and testing. The training part of the dataset shown in the same table is used to train the LSTM model, the validation part is used to set the LSTM hyperparameters, and the test part is used to measure the actual performance of the LSTM model. The sampling window with each progressive slice of the train and test splits are shown in Figure 3.

The sunspot dataset shown in 12 separate pieces of data in Figure 3 is composed of data from the first slice 01 January 1749 to 01 December 1838, while the second slice consists of data in the range of 01 February 1764 to 01 January 1854. The second slice starts here as far as the rolling dataset, called the skip-span. In other words, the next slice starts 180 records ahead of the previous slice. The skip-span is 180 records for the 12-slice-based model and 240 for the 6-slice-based plan.

Table 2 LSTM model training/test parameters.

Model cross-validation strategy	Number of training/test observation	Batch-size	Number of training/test iteration	Number of epochs
6 slice-based model	1200/600	10	120/60	100
12 slice-based model	840/240	10	84/24	100

3.1. LSTM Model Parameters

To improve the performance of the LSTM algorithm, training, validation, and testing data were scaled with the help of R's recipes package. If the data to be trained with the support of machine learning are large, training the existing data at one time may not give a good result in terms of performance. Instead of using the dataset as a single part, it is preferred to divide it and use multiple parts.

In this study, the same strategy was applied for the sunspot dataset. As shown in Table 2, for the sunspot dataset that was divided into 6 parts in the first place, training, test, and batch-size of 1200, 600, and 10 were used, respectively. Therefore, the number of training and test iterations was calculated as 120 and 60, respectively. Since DL algorithms use the gradient-descent algorithm to optimize their own models, weights are constantly updated by training the dataset to achieve more accurate estimates. Therefore, for better training of the LSTM model, the epoch value used was 100.

3.2. Results

The training, validation, and testing of the LSTM model were performed using the sunspot dataset configured in two different ways. Firstly, after the last part of the 6-slice dataset was trained and tested with the LSTM model, the same process was applied to the last slice of the 12-slice dataset. As shown in Figure 4, the loss of the 6-piece rolling-origin-forecast-resampling model is lower than for the 12-piece rolling-origin forecast of the resampling model.

In addition, as the 6-slice rolling origin forecast appears on the resampling model (left), the system has stopped the training process at approximately 60 epochs, since the validation loss remains constant. We can also say that the training performance is better than the validation performance, as shown in the graph on the left.

For the 6-slice and 12-slice models, the prediction for the last slices on training and testing are shown in Figures 5 and 6, respectively. In Figures 5 and 6, the performance of the model on the training and test data is shown graphically with the help of real-time periods. Red lines displayed at certain intervals visually present the performance of the training (left) and test (right-hand) estimates. It is clear that the estimation results using training data in both model plans are more successful than the test results.

In Figure 7, the RMSE test values of the final parts of the 6-part and 12-part approaches of the sunspot data were close to each other, but the 6-part approach showed better predictive performance. The RMSE test values for the 6-part and 12-part estimates were 35.9 and 36.9, respectively.

Using the LSTM model, each part of the 6-segment sunspot dataset was subjected to training, verification, and testing. At the end of training, verification, and testing process, the next 120 months, in other words, a ten-year (2018 – 2028) prediction was made, as shown in Figure 8. The LSTM model was used to estimate the time interval between 2018 and 2028

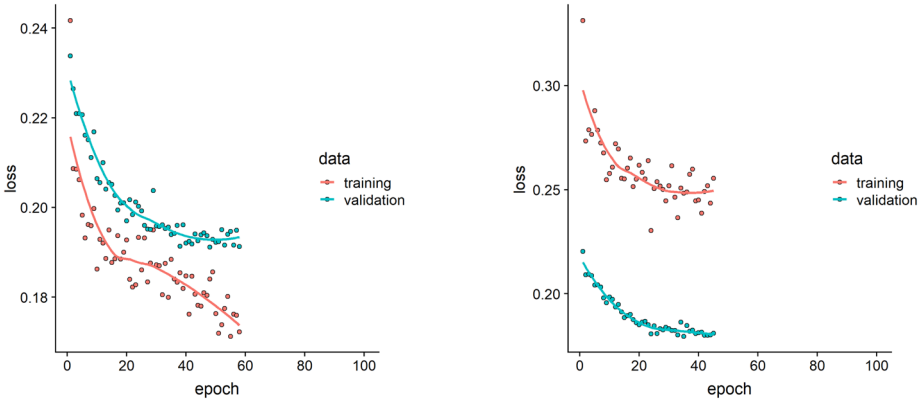

Figure 4 Training/validation loss-epoch graphics for 12-slices and 6-slices plan, respectively.

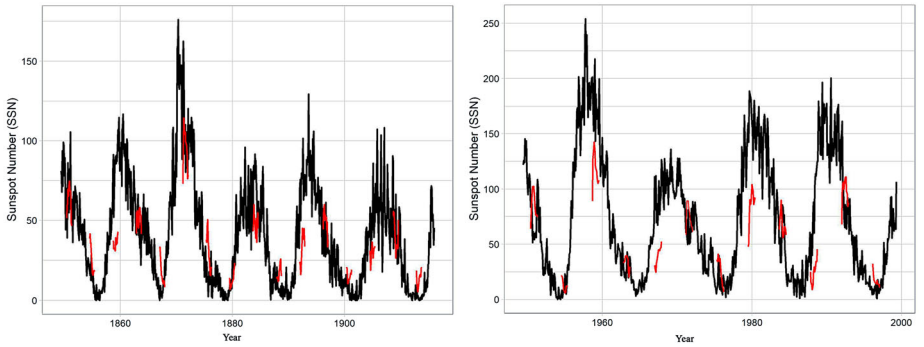

Figure 5 Training (*left-hand*) and test (*right-hand*) forecast graphs of the last piece of the six-part sunspot dataset (actual: *black*, prediction: *red*).

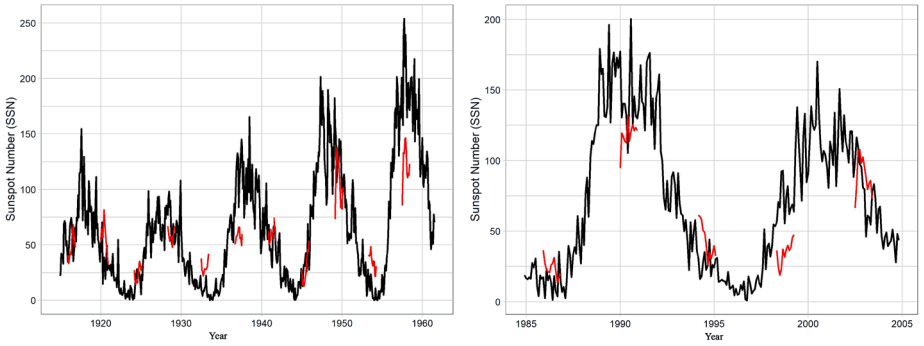

Figure 6 Training (*left*) and test (*right*) forecast graphs of the last piece of the 12-slice sunspot dataset (actual: *black*, prediction: *red*).

for the sunspot dataset containing 3240 observational data from 1749 to 2018, covering a period of 270 years.

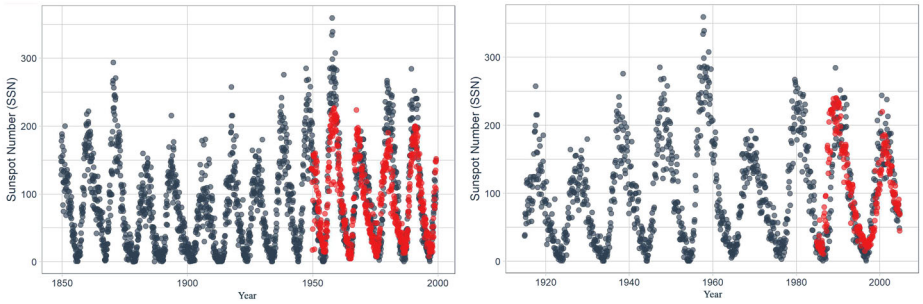

Figure 7 Forecast graphs of the last piece of the 6-part and 12-part of sunspot dataset (actual: *black*, prediction: *red*).

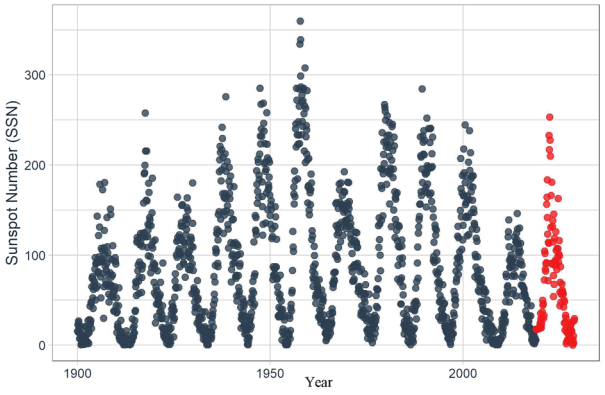

Figure 8 Sunspot dataset: ten-year forecast (forecast horizon 2018 – 2028).

The LSTM model is created using the Keras model sequential function, which is composed of a linear stack of layers. In this study, two LSTM layers, each consisting of 50 units, were used. The first LSTM layer was used for inputs. The model entries consist of time steps and a number of features, respectively. The second LSTM layer is defined as the first layer. In the second LSTM layer, the 2D data shape is returned while the 3D shape returned from the first LSTM layer. While compiling the model, adam was used as the optimizer. As an optimization algorithm adam can be used to update network weights iterative based on training data. It requires relatively low memory and usually works well even with a little tuning of hyperparameters. In the same model, mae metric, mean absolute error (mae) is a measure of a difference between two continuous variables, for the loss was used. Here, we have made the next ten-year SSN forecast from 2018 using DL methods. With the help of the LSTM model, we have shown successful results for a very large dataset like sunspot. We also saw that the cross-validation method for a time series revealed more satisfactory results.

In the R environment, which is a language for statistical computing and graphics, we used the Mean, Naïve, Seasonal Naive, Drift, ARIMA, and NNAR models to predict sunspot dataset as shown in Figure 9. The sunspot dataset for the LSTM DL model was modeled in two separate approaches, first as 6 pieces, then as 12 pieces. To calculate the average RMSE for each approach, the average of the 6-part approach (six different RMSE values) was calculated. Then the mean of the 12-part approach (12 RMSE values) was calculated. In order to use the method, we used in the LSTM model and thus to achieve more realistic

Table 3 RMSE values of the algorithms.

Model/method name	RMSE value
LSTM 6-part approach	35.9
LSTM 12-part approach	36.9
NNAR method	42.41
ARIMA model	45.60
Mean method	45.60
Naïve method	90.51
Seasonal Naïve method	103.61
Drift method	97.06

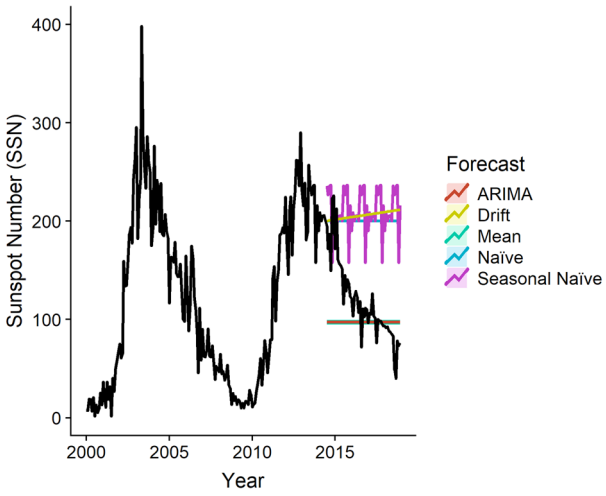

Figure 9 54-month forecast graph using classical algorithms.

results, we divided the sunspot dataset into 12 separate parts, which were obtained on a monthly basis and contained 3240 records. Each dataset part obtained as a result of the division process contained 540 records. Here we studied classic models using the last part of the time series studied. Of the 540 records, 90% were used for training (486 records) and the remaining 10% (54 records) were used for testing. As a result of the training of the algorithms used in this study, RMSE values of the algorithm test procedures are shown in Table 3.

As shown in Table 3, the best-performing algorithms are the LSTM 6-part approach, the LSTM 12-part approach, NNAR, and ARIMA, with RMSE values of 35.9, 36.9, 42.41, and 45.60, respectively. Test results of the NNAR model applied to the sunspot dataset and the future-ten-year forecast using the NNAR model are shown in Figures 10 and 11.

3.3. Comparison with Other Studies

3.3.1. Comparison of Analysis Results in Terms of Statistical Accuracy

Siami-Namini and Namin (2018) compared LSTM and ARIMA models by using different datasets. They found the mean value of RMSE for ARIMA and LSTM to be 511.481 and 64.213, respectively. In the same study, they showed that the LSTM model performed better

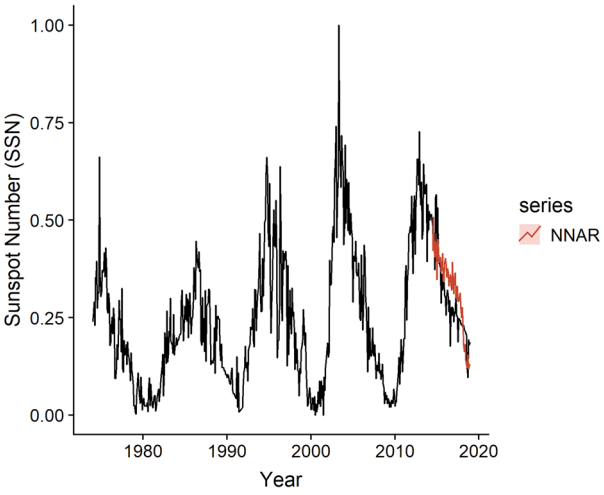

Figure 10 Test results of the NNAR model applied to the sunspot dataset.

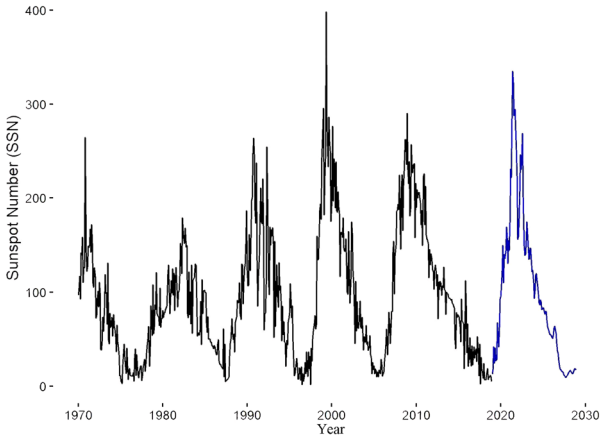

Figure 11 The future-ten-year forecast using the NNAR model.

than the ARIMA model. We agree with Siami-Namini and Namin (2018) that the LSTM model performs better than the classical models. Maleki *et al.* (2018) compared the performance of ARIMA and NNAR models and showed that the NNAR model performed better than the ARIMA model. Dancho (2018) tried to predict the next ten years by using LSTM in a sunspot dataset that contains monthly data between 1749 – 2013 in the R dataset. The RMSE on the test data was found to be approximately 26.91, based on the sunspot dataset between the years of 1749 and 2013, containing 265 years (with 3177 records). However, the data in our study cover the years 1749 to 2018, a period of 270 years, and contain 3240 records. There is a five-year difference between the two datasets used; in other words, 60 months. They used the ready and processed data in the R datasets library. However, we have downloaded the sunspot dataset from the SILSO site. Therefore, we tried to use the most recent data in our study. Considering the above-mentioned differences between the two datasets, it is expected that the RMSE values will be different. In the sunspot data downloaded from the SILSO site, some measurements were zero. Such values are kept the same as the previous measurement value. In this study, we used both LSTM and NNAR models and these models showed better performance than the ARIMA and Naive models.

Both the methodological approach to analyzing the problem and to solving the problem are quite similar to the approach of Maleki *et al.* (2018). Shaikh *et al.* (2008) analyzed by using the sunspot dataset between 1818 and 2002 with the help of the Fourier transform technique (FFT). Chattopadhyay and Chattopadhyay (2012) used ARIMA and an autoregressive neural network (AR-NN) models to predict sunspot time series between 1992 and 2008. For a homogeneity test of sunspot time series, they used Pettit's test, the Mann–Kendall test for the trend, and the Kolmogorov–Smirnov test for the normal distribution. In their main conclusion, they showed that the AR-NN(3) model gave better performance than ARMA(3, 1) and AR(3) models. Gkana and Zachilas (2015) used Average Mutual Information (AMI) and the False Nearest Neighbors (FNN) methods to estimate suitable embedding parameters for monthly sunspot time series between 1749 and 2013. They used a neural network-type core algorithm to predict the 2013 – 2014 interval.

We have briefly summarized the work that uses nearly the same dataset as above, especially in our study. Our study is based on SNN data from the years 1749 to 2018. For a growing dataset with new records added every day, such as sunspots, it is more advantageous to make ten-year forecasts with DL algorithms.

3.3.2. Comparison of SSN Forecasts

Rigozo *et al.* (2011) carried out a study to estimate the strength of solar activity in both Solar Cycle 24 and 25 based on extrapolation of spectral components. They estimated that the maximum number of sunspots in Cycle 25 will occur in April 2023 with a sunspot number of 132.1 (with a solar-cycle length of 118 months or 9.8 years). In the same study, Solar Cycle 24 was also estimated, and the maximum SSN value was projected to be 113.3 in November 2013. However, when the actual SSN values were examined, the maximum SSN value was reached in February 2014 with a SSN value of 146.1. Thus, it can be said that Rigozo *et al.* (2011) reached their maximum SSN estimation with an about 30% deviation from the actual SSN value, so the estimation methods are good methods.

In a similar study, Quassim, Attia, and Elminir (2007) estimated the amplitude and geomagnetic activity of SCs 24 and 25 by using a neural fuzzy approach. As a result, they stated that Solar Cycle 24 will reach the maximum with a SSN of 110 in 2011, and Solar Cycle 25 would reach maximum with a SSN of 116 in 2021. Their estimates are not consistent with reality for Solar Cycle 24, because the maximum SSN of Solar Cycle 24 was in February 2014. Similarly, it can be stated that, for Solar Cycle 25, their estimates may not be reliable.

In another study, Okoh *et al.* (2018) used a method called the Hybrid Regression-Neural Network, which combines regression analysis and neural network learning to estimate SSN. They predicted that the end of Solar Cycle 24 would be in March 2020 ($\pm$ seven months) with a SSN of 5.4 and the maximum of Solar Cycle 25 would be January 2025 ($\pm$ six months) with a SSN of 122.1 ($\pm$ 18.2). Kane (2007) predicted that the maximum amplitude for Solar Cycle 25 would be between 112 and 127 (mean 119) between 2022 and 2023.

Penn and Livingston (2011) found a decrease in the strength of sunspot magnetic fields, determined a future trend, and concluded that Solar Cycle 24 would peak in SSN value at about 66 and Solar Cycle 25 would be weaker than Solar Cycle 24, with SSN of 59. However, it is seen that the estimation is not correct for the Cycle 24, and therefore it may not be true for the Cycle 25.

Hiremath (2008) predicted that the SSN was 110 at the Cycles 24 and 25 maxima, and the activity would grow as of the Cycle 26. Javaraiah (2008) found that Cycle 25 would be more powerful than Cycle 24. Nielsen and Kjeldsen (2011) concluded that Cycle 25 would

be weaker than Cycle 24. Using the Neuro-Fuzzy model, Attia, Ismail, and Basurah (2013) found that Cycle 25 would be weaker, and the maximum SSN would occur in 2022 with a peak of 90.7 ± 8.

Pishkalo (2008) made estimates for SC 24 and SC 25 by examining the relationship between the different parameters for SC 1 – 23. He stated that SC 24 will reach a maximum amplitude of 110.2 ± 33.4 in April – June 2012 and the next minimum will be in December 2018 – January 2019, SC 24 will be approximately 11.1 years and SC 25 will reach a maximum amplitude of 112.3 ± 33.4 in April – June 2023. Furthermore, Pishkalo (2014) estimated that SC 25 would be slightly stronger than SC 24 and the maximum SSN would be 105 – 110.

Sabarinath and Anilkumar (2018) predicted that SC 25 would be weaker than SC 24 and the maximum SSN would be 75 – 95. Li, Feng, and Li (2015) stated that the SC 25 has an SSN of 109.1 and will reach maximum in October 2023. Recently, Sarp *et al.* (2018) predicted SC 25 using a nonlinear approach. They predicted that the SC 25 maximum will occur in the year 2023.2 ± 1.1 with peak SSN of 157 ± 12.

In our study, it was estimated that the maximum in Solar Cycle 25 will be reached with peak SSN of 167.3 in July 2022 and Cycle 25 will last about ten years. This result means that Cycle 25 will be stronger than both Cycle 23 and Cycle 24. The result obtained is consistent with the studies of Pishkalo (2008), Rigozo *et al.* (2011), Pishkalo (2014), Sarp *et al.* (2018), and Okoh *et al.* (2018), since all conclude that Cycle 25 would be stronger than Cycle 24.

4. Conclusions

In the present study, the prediction of the change of solar activity over 11-year periods was performed by means of DL algorithms, which is a subfield of machine learning. Although there are studies related to machine learning in the literature for solar-cycle prediction, SSN predictions were made for the Solar Cycle 25 with the NNAR and LSTM deep-learning algorithms using the monthly mean values of SSN from 1749 to 2018. SSN estimates were also made with classical models such as ARIMA, Naive, Seasonal naïve, and drift. However, NNAR and LSTM deep-learning models were found to be more suitable for SSN estimation. In some of the studies on the prediction of Cycle 25, this cycle was found to have a smaller amplitude than Cycle 24, and some were larger than Cycle 24. In this study, it was predicted that Cycle 25 will have greater amplitude than the previous cycle. It was also predicted that the maximum amplitude of the Cycle 25 will be reached between 2022 and 2023 as in some previous studies (*e.g.* Kane, 2007; Rigozo *et al.*, 2011; Attia, Ismail, and Basurah, 2013; Sarp *et al.*, 2018) in the literature.

As previously stated, predicting changes in solar activity is of vital importance for all systems related to the Sun. Therefore, considering the increasing usage areas of machine learning, this study will be an important step for the more widespread use of machine learning in ionospheric studies.

Disclosure of Potential Conflicts of Interest The authors declare that they have no conflicts of interest.

References

Adhikari, R., Agrawal, R.K.: 2013, arXiv.

Aguirre, L.A., Letellier, C., Maquet, J.: 2008, *Solar Phys.* **249**, 103. DOI.
Atici, R.: 2018, *Astrophys. Space Sci.* **363**, 231. DOI.
Attia, A.-F., Ismail, H.A., Basurah, H.M.: 2013, *Astrophys. Space Sci.* **344**, 5. DOI.
Chattopadhyay, G., Chattopadhyay, S.: 2012, *Eur. Phys. J. Plus* **127**, 43. DOI.
Dancho, M.: 2018, blogs.rstudio.com/tensorflow/posts/2018-06-25-sunspots-lstm/.
Dikpati, M., De Toma, G., Gilman, P.A.: 2006, *Geophys. Res. Lett.* **33**, L05102. DOI.
Gkana, A., Zachilas, L.: 2015, *J. Eng. Sci. Technol. Rev.* **8**, 79.
Hiremath, K.: 2008, *Astrophys. Space Sci.* **314**, 45. DOI.
Javaraiah, J.: 2008, *Solar Phys.* **252**, 419. DOI.
Kane, R.: 2007, *Solar Phys.* **246**, 487. DOI.
Kilcik, A., Anderson, C., Rozelot, J., Ye, H., Sugihara, G., Ozguc, A.: 2009, *Astrophys. J.* **693**, 1173. DOI.
Kim, K.B., Kim, J.H., Chang, H.Y.: 2018, *J. Astron. Space Sci.* **35**, 151.
Layden, A., Fox, P., Howard, J., Sarajedini, A., Schatten, K., Sofia, S.: 1991, *Solar Phys.* **132**, 1. DOI.
Lewis, N.D.: 2016, *Deep Time Series Forecasting with Python: An Intuitive Introduction to Deep Learning for Applied Time Series Modeling*, Create Space Independent Publishing Platform, New York.
Li, K., Feng, W., Li, F.: 2015, *J. Atmos. Solar-Terr. Phys.* **135**, 72. DOI.
Maleki, A., Nasseri, S., Aminabad, M.S., Hadi, M.: 2018, *KSCE J. Civ. Eng.* **22**, 3233. DOI.
Nielsen, M.L., Kjeldsen, H.: 2011, *Solar Phys.* **270**, 385. DOI.
Okoh, D., Seemala, G., Rabiu, A., Uwamahoro, J., Habarulema, J., Aggarwal, M.: 2018, *Space Weather* **16**, 1424. DOI.
Penn, M.J., Livingston, W.: 2011, In: Choudhary, D.P., Strassmeier, K.G. (eds.) *The Physics of Sun and Star Spots*, *IAU Symp.* **273**, 126.
Pesnell, W.D.: 2008, *Solar Phys.* **252**, 209. DOI.
Petrovay, K.: 2010, *Living Rev. Solar Phys.* **7**, 6. DOI.
Pishkalo, M.: 2008, *Kinemat. Phys. Celest. Bodies* **24**, 242. DOI.
Pishkalo, M.: 2014, *Solar Phys.* **289**, 1815. DOI.
Quassim, M.S., Attia, A.-F., Elminir, H.K.: 2007, *Solar Phys.* **243**, 253. DOI.
Raissi, M., Karniadakis, G.E.: 2018, *J. Comput. Phys.* **357**, 125. DOI.
Rigozo, N., Echer, M.S., Evangelista, H., Nordemann, D., Echer, E.: 2011, *J. Atmos. Solar-Terr. Phys.* **73**, 1294. DOI.
Sabarinath, A., Anilkumar, A.: 2018, *J. Earth Syst. Sci.* **127**, 84. DOI.
Sagir, S., Atici, R., Ozcan, O.: 2018, *Pramāna* **91**, 54. DOI.
Sagir, S., Yesil, A.: 2018, *Wirel. Pers. Commun.* **102**, 31. DOI.
Sagir, S., Karatay, S., Atici, R., Yesil, A., Ozcan, O.: 2015, *Adv. Space Res.* **55**, 106. DOI.
Sarp, V., Kilcik, A., Yurchyshyn, V., Rozelot, J., Ozguc, A.: 2018, *Mon. Not. Roy. Astron. Soc.* **481**, 2981. DOI.
Sena, D., Nagwani, N.K.: 2016, *J. Eng. Appl. Sci.* **11**, 13123.
Shaikh, Y.H., Khan, A., Iqbal, M., Behere, S., Bagare, S.: 2008, *Fractals* **16**, 259. DOI.
Shumway, R.H., Stoffer, D.S.: 2011, *Time Series Analysis and Its Applications: With R Examples*, Springer, Berlin, 47.
Siami-Namini, S., Namin, A.S.: 2018, arXiv.
Sirignano, J., Spiliopoulos, K.: 2018, *J. Comput. Phys.* **375**, 1339. DOI.
Thompson, R.: 1993, *Solar Phys.* **148**, 383. DOI.
Zazo, R., Lozano-Diez, A., Gonzalez-Dominguez, J., Toledano, D.T., Gonzalez-Rodriguez, J.: 2016, *PLoS ONE* **11**, e0146917. DOI.