

Forecasting Multivariate Time Series of the Magnetic Field Parameters of the Solar Events

Khaznah Alshammari^{1,*}, Shah Muhammad Hamdi², Ali Ahsan Muhammad Muzahed¹ and Soukaina Filali Boubrahimi²

¹New Mexico State University, Las Cruces, NM, 88003, USA

²Utah State University, Logan, UT, 84322, USA

Abstract

Solar magnetic field parameters are frequently used by solar physicists in analyzing and predicting solar events (e.g., flares, coronal mass ejection, etc). Temporal observation of the magnetic field parameters, i.e., multivariate time series (MVTS) representation facilitates finding relationships of magnetic field states to the occurrence of the solar events. Forecasting MVTS of solar magnetic field parameters is the prediction of future magnetic field parameter values based on historic values of the past, regardless of the event labels. In this paper, we propose a deep sequence-to-sequence (seq2seq) learning approach based on batch normalization and Long-Short Term Memory (LSTM) network for MVTS forecasting of magnetic field parameters of the solar events. To the best of our knowledge, this is the first work that addresses the forecasting of magnetic field parameters rather than the classification of events based on MVTS representations of those parameters. The experimental results on a real-life MVTS-based solar event dataset demonstrate that our batch normalization-based model outperforms naive sequence models in forecasting performance.

Keywords

Multivariate time series Forecasting, Solar Physics, Solar Magnetic Field Parameters, LSTM, Batch Normalization

1. Introduction

Solar events are characterized by magnetic field parameter values on the solar corona such as helicity, flux, Lorentz force, etc. These magnetic field parameter values indicate the occurrence of extreme solar events such as solar flares, coronal mass ejection (CME), and eruption of solar energetic particles (SEP) [1]. These events are caused by a sudden burst of magnetic flux from the corona. The X-ray radiation of such extreme solar events can have devastating effects on life and infrastructure in space and ground such as disruption in GPS and radio communication, damage to electronic devices, and radiation exposure-based health risks to the astronauts. The cost associated due to infrastructure damage after extreme solar events can rise up to trillions of dollars [2].

In recent years, the prediction of solar events given a predefined time window has become an important challenge in the heliophysics community. Since the theoretical relationship between magnetic field influx and the occurrence of extreme events in solar active regions (AR) is not yet established, space weather researchers depend on the data of science-based approaches for predicting solar

events. The primary data source used in these efforts is the images captured by the Helioseismic Magnetic Imager (HMI) housed in the Solar Dynamics Observatory (SDO). HMI images (captured in near-continuous time) contain spatiotemporal magnetic field data of solar active regions. For performing temporal window-based flare prediction of an AR instance, the spatiotemporal magnetic field data of that region is mapped into a multivariate time series (MVTS) instance[3]. MVTS instances, collected with a uniform sampling rate throughout a present *observation period*, are labeled with multiple event classes (e.g., flare classes), and machine learning-based classifiers are trained with labeled MVTS instances to predict the occurrences of the events after a preset *prediction window*. Although multiple research efforts [4, 5, 6] addressed MVTS-based solar event prediction, forecasting of MVTS-represented magnetic field parameters is yet to be explored.

In this work, we aim to forecast the future values of the magnetic field parameters, given past values in the MVTS representations. In case of a sudden data gap, i.e., interruption in the communication between the satellite and ground receiver, MVTS forecasting of magnetic field parameters can play an important role in extrapolation. To the best of our knowledge, this is the first attempt to forecast the solar magnetic field parameters. We used a deep sequence-to-sequence learning model based on batch normalization and Long-Short Term Memory (LSTM) network that is trained with input-output pairs of examples, where the inputs are formed by sampling the MVTS instances for an observation window, and the outputs

AMLS'22: Workshop on Applied Machine Learning Methods for Time Series Forecasting, co-located with the 31st ACM International Conference on Information and Knowledge Management (CIKM), October 17-21, 2022, Atlanta, USA

✉ kalshamm@nmsu.edu (K. Alshammari); s.hamdi@usu.edu (S. M. Hamdi); muzahed@nmsu.edu (A. A. M. Muzahed); soukaina.boubrahimi@usu.edu (S. F. Boubrahimi)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

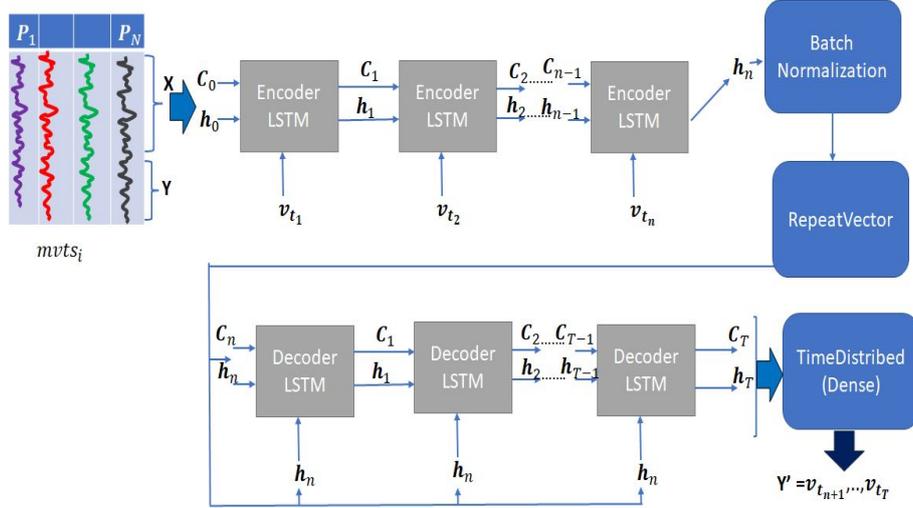


Figure 1: LSTM and Batch normalization-based seq2seq model for MVTS forecasting

are formed by sampling the MVTS instances for a prediction window (which follows the observation window). Our LSTM-based encoder-decoder model is trained with a backpropagation algorithm based on mini-batch gradient descent-based optimization for minimizing Mean Squared Error (MSE) between the observed MVTS (input) and predicted MVTS (output).

2. Related Work

Recent research efforts on solar event prediction are mostly based on data science. Data-driven extreme solar event prediction models stem from linear and nonlinear statistics. Datasets used in these models were collected from line-of-sight magnetogram and vector magnetogram data. Line-of-sight magnetogram contains only the line-of-sight component of the magnetic field, while vector magnetogram contains the full disk magnetic field data [7]. NASA launched Solar Dynamics Observatory (SDO) in 2010. Since then, SDO's instrument Helioseismic and Magnetic Imager (HMI) has been mapping the full-disk vector magnetic field every 12 minutes [1]. Most of the recent prediction models use the near-continuous stream of vector magnetogram data found from SDO [8]. Magnetic field parameters (e.g., helicity, flux, etc) were developed with the goal of finding a relationship between the phosphoric magnetic field behavior and solar activity, which usually occurs in the solar chromosphere and transition region of the solar corona.

Deep learning-based sequence-to-sequence models using Long Short Term Memory (LSTM), Recurrent Neural Network (RNN), and Gated Recurrent Unit (GRU)

have been used successfully in multiple Natural Language Processing (NLP) tasks such as machine translation [9, 10] and text summarization [11, 12]. Since multivariate time series are high-dimensional sequence data, previously MVTS forecasting has been addressed by different seq2seq models [13, 14]. In [15], batch normalization has shown promising improvements in the sentiment classification task, where a batch-normalized variant of LSTM architecture is used and each LSTM cell's input, hidden state, and cell state are normalized during training. Being inspired by encoder-decoder-based machine translation models, in this work we considered the MVTS forecasting of solar magnetic field parameters as a sequence-to-sequence learning task, and used batch normalization-based LSTM architecture for capturing long-term dependencies of multi-dimensional sequence data.

3. Methodology

3.1. Notations

Each solar active region results in different event occurrences after a given prediction window represents an event instance. The event instance i is represented by a MVTS instance $mvts_i$. The MVTS instance $mvts_i \in \mathbb{R}^{T \times N}$ is a collection of individual time series of N magnetic field parameters, where each time series contains periodic observation values of the corresponding parameter for an observation period T . In the MVTS instance $mvts_i = \{v_{t_1}, v_{t_2}, \dots, v_{t_T}\}$, $v_{t_i} \in \mathbb{R}^N$ represents a timestamp vector. We divide the dataset into (X, Y) pairs, where $X_i = mvts_i[t_1 : t_{obs}, :] \in \mathbb{R}^{t_{obs} \times N}$,

$Y_i = mvts_i[t_{obs+1} : t_T, :] \in \mathbb{R}^{t_{pred} \times N}$, t_{obs} is the observation time, and t_{pred} is the prediction time.

3.2. LSTM and Batch Normalization-based MVTS Forecasting

In this section, we present a batch normalization-based implementation of the encoder-decoder model that uses LSTM architecture and compare it with other baseline sequence models of naive stochastic gradient descent implementation (without batch normalization). There are different deep sequence learning models, which are frequently applied in machine translation, and they can be adapted for time series forecasting. In this study, we analyze two seq2seq models: the batch normalization-based seq2seq LSTM Model (BN seq2seq LSTM), and the seq2seq models based on LSTM/GRU/RNN, and compare their forecasting results.

Fig. 1 depicts our seq2seq-based model that uses batch normalization and LSTM architecture. First, in the encoder LSTM cells, the value of each time step is used as input to the encoder LSTM cell together with the previous cell state c and hidden state h , the process repeats until the last cell state c and hidden state h are generated. Then, the decoder LSTM cell uses the last cell state c and hidden state h from the encoder as the initial states for the decoder LSTM cell. The last hidden state of the encoder is also copied t_{pred} times using a *Repeat Vector* layer according to the length of the forecasting window, and each copy is inputted into the decoder LSTM cell together with the previous cell state c and hidden state h . The decoder outputs hidden states for all the t_{pred} time steps and the hidden states are connected to the final *Time-distributed-dense* layer in order to produce the final output sequence. The time-distributed-dense layer allows to apply a dense layer to every temporal slice of the input. We use this final layer to process the output from the LSTM hidden layer. Every input shape is three-dimensional, and the first dimension of the input is considered to be the temporal dimension. This means that we need to configure the last LSTM layer prior to the time-distributed-dense layer to return output sequences. The output shape will be three-dimensional as well, which means that if the time-distributed-dense layer is the output layer, then for predicting a sequence we need to reshape the final representation into a three-dimensional shape [16]. In the batch normalization-based seq2seq LSTM Model, we use mini-batches to feed the data into the model. Batch normalization is a useful method for making deep neural network training faster and more robust, and it normalizes the input activations to avoid gradient explosion caused by the activation function ELU (Exponential Linear Unit) in the encoder [17]. The batch normalization layer applies a transformation that maintains the mean output close to 0 and the output standard deviation close

to 1. We found batch normalization to be significant in maximizing the performance of MVTS forecasting for the magnetic field parameters of the solar events, which we demonstrate in more detail in the experiments section.

3.3. Evaluation Metrics

We used Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) to report our model results. The evaluation metrics (MAE, MSE, and RMSE) measure the amount of error in statistical models. They assess the average squared difference between the observed and predicted values.

Mean Absolute Error (MAE) is the average over the absolute values of the differences between predicted representations and ground truth representations.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

where y_i is the ground truth value and \hat{y}_i is the predicted value.

Mean Squared Error (MSE) is defined as the mean or average of the square of the difference between actual and predicted values.

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

Root Mean Squared Error (RMSE) is the difference between forecast and corresponding observed values, where each difference is squared and averaged over the sample space. It denotes the square root of the MSE.

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

Experiments

We compared the batch normalization-based seq2seq LSTM model with the baseline models on multivariate time series forecasting of magnetic field parameters of a solar events dataset. The source code of our model and the experimental dataset are available on our GitHub repository¹.

3.4. Dataset Description

As the benchmark dataset of our experiments, we used the MVTS-based solar flare prediction data set published by Angryk et al [3]. Each MVTS instance in the dataset is made up of 25 time series of active region magnetic field parameters (a full list can be found in [1]). The time series instances are recorded at 12 minutes intervals for a

¹https://github.com/Kalshammari/BN_Seq2Seq

Table 1

Forecasting Performance of Batch Normalization-based seq2seq (LSTM) Model compared to the baselines

Performance Metrics	Gradient Descent LSTM	Gradient Descent GRU	Gradient Descent RNN	BN seq2seq LSTM
Train MAE	14.481 \pm 0.043	14.942 \pm 0.052	15.578 \pm 0.036	0.094 \pm 0.002
Test MAE	14.55 \pm 0.103	15.042 \pm 0.092	15.68 \pm 0.107	0.057 \pm 0.010
Train MSE	18.238 \pm 0.075	19.631 \pm 0.062	21.269 \pm 0.031	0.070 \pm 0.003
Test MSE	22.598 \pm 0.251	24.906 \pm 0.821	24.589 \pm 0.726	0.002 \pm 0.001
Train RMSE	18.434 \pm 0.039	19.126 \pm 0.652	19.921 \pm 0.821	0.265 \pm 0.007
Test RMSE	18.492 \pm 0.348	19.245 \pm 0.542	20.092 \pm 0.672	0.005 \pm 0.001

total duration of 12 hours (60-time steps). The dataset has the number of observation points $T = 60$, and the number of dimensions in timestamp vectors $N = 25$, while the event occurrence window is 12 hours. Our experimental dataset consists of 1,540 MVTS instances that are evenly distributed across four flare classes (X, M, BC, and Q). We discarded the class labels to fit the dataset for MVTS forecasting [5, 4], where each MVTS instance is divided into input and output (ground truth) sequences according to the observation window (t_{obs}) and prediction window (t_{pred}). In our experiments, $t_{obs} = 40$, and $t_{pred} = 20$, while $T = t_{obs} + t_{pred}$.

3.5. Train/test splitting method

We performed random sampling for train/test splitting, where we use the stratified holdout method (80 % for training, and 20 % for testing) with six different random seeds, and reported the mean error rates along with standard deviation. Train and test datasets are z-normalized since magnetic field parameter values appear on different scales. The shapes of train and test datasets are as follows.

- X_train shape:(1232, 40, 25) and y_train shape:(1232, 20, 25)
- X_test shape:(308, 40, 25) and y_test shape:(308, 20, 25)

3.6. Baseline Models

We evaluated our model with LSTM, RNN, and GRU-based seq2seq implementations. In the forward pass, we have input the first t_{obs} vectors of each MVTS to the encoder cells (LSTM/RNN/GRU) to produce the encoded hidden state. That encoded hidden state is the input to the decoder cells of the same type. The decoder then predicts the next 25-dimensional timestamp vectors for each timestamp in t_{pred} and matches the prediction with ground truth to perform stochastic gradient descent-based backpropagation. In all three models, the number of dimensions in cell state and hidden state representa-

tions was 25, the number of epochs in training was 5, and the learning rate in stochastic gradient descent is 0.01.

3.7. Performance of LSTM and Batch Normalization-based seq2seq model

When we apply LSTM and batch normalization-based seq2seq model, we perform the following steps. First, we extract (X, Y) pairs from all 1,540 MVTS instances, where the length of each example X is $t_{obs} = 40$, the length of each output Y is $t_{pred} = 20$, and each timestamp vector is 25-dimensional.

In the encoder step, the input is of size $(b, 40, 25)$, where $b (= 10)$ is the batch size of the MVTS instances. For each encoder LSTM cell, the vector of each time step is used as the input to the encoder LSTM cell together with the previous cell state c and hidden state h , and the process repeats until the last cell state c and hidden state h are generated. The decoder LSTM cell uses the last cell state c and hidden state h from the encoder as the initial states for the decoder LSTM cell. The last hidden state of the encoder is also copied 20 times using the Repeat Vector layer and each copy is inputted into the decoder LSTM cell together with the previous cell state c and hidden state h . The decoder outputs a hidden state for all the 20-time steps, and these hidden states are connected to a time-distributed-dense layer to generate the final forecasting output which is of size $(b, 20, 25)$. We used Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) to report our model performance results. We reported the mean and standard deviation of the performance measures results in Table 1. We found that our approach of deep sequence-to-sequence learning based on batch normalization and Long-Short Term Memory (LSTM) network significantly outperformed the baseline methods' results as Table 1 shows. It is visible that batch normalization makes a difference of a large margin by producing errors near 0, whereas the traditional seq2seq models result in large error values due to the absence of batch normalization.

4. Conclusion

We propose a batch normalization-based deep seq2seq model for multivariate time series forecasting of magnetic field parameters of solar events. Unlike previous works of MVTS-based event classification, we perform forecasting of magnetic field parameter values irrespective of MVTS labels. We compare it with other seq2seq implementations based on LSTM, GRU, and RNN. Our proposed approach significantly improved the MAE, MSE, and RMSE results of MVTS forecasting on a benchmark solar magnetic field parameter dataset.

For future research, we plan to develop machine learning models for MVTS forecasting that leverage MVTS labels. We aim to use the forecasting models for augmenting (creating synthetic examples) MVTS instances of minority classes (rare events). In addition, to utilize inter-variable dependencies of the MVTS instances for the task of forecasting, we plan to incorporate graph construction (e.g., functional network computation from the correlation matrices of the MVTS instances) and graph neural network (GNN)-based representation learning.

Acknowledgments

This project has been supported in part by funding from CISE and GEO directorates under NSF awards #2153379 and #2204363.

References

- [1] M. G. Bobra, S. Couvidat, Solar flare prediction using sdo/hmi vector magnetic field data with a machine-learning algorithm, *The Astrophysical Journal* 798 (2015) 135.
- [2] J. P. Eastwood, E. Biffis, M. A. Hapgood, L. Green, M. M. Bisi, R. D. Bentley, R. Wicks, L.-A. McKinnell, M. Gibbs, C. Burnett, The economic impact of space weather: Where do we stand?, *Risk Analysis* 37 (2017) 206–218.
- [3] R. A. Angryk, P. C. Martens, B. Aydin, D. Kempton, S. S. Mahajan, S. Basodi, A. Ahmadzadeh, X. Cai, S. Filali Boubrahimi, S. M. Hamdi, et al., Multivariate time series dataset for space weather data analytics, *Scientific data* 7 (2020) 1–13.
- [4] S. M. Hamdi, D. Kempton, R. Ma, S. F. Boubrahimi, R. A. Angryk, A time series classification-based approach for solar flare prediction, in: 2017 IEEE Intl. Conf. on Big Data (Big Data), IEEE, 2017, pp. 2543–2551.
- [5] A. Muzahed, S. M. Hamdi, S. F. Boubrahimi, Sequence model-based end-to-end solar flare classification from multivariate time series data, in: 20th IEEE Intl. Conf. on Machine Learning and Applications, ICMLA 2021, Pasadena, CA, USA, December 13–16, 2021, IEEE, 2021, pp. 435–440.
- [6] R. Ma, S. F. Boubrahimi, S. M. Hamdi, R. A. Angryk, Solar flare prediction using multivariate time series decision trees, in: 2017 IEEE Intl. Conf. on Big Data, BigData 2017, Boston, MA, USA, December 11–14, 2017, IEEE Computer Society, 2017, pp. 2569–2578.
- [7] S. F. Boubrahimi, B. Aydin, D. Kempton, R. Angryk, Spatio-temporal interpolation methods for solar events metadata, in: 2016 IEEE Intl. Conf. on Big Data (Big Data), IEEE, 2016, pp. 3149–3157.
- [8] J. P. Mason, J. Hoeksema, Testing automated solar flare forecasting with 13 years of michelson doppler imager magnetograms, *The Astrophysical Journal* 723 (2010) 634.
- [9] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473 (2014).
- [10] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
- [11] M. Yousefi-Azar, L. Hamey, Text summarization using unsupervised deep learning, *Expert Systems with Applications* 68 (2017) 93–105.
- [12] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, *OpenAI blog* 1 (2019) 9.
- [13] L. Liu, A. M. Finch, M. Utiyama, E. Sumita, Agreement on target-bidirectional lstms for sequence-to-sequence learning, in: Proc. of the AAAI Conf. on Artificial Intelligence, February 12–17, 2016, Phoenix, Arizona, USA, AAAI Press, 2016, pp. 2630–2637.
- [14] P. H. Scherrer, J. Schou, R. Bush, A. Kosovichev, R. Bogart, J. Hoeksema, Y. Liu, T. Duvall, J. Zhao, C. Schrijver, et al., The helioseismic and magnetic imager (hmi) investigation for the solar dynamics observatory (sdo), *Solar Physics* 275 (2012) 207–227.
- [15] H. Margarit, R. Subramaniam, A batch-normalized recurrent network for sentiment classification, *Advances in Neural Information Processing Systems* (2016) 2–8.
- [16] J. Brownlee, Long short-term memory networks with python: develop sequence prediction models with deep learning, *Machine Learning Mastery*, 2017.
- [17] S. Santurkar, D. Tsipras, A. Ilyas, A. Madry, How does batch normalization help optimization?, *Advances in neural information processing systems* 31 (2018).