2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA) | 978-1-6654-4337-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICMLA52953.2021.00074

# Sequence Model-based End-to-End Solar Flare Classification from Multivariate Time Series Data

Ali Ahsan Muhummad Muzaheed*, Shah Muhammad Hamdi*, Soukaïna Filali Boubrahimi†
*Department of Computer Science, New Mexico State University, Las Cruces, NM 88003, USA
†Department of Computer Science, Utah State University, Logan, UT 84322, USA
Email: {muzaheed, shamdi1}@nmsu.edu, soukaina.boubrahimi@usu.edu

***Abstract*—Machine learning-based prediction of solar flares has become an important application of data science in space weather research. Spatiotemporal magnetic field data of solar active regions captured by solar imaging observatories are mapped into multivariate time series data to facilitate temporal window-based solar flare prediction. Existing methods of solar flare prediction leveraging multivariate time series data rely heavily on statistical features for the representation learning of individual time series instances. In this work, we used Deep Learning, more specifically Long Short Term Memory (LSTM) Networks for learning representations of multivariate time series instances that map into multiple flare classes. This work enables the end-to-end multivariate time series classification bypassing the requirements of hand-engineered features. Our experiments on a real-life solar flare dataset show better prediction performance in comparison with state-of-the-art baseline methods.**

***Index Terms*—Flare prediction; Time series; LSTM**

## I. Introduction

Solar flares are considered to be one of the most intense solar events caused by a sudden burst of magnetic flux from the solar surface. Ultraviolet and X-ray radiation of solar flares can have disastrous effects on life and infrastructure in both space and ground. From the radiation exposure-based health risks of the astronauts to the disruption in GPS and radio communication and damages in electronic devices, costs of replacement/repairing of infrastructure after major flaring events can rise up to trillions of dollars [1].

Due to the potentially hazardous impacts of the solar flares, the prediction of solar flares given a predefined time window has become a hot research topic in the heliophysics community in recent years. Since the theoretical relationship between magnetic field influx and flare occurrence in solar active regions (AR) is not yet established, space weather researchers rely on data science-based approaches for predicting solar flares. The primary data source used in these efforts is the images captured by the Helioseismic Magnetic Imager (HMI) housed in 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 [2]. The variables in the MVTS instance consist of solar magnetic field parameters. The time series represented by the magnetic field parameters are extracted based on two time windows: prediction window (the time window before which the flare happens), and observation window (the time window during which the AR parameter values are calculated). Each MVTS instance is labeled as one of six classes - Q, A, B, C, M, and X, where Q represents flare quiet active regions, and other labels represent flaring events with increasing intensity. In Fig. 1, we show the data model of a labeled MVTS instance.

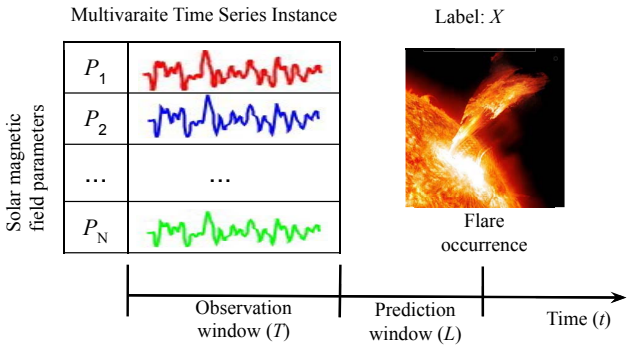



Fig. 1: Multivaraite time series instance with predefined observation and prediction window, and flare class label

Modeling flare prediction as multivariate time series classification resulted in higher accuracy in comparison with the single timestamp-based magnetic field vector classification models [3]. MVTS classification-based flare prediction was performed in two steps. Firstly, MVTS representations were learned by the summarization of the individual time series through predefined statistical features. This step embeds the MVTS instances into a low-dimensional space. Secondly, traditional classifiers are trained by the MVTS embeddings. The two-step process of MVTS classification relies heavily on hand-engineering of statistical features and choice of downstream classifiers, which eventually complicates the application of these models in datasets with varying properties.

Therefore, in this work, we propose an end-to-end MVTS classification approach leveraging deep learning-based sequence models. We use Long Short Term Memory (LSTM) networks for learning the representations of the MVTS instances. We train the model by sequentially feeding vectors representing magnetic field parameters into LSTM cells, and optimize the cell weights through gradient descent and back-propagation. Our model does not rely on hand-engineered

978-1-6654-4337-1/21/$31.00 ©2021 IEEE
DOI 10.1109/ICMLA52953.2021.00074

statistical features for learning low-dimensional representations of the individual time series, and incorporates automated feature learning for flare prediction.

The contributions made by this paper are listed below.

1) Representing MVTS instances as temporal sequences of high-dimensional vectors.
2) Replacing current *embedding followed by classification* methods by an end-to-end classification method by using sequence modeling.
3) Experimentally demonstrating the better performance of our model in comparison with state-of-the-art baselines on a benchmark solar flare prediction dataset.

The rest of the paper is organized as follows. In section II, we discuss the related work. We present our sequence model-based flare classification model at section III. In section IV, we present the experimental findings. Finally, we discuss our future work and conclude the paper in section V.

## II. Related Work

*Theo* is one of the earliest flare prediction systems [4]. It was an expert system that required human inputs. To predict different flare classes, it used a set of sunspots and magnetic field properties. In 1987, the Space Environment Center (SEC) of the National Oceanic and Atmospheric Administration (NOAA) adopted Theo for rule-based flare prediction.

Most of the later research efforts of flare prediction were based on data science. Data-driven flare prediction models stemmed from both 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 [5]. 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 [6]. Most of the recent flare prediction models use the near-continuous stream of vector magnetogram data found from SDO.

Linear statistical models aimed at finding the AR magnetic properties that are highly correlated with the occurrences of flares. Cui et al. [7] and Jing et al. [8] studied flare correlations with the line-of-sight magnetogram-based active region features. Before the launch of SDO, Leka and Barnes [9] collected vector magnetogram data from Mees Solar Observatory on the summit of Mount Haleakala, and used linear discriminant analysis (LDA) for flare classification.

Nonlinear statistical models are mostly machine learning-based classifiers. On line-of-sight magnetogram-based AR datasets, Yu et al. [10] used C4.5 decision tree, Song et al. [11] used logistic regression, Ahmed et al. [12] used the artificial neural network, and Al-Ghraibah et al. [13] used relevance vector machine as classification models. Bobra et al. [14] used Support Vector Machine (SVM) on the AR parameters derived from SDO-based vector magnetograms. Nishizuka et al. [15] used both line-of-sight and vector magnetograms and compared the performance of three classifiers - k-NN, SVM, and Extremely Randomized Tree (ERT).

Temporal window-based flare prediction was introduced by Angryk et al. [2], which extends the earlier single timestamp-based models. They published a curated and labeled multivariate time series dataset for predicting flares that might appear in an active region after a given period of time. This dataset contains 33 vector magnetogram parameter values recorded in 12 minutes cadence to create 12 hours observation window-based multivariate time series instances. Each multivariate time series instance is labeled as one of five flare classes. Hamdi et al. [3] used statistical summarization of individual time series for computing low dimensional representations of MVTS instances, and applied k nearest neighbors, logistic regression, and decision tree classifiers for binary classification of intense flares (M and X classes), and less intense flares or flare quiet active regions (Q, A, B, and C classes). Ma et. al. [16] introduced multivariate time series decision trees that approached the flare forecasting problem using clustering as a preprocessing step. Among other solar events classification, MVTS representation was used for Solar Energetic Particle (SEP) prediction [17].

Our work introduces an end-to-end learning paradigm for classifying MVTS instances of different flare classes. While our model is not limited to hand-engineered statistical features, it can achieve better prediction performance through deep learning-based sequence learning models.

## III. Sequence model-based Flare Classification

### A. Notations and Preliminaries

Each solar active region resulting in different flare classes (or staying as a flare quiet region) after a given prediction window represents a solar event. The solar event $i$ is represented by a multivariate time series instance $mvts_i$, and associated by a class label $y_i$. The class label $y_i$ represents the flare quiet state, or flare classes of different intensities. The multivariate time series 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 $T$. We denote the vector of $t$-th timestamp as $x^{<t>} \in \mathbb{R}^N$, and the time series represented by $j$-th parameter as $P_j \in \mathbb{R}^T$. After the observation period $T$ and prediction period $L$, the event is labeled by the active region state (flare quiet or different flare classes). The active region state of a particular timestamp is found from the NOAA records of flaring events. Fig. 1 shows the data model of a MVTS-based solar event.

### B. Data Preprocessing

Since the magnetic field parameter values are recorded in different scales, we perform z-score normalization of each individual time series of each MVTS instance. At $mvts_i$, parameter-based individual time series are denoted by $P_1, P_2, \ldots, P_N$. For each individual time series $P_j$, we perform z-normalization as follows.

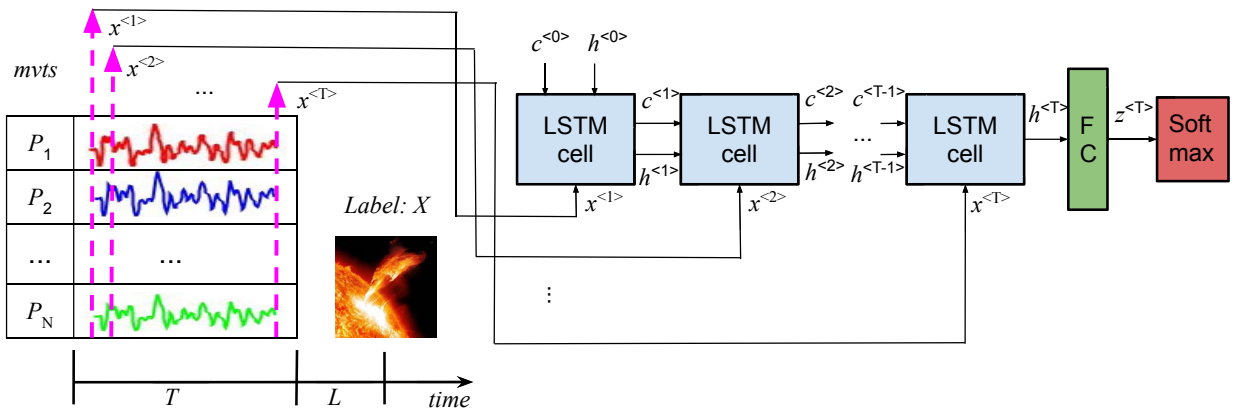



Fig. 2: Representation learning of an MVTS instance through LSTM, fully connected (FC), and softmax layers

$$x_k^{(j)} = \frac{x_k^{(j)} - \mu^{(j)}}{\sigma^{(j)}}$$

Here, $x_k^{(j)}$ is the $k$-th value of the time series $P_j$, where $1 \leq k \leq T$, $\mu^{(j)}$ is the mean of time series $P_j$, and $\sigma^{(j)}$ is the standard deviation of the time series $P_j$.

### C. LSTM-based MVTS classification

Instead of considering a single MVTS instance as a collection of univariate time series instances, we consider it a sequence of high-dimensional timestamp vectors. The timestamp vector $x^{<t>} \in \mathbb{R}^N$ represents the magnetic filed state of the active region ($N$ parameter vales) in the timestamp $t$. In the forward pass, at each timestamp, the timestamp vector $x^{<t>}$ is input to the LSTM cell. The last output (hidden representation) is input to a fully connected layer for getting a low-dimensional embedding, which is finally fed into a softmax layer (representing number of classes).

In Fig. 2, input sequence to LSTM cell is $x^{<1>}, x^{<2>}, x^{<3>}, ..., x^{<T>}$, cell state representations are $c^{<0>}, c^{<1>}, c^{<2>}, ..., c^{<T-1>}$, and hidden state representations are $h^{<0>}, h^{<1>}, h^{<2>}, ..., h^{<T>}$. After randomly initializing $c^{<0>}$ and $h^{<0>}$, we update the cell state and hidden state of the timestamp $t$ by following LSTM equations [18].

$$\begin{aligned}
\tilde{c}^{<t>} &= tanh(W_c[h^{<t-1>}, x^{<t>}] + b_c) \\
\Gamma_u &= \sigma(W_u[h^{<t-1>}, x^{<t>}] + b_u) \\
\Gamma_f &= \sigma(W_f[h^{<t-1>}, x^{<t>}] + b_f) \\
\Gamma_o &= \sigma(W_o[h^{<t-1>}, x^{<t>}] + b_o) \\
c^{<t>} &= \Gamma_u \odot \tilde{c}^{<t>} + \Gamma_f \odot c^{<t-1>} \\
h^{<t>} &= \Gamma_o \odot tanh(c^{<t>})
\end{aligned}$$

We denote number of dimensions of the cell state representation $c^{<t>}$ and hidden state representation $h^{<t>}$ of the LSTM cell as $N_h$. The concatenation of hidden state of previous timestamp and the input of current timestamp is $[h^{<t-1>}, x^{<t>}] \in \mathbb{R}^{N_h+N}$. The candidate cell state representation is $\tilde{c}^{<t>} \in \mathbb{R}^{N_h}$. The weight matrices are $W_c, W_u, W_f, W_o \in \mathbb{R}^{N_h \times (N_h+N)}$, and bias terms are $b_c, b_u, b_f, b_o \in \mathbb{R}$. Subscripts $u$, $f$, and $o$ represents the activations of update gate, forget gate, and output gate respectively, while $\odot$ refers to elementwise multiplication, and $\sigma$ represents sigmoid activation.

We consider $h^{<T>}$ as the final representation of the input MVTS, which we further input into a linear (fully connected) layer. In this layer, $W_l \in \mathbb{R}^{C \times N_h}$, and $b_l \in \mathbb{R}$, where $C$ is the number of classes. After this layer, we have a $C$-dimensional representation of the MVTS instance.

$$z^{<t>} = ReLU(W_l h^{<T>} + b_l)$$

Finally, we input $z^{<t>}$ into a softmax layer, whose number of units is equal to the number of classes. The softmax layer gives us the normalized class probabilities, and we finally get $\hat{y}^{<t>} \in \mathbb{R}^C$.

$$\hat{y}^{<t>} = \frac{e^{z^{<t>}}}{\sum_{j=1}^{C} e^{z_j^{<t>}}}$$

The predicted labels of training MVTS instances are matched against true labels, and Stochastic Gradient Descent-based optimizer updates the weight and bias parameter values of LSTM cell and the fully connected layer through Back-propagation algorithm.

Other sequence models such as Recurrent Neural Network (RNN) [19], and Gated Recurrent Unit (GRU) [20] can be easily integrated in our model by simply replacing the LSTM cells with RNN or GRU cells. We prefer LSTM cells for its comparative efficiency in handling long range dependencies in time series data.

## IV. Experiments

In this section, we demonstrate our experimental findings. We compared the multiclass classification performance of our model with three other baselines on a benchmark MVTS-based

TABLE I: Multiclass classification performance of the proposed method with the baselines

| Measures | *FLT* | *LTV* | *TS-SUM* | *RNN* | *LSTM* |
|---|---|---|---|---|---|
| Accuracy | $0.259 \pm 0.012$ | $0.323 \pm 0.02$ | $0.609 \pm 0.091$ | $0.427 \pm 0.025$ | $\mathbf{0.628 \pm 0.03}$ |
| Precision (X) | $0.232 \pm 0.024$ | $0.342 \pm 0.041$ | $0.712 \pm 0.054$ | $0.534 \pm 0.031$ | $\mathbf{0.757 \pm 0.028}$ |
| Recall (X) | $0.264 \pm 0.053$ | $0.392 \pm 0.043$ | $0.772 \pm 0.024$ | $0.631 \pm 0.028$ | $\mathbf{0.947 \pm 0.023}$ |
| F1 (X) | $0.244 \pm 0.032$ | $0.362 \pm 0.04$ | $0.741 \pm 0.034$ | $0.582 \pm 0.019$ | $\mathbf{0.841 \pm 0.014}$ |
| Precision (M) | $0.254 \pm 0.012$ | $0.324 \pm 0.033$ | $0.522 \pm 0.031$ | $0.411 \pm 0.014$ | $\mathbf{0.594 \pm 0.018}$ |
| Recall (M) | $0.26 \pm 0.023$ | $0.331 \pm 0.061$ | $\mathbf{0.552 \pm 0.022}$ | $0.402 \pm 0.03$ | $0.544 \pm 0.014$ |
| F1 (M) | $0.257 \pm 0.026$ | $0.327 \pm 0.042$ | $0.537 \pm 0.023$ | $0.406 \pm 0.029$ | $\mathbf{0.568 \pm 0.02}$ |
| Precision (BC) | $0.232 \pm 0.044$ | $0.263 \pm 0.024$ | $0.453 \pm 0.033$ | $0.282 \pm 0.031$ | $\mathbf{0.495 \pm 0.013}$ |
| Recall (BC) | $0.241 \pm 0.053$ | $0.212 \pm 0.02$ | $\mathbf{0.472 \pm 0.014}$ | $0.261 \pm 0.021$ | $0.409 \pm 0.023$ |
| F1 (BC) | $0.236 \pm 0.041$ | $0.234 \pm 0.024$ | $0.462 \pm 0.041$ | $0.271 \pm 0.031$ | $\mathbf{0.448 \pm 0.031}$ |
| Precision (Q) | $0.324 \pm 0.034$ | $0.343 \pm 0.044$ | $0.583 \pm 0.045$ | $0.483 \pm 0.024$ | $\mathbf{0.603 \pm 0.024}$ |
| Recall (Q) | $0.251 \pm 0.042$ | $0.362 \pm 0.071$ | $0.663 \pm 0.034$ | $0.413 \pm 0.042$ | $\mathbf{0.683 \pm 0.023}$ |
| F1 (Q) | $0.282 \pm 0.014$ | $0.352 \pm 0.013$ | $0.62 \pm 0.043$ | $0.445 \pm 0.032$ | $\mathbf{0.64 \pm 0.024}$ |

solar flare prediction dataset. We used PyTorch 1.7.1 with CUDA 11.3 for implementing our LSTM-based MVTS classifier. We conducted all of our experiments in a Windows server having a single processor (AMD Ryzen 7 with $3700 \times 3600$ MHz speed, 8 cores, and 16 threads), one NVIDIA GeForce RTX 3070 GPU (5888 CUDA cores), and 32 GB memory. The source code of our model and the experimental dataset are available at our GitHub repository. [1]

### A. Dataset description

As the benchmark dataset of our experiments, we used the solar flare prediction dataset published by the Data Mining Lab of Georgia State University [2]. Each MVTS instance in the dataset is made up of 33 individual time series of active region magnetic field parameters. The time series instances are recorded at 12 minutes intervals for a total duration of 12 hours (60 time steps). The MVTS instances are labeled according to the largest solar flare that occurred after 12 hours (five different classes: X, M, C, B, and Q). Therefore, the dataset has the number of the observation points $T = 60$, and the number of dimensions in timestamp vectors $N = 33$, while the prediction window is $L = 12$ hours.

Since solar flares are rare events, the original data release (with 4,098 MVTS instances of X, M, B, C, and Q classes) suffers from high class imbalance problem, e.g., there are much fewer "X" class (extreme flares) instances in comparison with "Q" class (no flares) instances. To remove this class imbalance, we undersampled the dataset, and merged B and C classes (less intense flaring events) into a single class (BC). Our experimental dataset consists of 1,354 instances evenly distributed across four classes (X, M, BC, and Q).

### B. Baseline methods

We evaluated our LSTM-based MVTS classification model with four other baselines.

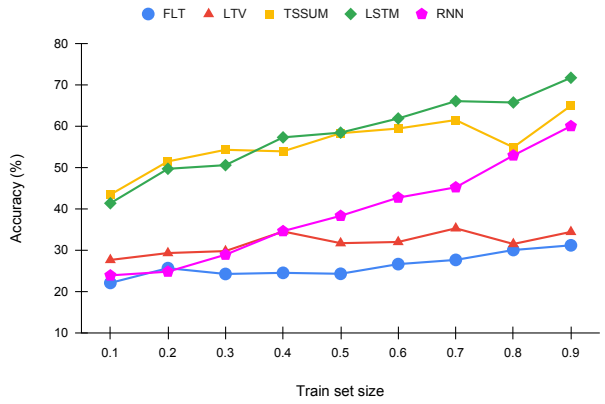
- **Flattened vector method (FLT)**: This is a naive method, where each $60 \times 33$ MVTS instance is flattened into a $1,980$-dimensional vector.
- **Vector of last timestamp (LTV)**: This method was introduced by Bobra et al [14], where vector magnetogram data (feature space of all magnetic field parameters) were used for classification. Since the last timestamp of the MVTS is temporally nearest to the flaring event, we sampled the vector of the last timestamp (33-dimensional) to train the downstream classifier.
- **Time series summarization-based MVTS representation (TS-SUM)**: This method, proposed by Hamdi et al [3] summarizes each individual time series of length $T$ by eight statistical features: mean, standard deviation, skewness, and kurtosis of the original time series, and the first-order derivative of the time series. As a result, we get a $8 \times 33$-dimensional vector space, which is used for classifier training and test.
- **Recurrent Neural Network (RNN):** As the fourth baseline, we replace LSTM cells of our model (Fig. 2) with standard RNN cells, where the number of hidden dimensions is 128, and number of training epochs is 500.

The first three baselines are *embedding followed by classification* methods. After performing the embedding of MVTS instances using the aforementioned methods, we use logistic regression classifier with L2 regularization. In all experiments, we split the dataset into train and test using stratified holdout method (two-third for training and one-third for test). In LSTM model, 128 is used as the number of dimensions in cell state and hidden state representations, number of epochs in training is 500, and learning rate in stochastic gradient decent is 0.01.

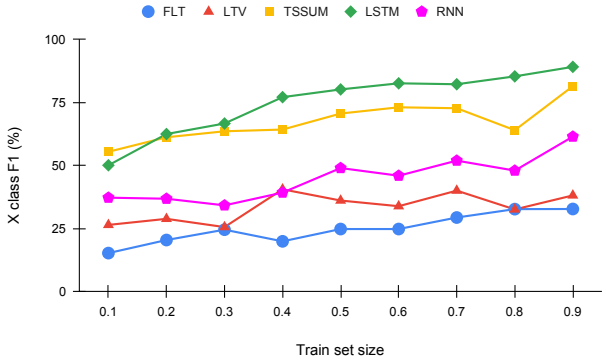
### C. Multiclass classification performance

In Table I, we show the classification performances of our LSTM-based MVTS classifier along with that of the baseline methods. For a comprehensive classification report, we show accuracy along with precision, recall, and F1 of each class. We performed five experiments with different train/test sets sampled by stratified holdout and reported the mean and standard deviation of the experiments. From the results, it is visible that the LSTM-based MVTS classification model outperforms all other baselines in almost all measures. Although the time series summarization-based method classifies flare classes almost in similar accuracy to our model, with datasets of more instances and experiments of longer training time, our model can perform even better.

[1] https://github.com/ahsan-muzaheed/SolarFlareClassification2021




(a) Multiclass classification accuracy with increasing training data




(b) F1 of X class with increasing training data

Fig. 3: Multiclass classification with varying training set size.

### D. Classification varying training set size

To verify the adaptability of our model with bigger training datasets, we experimented by varying the training set size. We varied the training set size from 10% to 90% of the dataset size, while tested the models with the rest of the instances (Fig. 3). We performed stratified train/test sampling with a given training set size, and evaluated the classification performance of the classifiers five times with five distinct training and test sets. In Fig. 3a and 3b, we plotted the mean accuracy values and mean F1 (X class) values found in all runs of different train/test samples with different preset training data size. Although all models become more accurate with the gradual increase of training set size, we observe more consistent and steep increasing patterns in LSTM and RNN models. It proves that with sufficiently large datasets, deep learning models can outperform the traditional classifiers or embedding methods in a larger margin. The time series summarization-based method shows promising performance throughout the experiments, but the generalization capability of this model can be limited in a more complex dataset due to its less flexible learning methodology. In comparison with deep learning-based and time series summarization-based methods, the LTV model performs poorly, which proves the importance of time series for robust flare prediction.

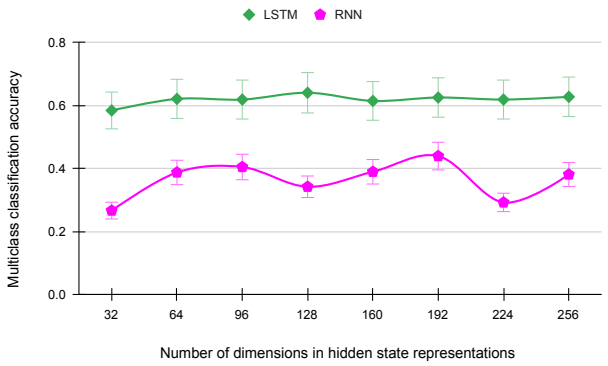



Fig. 4: Multiclass classification performance of LSTM and RNN-based models with varying number of hidden dimensions.

### E. Varying dimensionality of sequence models

The performance of deep learning-based sequence models such as LSTM and RNN depend greatly on the number of dimensions in the latent/hidden space. In both networks, the increase of dimensionality in hidden space reduces the underfitting tendency, while increases the computation cost and overfitting tendency. We computed the test accuracy of four-class classification on our MVTS dataset using stratified holdout sampling strategy, while changing the number of hidden dimensions in the range of $32, 64, ..., 256$ in the cases of both LSTM and RNN-based networks. We observed that the LSTM-based network shows more stable performance with the varying number of hidden dimensions, in comparison with the RNN-based network. In Fig. 4, we show the mean accuracy of both models in five runs with different network initialization and different sets of training data. Although this experiment shows the performance of sequence models by tuning one hyperparameter, i.e., the number of hidden dimensions, tuning other hyperparameters such as learning rate, linear layers, optimization algorithms, etc can result in better performances.

### F. Binary classification performance

In addition to classifying the solar active regions in different flare classes, a major use case in data-driven flare prediction is the binary classification, i.e., distinguishing major flaring events from minor flaring events or flare quiet events. In this experiment, we considered X and M class MVTS instances as flaring events, while we considered all other instances as non-flaring events. This transformation makes the dataset imbalanced, because non-flaring instances outnumber flaring instances. In Fig. 5, we show the binary classification performances of all models in terms of classification accuracy along with precision, recall, and F1 of flaring and non-flaring classes. In binary classification, the LSTM-based model outperforms all other baselines. We observe the similar performance of the

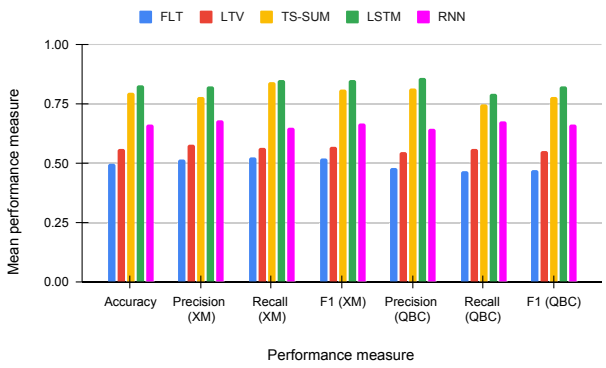



Fig. 5: Binary classification performance of all baselines

models as that of multiclass classification. Although the RNN-based model performed poorer than the TS-SUM method, the RNN-based model is an end-to-end classification model, which might outperform TS-SUM with more training data, more complex model, and more efficient hyperparameter search.

## V. Conclusion

In this work, we presented an end-to-end deep learning-based flare prediction model that leverages sequence models such as LSTM for learning representations of multivariate time series instances. In contrary to other state-of-the-art MVTS classification models, our proposed model is free of predefined (hand-engineered) statistical features for learning low dimensional representations of high dimensional multivariate time series data. Our experiments on a real-life solar flare prediction dataset demonstrate the superior performance of our model in performing multiclass and binary MVTS classification.

In the future, we look forward to designing more complex models such as: (1) introducing more than one linear layer before the final softmax layer, (2) adding additional layers in LSTM cells, and (3) learning the LSTM representation of each univariate time series whose concatenation will be fed into a series of linear layers. We also plan to apply our models in other MVTS-based solar event datasets [21], datasets where multivariate time series data are generated from other sources such as functional MRI (fMRI) time series data of different brain regions [22], and representation learning of dynamic graphs [23] where graph neural network-based temporal graph representations can be combined through sequence models.

## References


[1] J. Eastwood, E. Biffis, M. Hapgood, L. Green, M. Bisi, R. Bentley, R. Wicks, L.-A. McKinnell, M. Gibbs, and C. Burnett, "The economic impact of space weather: Where do we stand?" *Risk Analysis*, vol. 37, no. 2, pp. 206–218, 2017.

[2] R. A. Angryk, P. C. Martens, B. Aydin, D. Kempton, S. S. Mahajan, S. Basodi, A. Ahmadzadeh, X. Cai, S. F. Boubrahimi, S. M. Hamdi *et al.*, "Multivariate time series dataset for space weather data analytics," *Scientific data*, vol. 7, no. 1, pp. 1–13, 2020.

[3] S. M. Hamdi, D. Kempton, R. Ma, S. F. Boubrahimi, and 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.

[4] P. S. McIntosh, "The classification of sunspot groups," *Solar Physics*, vol. 125, no. 2, pp. 251–267, 1990.

[5] S. F. Boubrahimi, B. Aydin, D. Kempton, and R. A. Angryk, "Spatio-temporal interpolation methods for solar events metadata," in *2016 IEEE Intl. Conf. on Big Data, BigData 2016, Washington DC, USA, December 5-8, 2016*. IEEE Computer Society, 2016, pp. 3149–3157.

[6] J. P. Mason and J. Hoeksema, "Testing automated solar flare forecasting with 13 years of michelson doppler imager magnetograms," *The Astrophysical Journal*, vol. 723, no. 1, p. 634, 2010.

[7] Y. Cui, R. Li, L. Zhang, Y. He, and H. Wang, "Correlation between solar flare productivity and photospheric magnetic field properties," *Solar Physics*, vol. 237, no. 1, pp. 45–59, 2006.

[8] J. Jing, H. Song, V. Abramenko, C. Tan, and H. Wang, "The statistical relationship between the photospheric magnetic parameters and the flare productivity of active regions," *The Astrophysical Journal*, vol. 644, no. 2, p. 1273, 2006.

[9] K. Leka and G. Barnes, "Photospheric magnetic field properties of flaring versus flare-quiet active regions. ii. discriminant analysis," *The Astrophysical Journal*, vol. 595, no. 2, p. 1296, 2003.

[10] D. Yu, X. Huang, H. Wang, and Y. Cui, "Short-term solar flare prediction using a sequential supervised learning method," *Solar Physics*, vol. 255, no. 1, pp. 91–105, 2009.

[11] 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*, vol. 254, no. 1, pp. 101–125, 2009.

[12] 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*, pp. 1–19, 2013.

[13] A. Al-Ghraibah, L. Boucheron, and R. McAteer, "An automated classification approach to ranking photospheric proxies of magnetic energy build-up," *Astronomy & Astrophysics*, vol. 579, p. A64, 2015.

[14] M. G. Bobra and S. Couvidat, "Solar flare prediction using SDO/HMI vector magnetic field data with a machine-learning algorithm," *The Astrophysical Journal*, vol. 798, no. 2, p. 135, 2015.

[15] 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*, vol. 835, no. 2, p. 156, 2017.

[16] R. Ma, S. F. Boubrahimi, S. M. Hamdi, and 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.

[17] S. F. Boubrahimi, B. Aydin, P. C. Martens, and R. A. Angryk, "On the prediction of >100 mev solar energetic particle events using GOES satellite data," in *2017 IEEE Intl. Conf. on Big Data, BigData 2017, Boston, MA, USA, December 11-14, 2017*. IEEE Computer Society, 2017, pp. 2533–2542.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH 2010, 11th Annual Conf. of the Intl. Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, T. Kobayashi, K. Hirose, and S. Nakamura, Eds. ISCA, 2010, pp. 1045–1048.

[20] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014.

[21] S. F. Boubrahimi, S. M. Hamdi, R. Ma, and R. A. Angryk, "On the mining of the minimal set of time series data shapelets," in *IEEE Intl. Conf. on Big Data, Big Data 2020, Atlanta, GA, USA, December 10-13, 2020*. IEEE, 2020, pp. 493–502.

[22] S. M. Hamdi, B. Aydin, S. F. Boubrahimi, R. A. Angryk, L. C. Krishnamurthy, and R. D. Morris, "Biomarker detection from fmri-based complete functional connectivity networks," in *IEEE Intl. Conf. on Artificial Intelligence and Knowledge Engineering, AIKE 2018, Laguna Hills, CA, USA, September 26-28, 2018*. IEEE, 2018, pp. 17–24.

[23] S. M. Hamdi and R. A. Angryk, "Interpretable feature learning of graphs using tensor decomposition," in *2019 IEEE Intl Conf. on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019*. IEEE, 2019, pp. 270–279.