Interpretable Deep Learning for Prediction of Solar Flares

Gautam Datla and Jason T. L. Wang

New Jersey Institute of Technology
University Heights, Newark, NJ 07102, USA

Abstract

Machine learning and Deep learning models are often considered black box models as their internal workings tend to be opaque to the users. Because of this lack of transparency, it might be challenging to understand the reasoning behind the model's predictions, which can raise doubts about the model's reliability, accountability, and fairness. So, In this paper, we try to present a novel interpretable deep learning model which uses a long short-term memory with attention to predict if an Active region (AR) would be producing a γ class flare within the next 24 hours. In particular, we considered M-class flares and used an LSTM-attention-based model to predict their occurrences. The data used for the $\geq$ M-class LSTM model was based on Geostationary Operational Environmental Satellite X-ray flare catalogs provided by the NCEI. The crux of this approach is to first represent the data collected from multiple active regions as sequential data and then use LSTM with attention mechanisms to capture the temporal dynamics of the data. Now, to make the predictions made by the model more accountable and reliable, we have leveraged post hoc model-agnostic global interpretability techniques. Interpretability techniques such as Hidden State Visualization and Input saliency using guided-back propagation were employed to elucidate the factors contributing to a given prediction for a given input data sequence and to provide insights into model behavior across multiple test data sequences within an active region.

Keywords: Interpretable Deep Learning, LSTM, Hidden state visualization, Input saliency, Guided-backpropagation, Solar flare predictions

1 Introduction

Solar flares are swift and intense bursts of energy on the Sun's surface, which tend to occur when the magnetic energy trapped in the Sun's atmosphere is released. While most of these flares are pretty small and harmless, few of these can turn out to be significant and violent as they tend to release massive amounts of energy into space. These intense flares often produce coronal mass ejections (CME), which can cause dangerous geomagnetic storms that could potentially negatively impact Earth's life and technological advancements (Norsham and Hamidi, 2019). This is because the high-energy particles emitted by these flares could possibly disrupt communication systems, damage satellites, or cause power outages on earth. Moreover, they can also pose a radiation hazard to astronauts and airline passengers traveling at high altitudes. Thus, in order to mitigate the potential damage, it is imperative that we accurately predict the occurrence of these flares.

Though the triggering mechanism which is causing these flares is not entirely understood yet, quite a lot of studies have shown us that the occurrence of these flares and coronal mass ejections tend to be fueled by the build-up of magnetic energy within the Sun's outer atmosphere (Corona), which are then abruptly released when the magnetic field breaks apart through a phenomenon called magnetic reconnection. In the occurrence of such a phenomenon, the energy gets converted to various forms such as heat, light, and kinetic energy, resulting in the solar flares that we tend to observe as bright chromospheric ribbons. Since the build-up of this coronal free energy is caused by the long-term evolution of magnetic fields on the Sun's photosphere, we can observe and analyze these fields through photospheric vector magnetograms to understand and predict this phenomenon. While analyzing magnetograms, we could consider various aspects such as magnetic shear, magnetic energy dissipation, magnetic flux, etc. Moreover, many researchers in this field have already shown that we can leverage predictive modeling with the features extracted from these magnetograms to predict the occurrence of solar flares accurately. In order to accurately predict the occurrence of these flares, it is imperative that we identify the right set of predictive parameters. For example, (Nishizuka et al., 2017)

examined chromospheric, X-ray intensity, and photospheric vector magnetic field data to forecast significant flares, while Bobra and Couvidat used 25 predictive parameters provided by the Helioseismic and Magnetic Imager (HMI). Ano2ther essential aspect we must consider is the modeling technique we will be using on the extracted predictive parameters. For example, (Barnes et al., 2016; Breiman, 2001; Liu et al., 2017) used Random Forests, (Liu et al., 2017) used Decision trees, (Huang et al., 2013; Liu et al., 2017; Winter and Balasubramaniam, 2015) used KNNs, SVMs were used by (Bobra and Couvidat, 2015; Huang et al., 2013; Muranushi et al., 2015; Yuan et al., 2010), (Nishizuka et al., 2017) used Extremely randomized trees and finally neural networks were used by (Ahmed et al., 2013; Colak and Qahwaji, 2009; Higgins et al., 2011; Nishizuka et al., 2018). Thus the choice of the predictive model and predictive parameters both play a pivotal role in deciding how accurately we are forecasting flares.

In this research paper, we try to use a deep learning method called LSTM (Long short-term memory) with attention mechanisms for forecasting $\geq$ M-class solar flares in Active regions (Within 24 hours from a given point in time). The use of attention mechanisms in LSTMs allows our model to focus on the significant components of the input sequence. To train the LSTM model, we have used the data from the Solar Dynamics Observatory/Helioseismic and Magnetic Imager (SDO/HMI) vector magnetic field data along with flaring history. Now even though a plethora of research has shown the use of RNNs/ LSTMs for flare forecasting (Ahmed et al., 2019; Chatterjee and Mandal, 2019; Liu et al., 2019; Sharma et al., 2020; Xu et al., 2018; Zhang and Su, 2018), there has been a minimal amount of research with regards to the interpretability aspect of these models. Now, since the end user only has access to the final predicted output and there is a minimal amount of disclosure with regards to the internal workings, this could lead to potential trust issues on the predictions made by these models. So, in this paper, we try to create an interpretable LSTM model, using which we can not only determine the relevance and sensitivity of features to the outputs but also try to understand how "the black model" operates internally. While hidden state visualizations were used to uncover how changes in the LSTM layer's hidden states result in changes in the forecasted flares, we used guided backpropagation to show how the model focuses on a different set of features before and after the attention mechanisms. By doing so, we can have a better understanding of how our model is processing the input sequences internally.

2 Exploring the Feature Space

The research predominantly relied on two data sources, Space-weather HMI Active Region Patches (SHARP) data product produced by the SDO/HMI team and cgem.Lorentz data series. Here, the SHARP data includes automatically identified and tracked active regions in map patches and provides a range of physical parameters useful for predicting solar flares (Bobra et al., 2014), while cgem.Lorentz data series provides us with estimates of integrated Lorentz forces, which enable the examination of the dynamic processes within each active region (Fisher et al., 2012). By analyzing these integrated Lorentz forces, we can better understand the accumulation and dissipation of magnetic energy.

Now, to train the LSTM model, a Flare database having B, C, M, and X flare classes (Which identifies ARs between May 2010 and May 2018) was collected using the GOES (Geostationary Operational Environmental Satellite) X-ray flare catalogs provided by the NCEI (National Centers for Environmental Information) and then the HMI.sharp and cgem.Lorentz data series were used to study these flares (Liu et al., 2019). Since the data was collected at a 1-hour cadence, the model was trained using the physical parameters of active regions and the estimated Lorentz forces for each hour between May 2010 and May 2018. Now talking of the feature space in particular, two groups of predictive parameters were primarily used in this study. The first group of parameters (25 parameters) described the Active region's magnetic field properties similar to (Bobra and Couvidat, 2015), while the second group (15 parameters) was primarily related to the active region's flaring history. Now, six of these flaring history parameters are related to the time decay values, with the rest showing the flare history of a data sample x_t in AR as described in (Liu et al., 2019; Nishizuka et al., 2017). These 9 features included Bhis, Chis, Mhis, Xhis, Bhis1d, Chis1d, Mhis1d, and Xhis1d, representing the total number of B, C, M and X-class flares that occurred before the observation (Here 1d represents the features showing their occurrence only within the past 24 hours).

2.1 Feature Selection

Now, In the context of forecasting solar flares using the above-described set of 40 features, it is possible that some features may demonstrate greater relevance for this task than others. Thus, feature selection becomes extremely important so that we can remove irrelevant and redundant features for the flare forecasting model. To do so, a cross-validation strategy was adopted wherein the predictions of the model were evaluated with the performance metric (True skill statistics) being computed every two folds. Now, the importance of each of the 40 features was studied by making predictions using only one feature at a time, with the probability threshold set to maximize the test score in each test. Then the corresponding mean scores were recorded for each feature and sorted in descending order to identify the most relevant features for the LSTM model similar to (Liu et al., 2019). On ranking the importance of all these predictive parameters, it was found that using all 40 features did not result in a good performance rather, using only the top 22 important features gave us the highest mean cumulative TSS score, thus resulting in better prediction performance. And among these ten were SDO/MHI magnetic field parameters, which have been cited as crucial forecasting parameters in prior studies like (Bobra and Couvidat, 2015; Liu et al., 2017, 2019). Table-1, given below, shows the final set of features that were used for the $\geq$ M-class LSTM model.

Feature Name	**Description**
TOTUSJH	Total Unsigned Current helicity
Cdec	Time decay value based on Previous C-Class flares
TOTUSJZ	Total Unsigned vertical current
Chis1d	One day history o C-class flares in Active region
USFLUX	Total unsigned flux
TOTBSQ	Total magnitude of lorentz force
R VALUE	Sum of flux near polarity inversion line
TOTPOT	Total photospheric magnetic free energy density
Chis	Total history of C-class flares in Active region
SAVNCPP	Sum of the modulus of the net current per polarity
AREA ACR	Area of strong field pixels in the active region
Edec	Time decay values based on the magnitudes of all the previous flares
Xmax1d	Maximum X-ray intensity one day before
ABSNJZH	Absolute value of net current helicity
Mhis	Total history of M-lass flares in Active region
Mdec	Total decay value based on the previous M-class flares only
MEANPOT	Mean photospheric magnetic free energy
Mhis1d	1-day history of M-class flares in Active region
TOTFX	Sum of the X-components of Lorentz force
TOTFZ	Sum of the Z-components of Lorentz force
MEANSHR	Mean shear angle
SHRGT45	Area fraction with shear greater than 45 degrees

Table-1 Features used in the M-class LSTM model

2.2 Feature Normalization

Since the features we deal with tend to vary on scales and units, it is necessary that we normalize the features. For physical magnetic features, standardization was used, and for the features dealing with flare history min-max scaling was used. This was done by observing the feature distributions of both the set of features as previously done in (Liu et al., 2019)

$$physical_i^k = \frac{v_i^k - \mu_i}{\sigma_i} \qquad f_history_i^k = \frac{v_i^k - min_i}{max_i - min_i}$$

3 Methodology

3.1 Prediction Task

Similar to what (Bobra and Couvidat, 2015; Jonas et al., 2018; Nishizuka et al., 2017) have done in their prior work, we try to use an active region's past observations to predict the occurrence of future flares. In particular, our primary focus is on the prediction of the occurrence of M-class flares in a particular AR within the next 24 hours. In order to construct the negative and positive data samples for the dataset, positive labels were assigned to the data samples collected 24 hours prior to a $\geq$ M-class or $\geq$ X-class flare in an active region, while negative labels were assigned to all other data samples in the same active region. In case there were some missing time points in the data, or if there weren't enough data samples in the 24-hour period prior to the flare occurrence, synthetic data samples were added to the dataset with the feature values being set to zero. Doing this ensured that we got a complete time-series dataset without any gaps. This zero-padding approach was applied to the dataset after normalizing the data so that the normalization process remains unaffected by zero-padding. In accordance with the methodology stated by (Bobra and Couvidat, 2015), any incomplete features or any observations for which ARs are outside the $\pm 70^\circ$ of the center meridian were not considered for the prediction task. The train, validation, and test data were used from the data collected in the years 2010-2013, 2014, and 2015-2018 respectively. Now by doing so, we can ensure that our LSTM model will make predictions on Active regions that weren't a part of the data used for training the model. This is quite crucial as it ensures that our model can accurately predict the flaring activity for unseen active regions.The analysis of Table 2 shown below indicates that the dataset under consideration is imbalanced, with a higher frequency of negative data samples compared to positive ones. This imbalance caused in the dataset stems from the fact that most of the active regions do not tend to produce flares.

Dataset Type	M-Class	
	Positive	Negative
Train	2,710	81,867
Validation	1,347	25,126
Test	1,278	43,411

Table 2: Positive and negative data samples used for the $\geq$ M-class model

3.2 The Model

We have used a deep learning model for forecasting the occurrence of $\geq$ M-class flares in an AR. Here the deep learning model that we used was designed to predict whether an active region will produce a $\geq$ M-class flare within the next 24 hours. For building this model, we used a type of neural network called LSTM along with an attention mechanism. LSTM networks were primarily designed to address the problem of vanishing gradients in Recurrent neural networks. LSTMs achieve this by using a memory cell, input gate, output gate, and forget gate as shown in Figure 1. Here the input gate determines what information should be stored in

the cell state, Forget gate tells which information should be thrown away from the cell state, and the output gate is used to provide the activation to the final output of the LSTM unit. The equations i_t, f_t, and o_t shown below represent the outputs of the input gate, forget gate, and output gate, respectively. All these gates use the Sigmoid activation function (σ), thus outputting a value between 0 and 1. Here, 0 means that the gates are blocking everything, while 1 represents that they are allowing everything to pass through them.

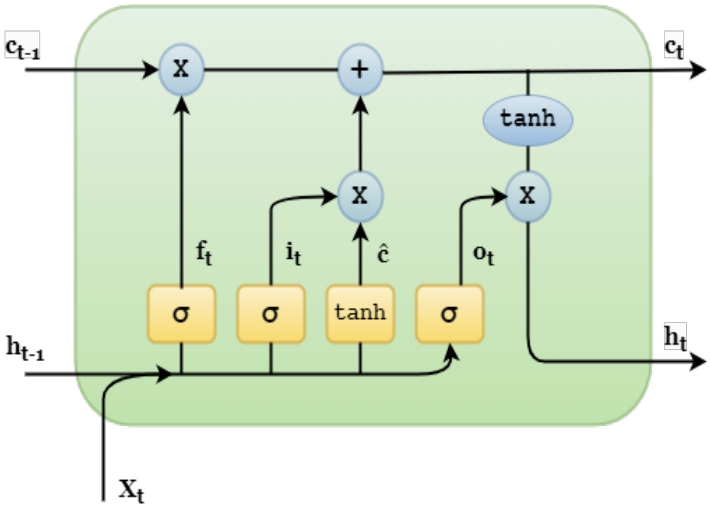


Fig. 1: **LSTM unit illustration. Here X_t,f_t, i_t, $\widetilde{c}$, and o_t represent the input vector , forget gate, input gate, candidate state, and output gates respectively. While h_{t-1}, and c_{t-1} represent the hidden state, and cell states from the previous time step, h_t, and c_t represent the final hidden state and cell state at the current time step that is output by the LSTM unit**

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$
$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$
$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

The candidate state is then determined using the input and forgets gates. These gates basically control what the cell state should forget from the previous timestamp and what the cell state should consider from the current time stamp. The resulting cell state that should be output by the LSTM block is then determined by filtering this and then passing it through an activation function. The candidate state and output vectors of LSTM unit are computed using the equations shown below,

$$\widetilde{C} = tanh(w_c[h_{t-1}, x_t] + b_c)$$
$$C_t = f_t * c_{t-1} + i_t * \widetilde{C}$$
$$h_t = o_t * tanh(C_t)$$

Here, C_t represents the cell state at a given timestamp t, while $\widetilde{C}$ at a timestamp t represents the candidate for cell state. And finally, the output vector represented by h_t is calculated using the new cell state C_t. The neural network architecture that we will employ includes an LSTM layer with ten units. Figure 2 depicts the unrolled version of this LSTM layer over time. In this figure, Each LSTM unit is represented by LSTM_x, where x ∈ (1,m) and the hidden states, $\text{h}_{x,t-s}$ represent the hidden state passed on from LSTM_x unit to the LSTM_{x+1} unit at t-sth time step. The curved arrow at the bottom of each LSTM unit represents the input

x_{t-s} at the corresponding time step. So every LSTM unit in the LSTM layer takes as input the hidden state from the previous unit as well as the input at the corresponding time step.

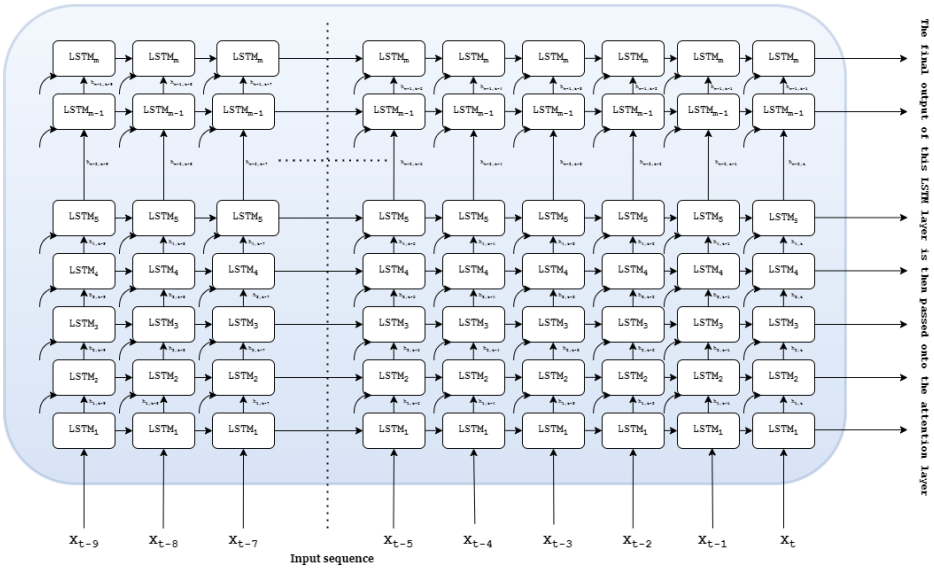


Fig. 2: **Illustration of the LSTM layer unrolled through time. At each time step s, the corresponding input sequence x_{t-s} is passed on to each LSTM unit along with the hidden state from the previous LSTM unit.**

Following an LSTM layer with m units, attention mechanisms were added so that the model could automatically search for parts of the input sequences most relevant to the target predictions. This layer would take into consideration the hidden state at every time step and then calculate a weight for each state. By doing so we try to signify the relevance of information in each state. To achieve this, we employ an attention layer that incorporates a content-based function (score), which has been previously utilized by (Luong et al., 2015).In the equations shown below, h_i and h_t represent the source state and the target state, respectively, while W represents all the learnable parameters.

$$score(h_i, h_t) = h_t^T W h_i$$

$$w_i = \frac{e^{score(h_i, h_t)}}{\sum_j e^{score(h_j, h_t)}}$$

Once all the weights, w_i are calculated, we can then use these to compute the context vector as shown below.

$$c_t = \sum_i w_i h_i$$

Then the final attention vector A_v (Output of our attention mechanism) is derived by obtaining the hyperbolic tangent activation of the concatenation of the last hidden state h_t and context vector c_t

$$A_v = tanh(W_V[c_t; h_t])$$

Once the attention vector is computed it is sent to fully connected layers with the first layer having 200 neurons and the second layer having 500 neurons. Then the final predictions are done by the output layer using the softmax activation function. The entire neural network architecture can be summarized as shown in Figure 3. Now say we are passing in a data sample x_t at a time "t" to our neural network, then during training, we select a sequence of 10 data points i.e $x_{t-9}, x_{t-8}, x_{t-7}$....., x_{t-1}, x_t from the train data and

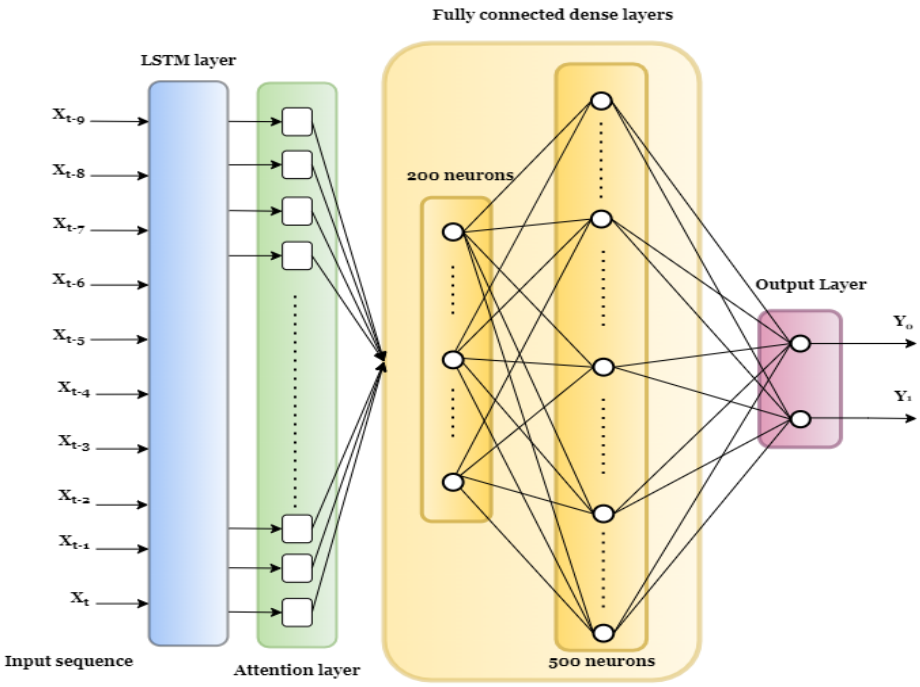


Fig. 3: **Illustration of the overall neural network architecture used for forecasting $\geq$ M-class flares. Here the LSTM network takes an input sequence $X_{t-9}, X_{t-8}, X_{t-7}, ..., X_{t-1}X_t$ as input and produces the final output as a two-dimensional vector [y0,y1], which is determined by the output layer's probability based on softmax activation.**

uses these "10" consecutive data points for training our neural network. Now since the data was collected at a 1-hour cadence, the input sequence that we will be using spans over a period of 10 hours, and the label of x_t would be the label for the corresponding input sequence. Now, say for the $\geq$ M-class model, we have x_t as a positive class, then the corresponding input sequence is defined as positive else, the input sequence would be defined negatively. Another aspect to consider would be the active regions, as we would process these ARs separately. Say, we have two active regions, AR1 and AR2, with x, and y data samples for each AR, respectively. Now using the zero-padding strategy, we generate x and y sequences for AR1 and AR2 and feed these sequences one at a time to the LSTM model. Though AR1 and AR2 might have some overlapping points, we process them separately one at a time. For optimizing the parameters in the neural network, we use a weighted cross-entropy cost function since we are dealing with an imbalanced dataset. The cost function can be computed using the formula shown below as,

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} w_k y_{nk} log(\hat{y_{nk}})$$

Here, N in this equation represents the total number of sequences in our dataset, with each sequence having 10 consecutive data samples. K here represents the number of classes i.e 2 (positive and negative) and y_{nk}, $\hat{y_n k}$ denote the observed and predicted probability for the n^{th} sequence to belong to the k_{th} class respectively. During the testing phase to predict if an AR will be producing an $\geq$ M-class flare at a timepoint "t," we will be using the last "m" consecutive data samples along with x_t, which is similar to what was done in (Liu et al., 2019). That is we will be passing in all the samples x_{t-m+1}, x_{t-m+2}, x_{t-m+3}, x_{t-m+4}, ..., x_t to our pre-trained model and obtain the results as a two-dimensional vector [1,0] or [0,1] depending on the class that x_t belongs to. However, It is worth noting that this approach differs significantly from other research studies, such as those conducted by (Bobra and Couvidat, 2015; Nishizuka et al., 2018), which only utilize the data sample x_t to generate predictions.

4 Performance Evaluation

The LSTM architecture shown in Figure 3 was implemented in Python using Keras and TensorFlow. To have an accelerated convergence during backpropagation, a mini-batch strategy described in (Goodfellow et al., 2016) was used. Adam optimizer (learning rate = 10^{-3}, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ with a batch size of 256 was used to train the neural network and the model's hyperparameters were tuned using the validation dataset. Since we are dealing with an imbalanced dataset with a number of negative data samples outnumbering the number of positive samples, traditional performance evaluation metrics such as accuracy, precision, and recall may prove to be inefficient. This stems from the fact that the classifier can simply predict the majority class (Negative class) for all the samples and achieve higher accuracy. So, we have used the following evaluation metrics to evaluate the classification accuracy of the LSTM models,

i) **Balanced Accuracy score (BACC):** BACC is calculated as the average of sensitivity and specificity and is a classification metric that considers the class distribution imbalance, especially in binary classification tasks (García and Herrera, 2009). The balanced accuracy score is bounded between 0 and 1, where a value of 1 indicates that the classifier has achieved a perfect classification performance, while a value of 0 suggests that the classifier's performance is no better than random guessing.

$$\text{BACC} = \frac{TPR+TNR}{2}$$

ii) **Matthews Correlation coefficient(MCC):** MCC is an excellent performance evaluation metric for binary classifications, especially when class distributions are highly skewed. This metric returns a value between -1 and 1, with 1 representing a perfect prediction and -1 representing a completely wrong prediction (Chicco et al., 2021).

$$\text{MCC} = \frac{TPxTN-FPxFN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

iii) **AUC-PR (Area under the Precision-Recall curve)** : AUC-PR metric measures the trade-off between precision and recall at different threshold values and comes in handy in situations wherein the positive class is rare as it primarily focuses on the classifier's performance on the positive class.

iv) **F1-Score**: F1-score computes the harmonic mean of precision and recall and tries to provide a balance between these two metrics. It is quite a useful metric as it is not sensitive to class imbalance.

$$\text{F1-Score} = 2 * \frac{Precision*Recall}{Precision+Recall}$$

v) **TSS (True-skill statistics) :** TSS is a statistical indicator measure of agreement between the observed and predicted values in a binary classification problem. This metric is particularly useful for imbalanced datasets as it considers the number of positive class instances.

$$\text{TSS} = \text{TPR} + \text{TNR} - 1 = \text{TPR} - \text{FPR}$$

In addition to these metrics, bootstrap sampling was also utilized to assess the validation accuracy of the LSTM model. Using the bootstrap method, 100 resampled datasets with replacements were generated from the original dataset to measure the variability in the LSTM model's accuracy caused by variations in training data. Tables 3 and 4 provide a concise summary of the experimental results. Upon careful analysis of these results, it is evident that our model achieved the highest accuracy in forecasting flares at a dropout rate of 5%.

5 Interpretability

To forecast the occurrence of flares in an active region using the LSTM model, the input sequences that we provide go through several layers, including LSTM, attention, and fully-connected dense layers, with each layer, performing numerous mathematical operations and applying non-linear transformations to the

	F1-score		AUC-PR	
Dropout rate	Mean	S.D	Mean	S.D
0.5	0.9202	0.0315	0.8250	0.0503
0.1	0.9662	0.0206	0.9382	0.0605
0.05	**0.9796**	0.0199	**0.9623**	0.0497

Table 3. Validation accuracies at different dropout rates

Dropout rate	BACC	F1-Score	AUC-PR	MCC	TSS
0.5	0.835	0.951	0.995	0.380	0.670
0.1	0.868	0.939	0.996	0.368	0.669
0.05	**0.874**	**0.962**	**0.996**	**0.402**	**0.770**

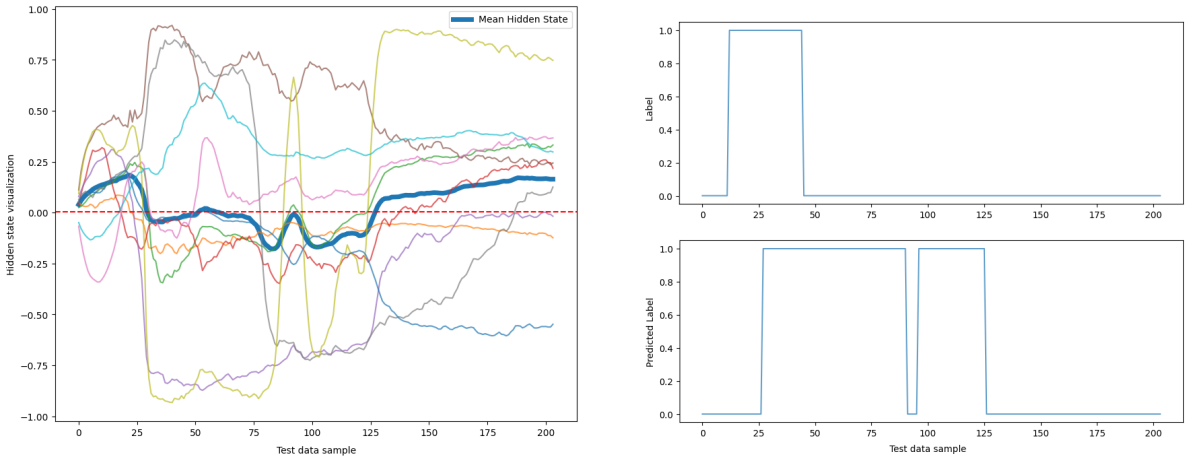
Table 4. Testing accuracies at different dropout rates

input data. Now because of the intricacy involved and the lack of clear and comprehensible explanations, it becomes quite difficult to interpret the predictions being made by our model. So, in order to make its predictions accountable and reliable, it becomes quite crucial that we understand their internal working mechanisms, either through introspection or through a produced explanation. To enhance the interpretability of our $\geq$ M-class LSTM model, we have employed post hoc global model-agnostic techniques as it would offer us reliability in terms of the predictions made across multiple input sequences from an active region. To accomplish this, we have used a test dataset that was sourced from the NOAA (National Oceanic and Atmospheric Administration) active region 12381 with 204 data samples between 2015-07-04T21:22:24.70Z to 2015-07-13T18:22:24.70Z (Approximately 1-hour cadence data for about 213 hours). In the chosen data, around 16.17% samples exhibited a positive outcome of $\geq$ M-class flare occurrence within the following 24 hours.

5.1 Visualization of Hidden States in LSTM Units

The methodology of visualizing the hidden states in RNNs/ LSTMs has been a topic of active research and has been used in prior studies such as (Garcia et al., 2021; Karpathy et al., 2015). The reason for us to choose this methodology is because of the fact that at any given time step, the hidden state of an RNN/LSTM summarizes the network's memory of the inputs it has seen up to that point of time and thus, in a way depicts the internal state of the network at that particular time step. Now the way in which the hidden state will be interpreted is specific to the RNN/LSTM architecture being used and the task for which it is being used. For example, if we are dealing with a language modeling task like (Garcia et al., 2021), the hidden state might be used to represent the network's understanding of the previous words in a sentence. However, in our case, for predicting the occurrence of solar flares, the interpretation of the hidden state would depend on the SHARP and cgem.Lorentz data series being used. For instance, the hidden state could possibly capture a pattern/trend in the input data, such as the sudden accumulation of magnetic energy in an AR which could an indicative factor of the likelihood of the occurrence of a solar flare in near future. Now, using these hidden states, our LSTM model would try to capture temporal dependencies in input data and predict the likelihood of the occurrence of a solar flare. Now, as we know, the hidden state is a vector of values that is updated every time the network processes and input sequence. Hidden state visualization is the process of graphically visualizing these changes in the hidden state of an RNN/LSTM as it interprets a series of inputs. Now talking in particular about the plots that were used in this study, in addition to examining the changes in the mean-hidden state of the LSTM layer at the last time step, we also try to visualize the changes in the hidden state of each LSTM unit. By doing so, we can not only interpret how the overall LSTM layer is learning and encoding the input sequence as a whole but can also gain insights into the specific aspects of the input sequence that each LSTM unit is learning and how these units are interacting with each other

to learn the overall behavior of the layer. The obtained results are concisely summarized in Figures 4 and 5 below.



(a) Hidden state trajectories

(b) Distribution of the original and predicted class labels

Fig. 4: **Hidden state visualization of the LSTM layer along with the distribution of original and predicted class labels for NOAA AR 12381 shown for reference**

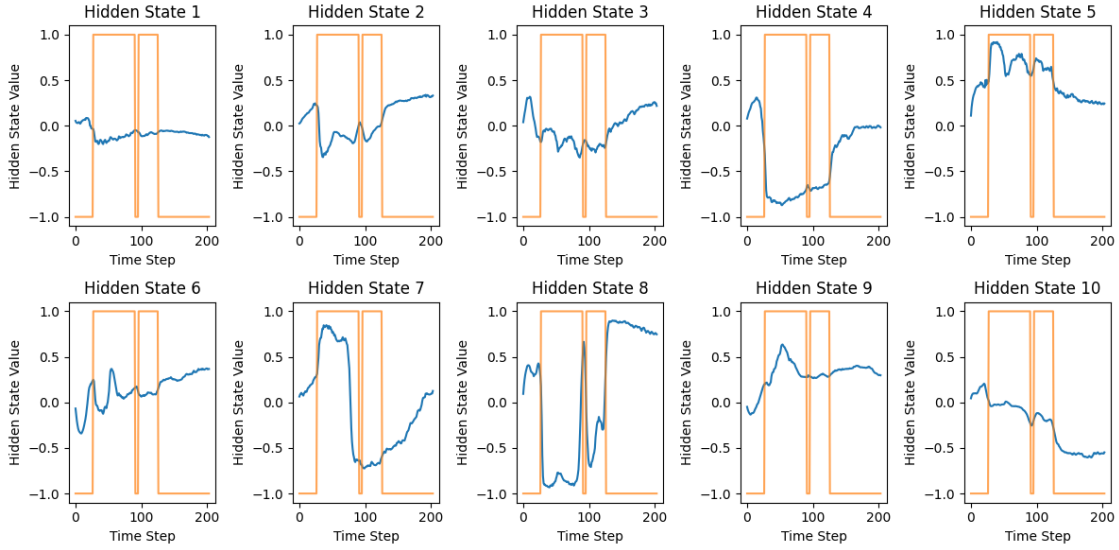


Fig. 5: **Hidden state trajectories of individual LSTM units along with the distribution of the predicted class labels shown for reference**

Figure 4(a), shown above, illustrates the mean hidden state trajectory of the LSTM layer at the last time step for each sample in the input sequence, along with the hidden state trajectories of individual LSTM units. The dotted line indicates the selected threshold for analysis in the plot. This threshold was determined based on the average of all mean hidden state values taken immediately before and after a change occurred in the flare forecast. Upon analyzing the plot, it was observed that $\geq$ M–class flare forecasts were negative when the mean hidden state was above the threshold. Conversely, the flare forecasts were positive when the mean hidden state was below the defined threshold, as Figure 4(b) depicts. However, on analyzing the 8th LSTM unit's hidden state trajectory depicted in Fig 5, we find out a similar pattern but rather with more pronounced changes occurring whenever there was a sudden change in the forecasted flare. Upon further examination, it was observed that these outcomes were generalizing effectively to other active regions, with

a change only in the LSTM unit that represented these changes. Thus we can conclude that while the mean hidden state by itself captured the underlying reasoning behind the model's predictions, the individual unit's hidden state proved to be more robust, especially in scenarios with abrupt shifts in the predicted class. This observed correlation between the LSTM layer's mean hidden state and changes in the predicted class suggests that the LSTM layer learns the data by encompassing the entire feature space rather than overfitting a particular feature. Furthermore, Establishing such a correlation between the mean hidden states and flare forecasts could facilitate a focused analysis of local state changes, pattern matching within large data sets, and alignment with solar domain structural annotations.

5.2 Analysis of Feature Sensitivity

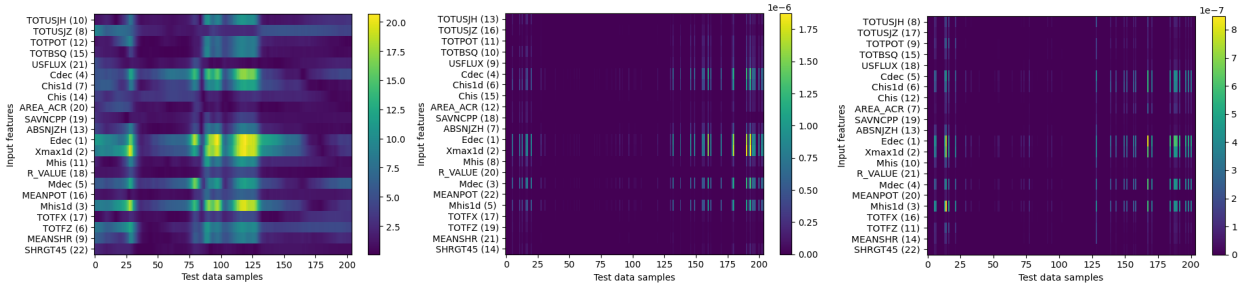
Having visualized the changes in hidden states, we must also identify which predictive parameters from the input feature space are causing these changes. To do so, we computed the absolute sum of gradients across the time dimension of the LSTM output layer with respect to the input sequence being passed into the network and then visualized these gradients using a heatmap (Li et al., 2015). By analyzing this heatmap, we can understand how changes in the values of an input feature affect the output of the LSTM layer with respect to this input feature. If an input feature has consistently high gradient values across multiple test data samples, the sensitivity of the LSTM layer to changes in the values of this input feature is high. In other words, the LSTM layer is very sensitive to the changes in the values of this input feature.

Figure 6(a) illustrates the heatmap resulting from the evaluation of the LSTM layer on the test data samples obtained from the active region NOAA AR 12381. While the X-axis of the plot enumerates 204 test data samples in AR 12381, the Y-axis lists the input features along with a numerical value which signifies the degree of how changes in the values of the corresponding input feature impact the LSTM layer's output with respect to the input feature, with a lower number indicating a higher impact. The color bar positioned on the right corresponds to the absolute sum of gradients across the time dimension of the LSTM layer with respect to the input sequence from NOAA 12381. This color bar provides an indication of the LSTM layer's sensitivity to the input features, where a higher value with a brighter color signifies a greater level of sensitivity at the corresponding test data sample. It can be seen from Figure 6(a) that changes in the values of different features have varying effects on the LSTM layer's output. Moreover, we can observe from this heatmap that the LSTM layer is most sensitive to the features related to flaring history, such as Edec Xmax1d, Mhis1d, Cdec, and Mdec, indicating that changes in the values of these flaring history features exhibited the highest impact on the output of the LSTM layer with respect to these features.

Upon expanding a similar analysis to include the post-attention and post-fully connected dense layers and examining the corresponding heatmaps presented in Figure 6(b) and Figure 6(c), we observe that the attention layer and the entire flare forecasting model demonstrate substantial sensitivity to flaring history parameters, akin to the pre-attention layers. Though there was a relatively minor change in the sensitivity of individual features, further analysis of these heatmaps indicates that the time decay values computed based on the magnitudes of all the previous flares (Edec) and the maximum X-ray intensity one day before (Xmax1d) remained the most sensitive features throughout our flare forecasting model. Furthermore, another notable observation from the heatmaps is that the attention mechanism greatly diminishes the magnitude of sensitivity associated with these features.

5.3 Analysis of Feature Relevance

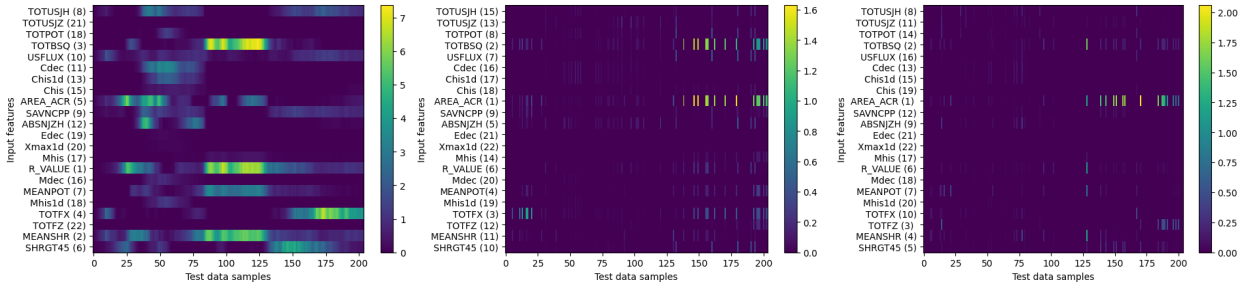
Given that the deep learning model for $\geq$ M-class flare forecasting under consideration is not solely constrained to a basic Long Short-Term Memory (LSTM) layer, as depicted in Figure 3, it is imperative to examine other aspects of the deep learning model, particularly the impact of the attention mechanism and dense layers. The attention layer in our deep learning model operates on the output hidden states of the LSTM layer, thus typically causing it to focus on a different set of features than that of the LSTM layer. Now, since the features relevant to the output of these layers change, it is imperative that we analyze how the attention mechanism causes a change in the relevance of features. So, in addition to measuring the sensitivity of the model's output to individual input features, we will employ feature relevance methods to quantify the overall importance of each input feature. To achieve this, we use a gradient-based saliency method called



(a) Pre-Attention (b) Post-Attention (c) Entire Model

Fig. 6: **Heatmaps illustrating the sensitivity of the input feature space.**

guided backpropagation on these layers. This algorithm works by calculating gradients of the output with respect to the input sequence, but only positive gradients are backpropagated through the activation functions in the network (Springenberg et al., 2015). This is done by setting the negative gradients to zero while passing on the positive gradients unchanged. By doing so, we can emphasize the positive contributions of the input features (as we tend to suppress the negative contributions by removing the negative gradients). Then we try to visualize these gradients as a heatmap showing which feature has the most contribution towards the output. We do this for both the outputs of the LSTM layer and the attention layer, then try to analyze which features are the most relevant before the attention mechanism (i.e., after the LSTM layer) and after the attention mechanism. The heatmaps illustrated in figure 7(a) and figure 7(b) depict the degree of feature relevance in the input feature space before and after the application of the attention mechanism (Luong et al., 2015), while the heat map in figure 7(c) elucidates the feature relevance of the $\geq$ M-class model utilized for NOAA AR 12381, thereby facilitating comprehension of the most pivotal features that play a prominent role in the model's decision-making process. In this figure, the numerical values associated with each feature name represent the impact of the respective feature on the output, where a lower number suggests a higher overall importance. The color bar on the right provides an indication of the network's relevance to the input features, where a higher value corresponds to a greater level of importance in the corresponding data sample.



(a) Pre-Attention (b) Post-Attention (c) Entire Model

Fig. 7: **Heatmaps illustrating the feature relevance of the input feature space.**

Although the significance of the parameters pertaining to the magnetic field properties of the active region was observed from fig7., further analysis indicates that our model with $\geq$ M-class accuracy prioritizes a distinct set of features prior to and subsequent to the implementation of the attention mechanism. For instance, post attention, the importance of Total Unsigned Current helicity (TOTUSJZ) increases, supporting the findings of (Liu et al., 2019), which considered TOTUSJZ to be an essential feature in all of their LSTM models. This observed shift in feature relevance can be attributed to the attention layer's goal of improving the model's performance by identifying and focusing on the most salient features of the input sequence for

accurately forecasting solar flares. However, It is essential to keep in mind that the relevance of features identified by the attention mechanism may not be the same for the entire model as it is for a single layer, hence evaluating the effectiveness of the entire model for flare forecasting is crucial rather than solely relying on the attention mechanism. Thus, Apart from interpreting the Long Short-Term Memory (LSTM) layer and attention mechanisms, it is imperative to adopt a holistic approach to identify the salient features within the input feature space that contribute to the model's ability to predict M-class flares. This entails a thorough exploration of the input feature space to isolate the relevant features that drive the model's performance. Such an approach is essential in gaining a comprehensive understanding of the model's underlying mechanisms and its capacity to capture and exploit the relevant information within the input data. In order to achieve this, it is crucial that we consider feature relevance for the entire model, which identifies the input features that have the greatest impact on the model's overall predictive performance (Ali et al., 2018). Upon analyzing the heatmap shown in Fig8., we discover that the magnetic field properties of ARs are the most crucial input features for forecasting flares. On comparing figure 7(b) and figure 7(c), In the context of feature importance analysis, it has been observed that the relevance of important features undergoes a change before and after passing through dense layers. Though there was a relative change in the feature relevance, the most critical factor that influences the forecasted $\geq$ M-class flares remained consistent, which is the area of strong field pixels in the active region. Furthermore, our findings align with the results of previous studies, such as (Aschwanden and Freeland, 2012; Li et al., 2018), which reported a significant correlation between magnetic field properties and the prediction of major flares. Additionally, after the fully connected dense layers, consistent with (Wang et al., 2017), our study highlights an increased importance in the role of the SHRTG45 and MEANSHR features in forecasting M-class flares, implying that active regions with twisted and sheared magnetic fields are more prone to major flares such as $\geq$ M-class flares.

6 Discussion and Conclusions

In this paper, we present a novel interpretable deep learning architecture that leverages flaring history parameters and the magnetic field properties of an Active Region (AR) to predict the occurrence of $\geq$ M-class flares. Though our deep learning model forecasts $\geq$ M-class flares at high accuracy, the lack of transparency in the internal workings of the model necessitates the use of post hoc global model agnostic approaches to provide reliability and accountability for the predictions it generates. We aim to enhance the interpretability of our model by tracing the trajectory of the input sequences from an active region within the model, which allows us to gain insights into how the input feature space is transformed and processed as it passes through different layers of our model. To achieve this, we utilized a test dataset consisting of 204 data samples obtained from the National Oceanic and Atmospheric Administration (NOAA) active region 12381, covering a time period from 2015-07-04T21:22:24.70Z to 2015-07-13T18:22:24.70Z, with approximately 1-hour cadence data spanning around 213 hours. Beginning with the LSTM layer, we first try to analyze how the input feature space is effectively getting encoded as hidden states and then try to find a correlation between the hidden states and the forecasted flares via hidden state trajectories (Garcia et al., 2021). On analyzing the hidden state visualizations of the individual LSTM units along with the mean hidden state trajectory of all the LSTM units, we find that the mean hidden state exhibits a correlation to forecasted flares, with its values below the defined threshold for positive flares and above the threshold for negative flares. To further analyze what aspect of input feature space is triggering such changes, we employ a gradient-based feature sensitivity analysis (Li et al., 2015) and find out that the changes in the flaring history parameters impact the output of the LSTM layer the most. On further expanding the scope of our feature sensitivity analysis to encompass the entire model, we discovered that the parameters associated with flaring history, namely Edec Xmax1d, Mhis1d, Cdec, and Mdec, remained the most sensitive features. Thus, by meticulously selecting and engineering the features that relate to flaring history and optimizing the LSTM layer's sensitivity to these features, we can enhance the accuracy of the $\geq$ M-class model in predicting solar flare occurrences. Furthermore, to interpret the impact of the attention mechanism and fully connected layers and how these layers cause a change in feature importance, we utilized a gradient-based input saliency technique, guided backpropagation (Springenberg et al., 2015). On analyzing these results, we find out that the feature relevance changes after the attention layer causing the model to particularly prioritize the ARs magnetic field parameters post attention. Though the magnetic field parameters remained the most significant features even after the fully connected dense layers, their importance exhibited a relative

change. However, the area of strong field pixels in the AR remained the most important feature, and the model was still considering the total magnitude of Lorentz force along with the features related to magnetic shear as the most important features for forecasting $\geq$ M-class flares. Furthermore, many prior studies including, (Aschwanden and Freeland, 2012; Bobra and Couvidat, 2015; Li et al., 2018; Wang et al., 2017) have demonstrated congruent observations, which further substantiate the reliability and accountability of $\geq$ M-class flare forecasts made by our model. Thus to conclude, our deep learning model has shown promising results in accurately predicting $\geq$ M-class solar flares while concurrently furnishing interpretive insights into the internal mechanisms driving these predictions. These outcomes imply that our model holds substantial promise for potential deployment in operational space weather prediction systems.

Bibliography

Ahmed, O. W., Qahwaji, R., and Colak, T. (2019). A hybrid flare prediction model based on feature engineering and recurrent neural networks. *Solar Physics*, 294(1):2.

Ahmed, O. W., Qahwaji, R., Colak, T., and Higgins, P. A. (2013). Forecasting of solar flares: A review. *Solar Physics*, 283(1):157.

Ali, S. M., Saxena, S. M., and Kale, A. (2018). Guided backprop: Explaining and harnessing neural networks. In *International Conference on Intelligent Computing and Optimization*.

Aschwanden, M. J. and Freeland, S. L. (2012). Predicting solar flares from magnetic field observations. *Space Science Reviews*, 171:349–371.

Barnes, G., Leka, K. D., Schrijver, C. J., Colak, T., Qahwaji, R., Dacie, S., Higgins, P. A., Flaherty, J., and Wheatland, M. S. (2016). Solar flare forecasting using a nonlinear dynamics perspective. *The Astrophysical Journal*, 829(2):89.

Bobra, M. G. and Couvidat, S. (2015). Solar flare prediction using sdo/hmi vector magnetic field data with a machine-learning algorithm. *The Astrophysical Journal*, 798(2):135.

Bobra, M. G., Sun, X., Hoeksema, J. T., and et al. (2014). The helioseismic and magnetic imager (hmi) vector magnetic field pipeline: Sharps - space-weather hmi active region patches. *Solar Physics*, 289(11):3549–3578.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Chatterjee, S. and Mandal, S. (2019). Solar flare prediction using recurrent neural network based deep learning techniques. *arXiv preprint arXiv:1903.07866*.

Chicco, D., Tötsch, N., and Jurman, G. (2021). The matthews correlation coefficient (mcc) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. *BioData Mining*, 14(1):13.

Colak, T. and Qahwaji, R. (2009). Prediction of geomagnetic storms from coronal mass ejections. *Space Weather*, 7:S06001.

Fisher, G. H., Bercik, D. J., Welsch, B. T., and Hudson, H. S. (2012). *Solar Physics*, 277:59.

Garcia, R., Munz, T., and Weiskopf, D. (2021). Visual analytics tool for the interpretation of hidden states in recurrent neural networks. *Visual Computing for Industry, Biomedicine, and Art*, 4(24).

García, S. and Herrera, F. (2009). Index of balanced accuracy: A performance measure for skewed class distributions. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 441–448. Springer.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA.

Higgins, P. A., Gallagher, P. T., McAteer, R. T. J., and Bloomfield, D. S. (2011). A statistical study of coronal mass ejection properties and their geoeffectiveness. *Advances in Space Research*, 47(12):2105.

Huang, X., Zhang, L., Wang, H., and Li, L. (2013). Magnetohydrodynamic simulation of flare-productive active region 11158. *Astronomy & Astrophysics*, 549:A127.

Jonas, E., Bobra, M., Shankar, V., Todd Hoeksema, J., and Recht, B. (2018). Predicting Flares and Coronal Mass Ejections Using Machine Learning: The Flare Likelihood Model. *Solar Physics*, 293(3):48.

Karpathy, A., Johnson, J., and Fei-Fei, L. (2015). Visualizing and understanding recurrent networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS)*.

Li, D., Chen, H., Jing, J., Xu, Y., Wang, H., Yang, S., Wang, S., and Zhang, L. (2018). Magnetic field properties and evolution of flaring active regions leading to major solar flares in the current solar cycle. *The Astrophysical Journal*, 869(2):104.

Li, J., Luong, M.-T., and Jurafsky, D. (2015). A hierarchical neural autoencoder for paragraphs and documents. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1106–1115.

Liu, C., Deng, N., Wang, J. T., and Wang, H. (2017). The nasa flare forecasting system (fff): The geostationary operational environmental satellite (goes) x-ray flare predictor (gxfp) and flare auto-nowcaster (fan). *The Astrophysical Journal*, 843(2):104.

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

Luong, M., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In Marquez, L., editor, *Conference on Empirical Methods in Natural Language Processing*, page 1412, Stroudsburg, PA. The Association for Computational Linguistics.

Muranushi, T., Shibayama, T., Muranushi, Y. H., Isobe, H., Ishii, T. T., and Hoshino, M. (2015). A study of the large x9. 3 solar flare on 6 september 2017 and its associated space weather effects. *Space Weather*, 13(11):778–787.

Nishizuka, N., Sugiura, K., Kubo, Y., Den, M., and Ishii, M. (2017). Global distribution of solar flare occurrences. *The Astrophysical Journal*, 835(2):156.

Nishizuka, N., Sugiura, K., Kubo, Y., Den, M., and Ishii, M. (2018). Predicting solar flares with the coronal three-dimensional magnetic field. *The Astrophysical Journal*, 858(2):113.

Norsham, N. A. M. and Hamidi, Z. S. (2019). CME Event Produced Along with Solar Flares and its Relation with the Magnetic Field. *Journal of Physics: Conference Series*, 1349:012064.

Sharma, R., Mishra, S., and Kathiravan, C. (2020). Solar flare prediction using long short-term memory recurrent neural network. *Solar Physics*, 295(5):64.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. (2015). Striving for simplicity: The all convolutional net. In *International Conference on Learning Representations*.

Wang, S., Huang, N., Liu, J., Zhang, H., and Wang, Y. (2017). A statistical study of the relationship between magnetic field properties and the occurrence of m-class flares. *The Astrophysical Journal*, 834(2):110.

Winter, L. M. and Balasubramaniam, K. (2015). *Space Weather*, 13:286.

Xu, Y., Yu, L., Huang, Q., and Wang, B. (2018). A novel approach for solar flare prediction based on convolutional neural network. *The Astrophysical Journal*, 853(2):119.

Yuan, Y., Shih, F. Y., Jing, J., and Wang, H.-M. (2010). An automatic and quantitative method to identify solar active regions from soho/eit images. *Research in Astronomy and Astrophysics*, 10(8):785–798.

Zhang, Y. and Su, Y. (2018). Solar flare prediction with lstm networks and sunspot magnetic field data. *The Astrophysical Journal*, 860(2):94.