End-to-End Attention/Transformer Model for Solar Flare Prediction from Multivariate Time Series Data

Khaznah Alshammari[1], Kartik Saini[2],Shah Muhammad Hamdi[2], Soukaina Filali Boubrahimi[2]
[1]New Mexico State University,Las Cruces, NM 88001, USA;kalshamm@nmsu.edu
[2]Utah State University, Logan, UT 84322, USA; s.hamdi@usu.edu

***Abstract*—Classifying solar flares is crucial for comprehending their potential impact on space weather forecasting. In this study, we propose a novel approach to classify multivariate time series (MVTS) solar flare data using an Attention/Transformer-based framework. By utilizing the power of attention mechanisms and transformer architectures, we can capture complex temporal dependencies and interactions among different features in the time series data. Our model simultaneously attends to relevant features and learns their dependencies, enabling accurate classification of solar flare events. We evaluate our approach on a solar flare MVTS dataset and compare its performance against several state-of-the-art methods. The experimental results demonstrate that our approach achieves superior classification accuracy, outperforming existing ones. These findings highlight the effectiveness of attention mechanisms and transformer models in capturing the complex patterns in multivariate time series solar flare data. This research contributes to the advancement of solar physics and space weather forecasting, facilitating a deeper understanding of solar flare dynamics and enabling more accurate predictions to improve space weather forecasting capabilities.**

***Index Terms*—Solar flares, Multivariate time series, Attention/Transformer-based framework, Classification, Space weather forecasting**

I. Introduction

Solar flares are sudden and intense bursts of magnetic flux that occur in the solar corona and heliosphere. These events have significant consequences, including Extreme Ultra-Violet (EUV), X-ray, and gamma-ray emissions, which can have catastrophic effects on our technology-dependent society. They result in radiation exposure-based health risks for astronauts, disruption in GPS and radio communication, and damage to electronic devices. The economic impact of such extreme solar events can reach trillions of dollars [1]. To address these challenges, research efforts have been directed towards predicting and mitigating the effects of solar eruptive activities. In recent years, the heliophysics community has focused on predicting solar flares by analyzing current and historical magnetic field data from solar active regions. Although there is no direct theoretical relationship between magnetic field influx and flare occurrence in these regions, researchers rely on data science-based approaches for their predictions.

The Helioseismic Magnetic Imager (HMI) within the Solar Dynamics Observatory collects full-disk vector magnetograms that contains spatiotemporal magnetic field data of active regions. To predict solar flares, time series modeling of the magnetic field data is required. Consequently, spatiotemporal magnetic field data is mapped into multiple instances of Multivariate Time Series (MVTS) [2]. Each MVTS instance contains solar magnetic field parameters such as flux, current, helicity, and Lorentz force. The time series corresponding to these parameters are extracted based on two-time windows: the observation window (during data collection) and the prediction window (before the flare occurrence). The instances are then labeled into six classes: Q, A, B, C, M, and X. "Q" represents flare quiet active regions, while the other labels represent flaring events with increasing intensity. Notably, X and M-class flares denote the most intense flaring events.

Recent MVTS-based models have proven more effective in predicting flaring activities compared to earlier single timestamp-based magnetic field vector classification models [2]. There are two main categories of MVTS-based models targeting flare prediction. The first category is the statistical feature-based method [3], where low-dimensional representations of MVTS instances are calculated by aggregating summarization statistics of the univariate time series components. Traditional classifiers like kNN and SVM are then trained using these labeled MVTS representations. The second category involves end-to-end deep learning-based methods [4], using RNN/LSTM-based deep sequence models. These models train by sequentially feeding vectors representing magnetic field parameters into sequence model cells and optimizing the cell weights through gradient descent-based backpropagation. However, they can only utilize the time dimension of the MVTS instances, leading to sub-optimal classification performance due to limited usage of underlying patterns.

To address these limitations, Vaswani et al. [5] introduced the Transformer model, a neural network architecture based solely on self-attention mechanisms. The Transformer model revolutionized the field of natural language processing (NLP) and became the foundation for many subsequent advancements, including state-of-the-art models such as BERT [6] and GPT [7]. Its key advantage is the ability to capture long-range dependencies efficiently and in parallel, leading to faster training and inference times compared to previous models. Given the effectiveness of the transformer model, it can be a powerful choice for MVTS classification, leveraging its ability to capture long-range dependencies and handle multivariable, temporal data effectively. Thus, in this study, we aim to explore an alternative approach using attention/transformer

1946-0759/23/$31.00 ©2023 IEEE
DOI 10.1109/ICMLA58977.2023.00083

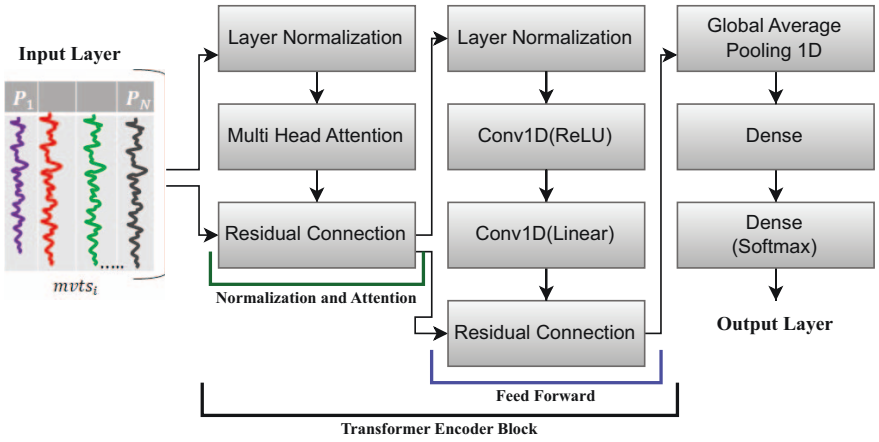

Fig. 1: Transformer/Attention Model for MVTS Classification

model-based techniques. By harnessing the power of self-attention mechanisms in transformers, we strive to capture long-range temporal dependencies among magnetic field parameters in the MVTS data, ultimately improving solar flare classification performance and deepening our understanding of these potentially catastrophic events.

1) We propose a transformer/attention-based framework for the MVTS classification.
2) We utilize the power of self-attention mechanisms to improve the MVTS classification performance.
3) Experimental results of our model demonstrate a test accuracy of 83% on the solar flare MVTS dataset when using the proposed transformer-based model, outperforming the baselines by more than 10%.

II. Related Work

In the past, flare prediction systems heavily relied on human inputs and expert knowledge. An early system called THEO, adopted by the Space Environment Center (SEC) of NOAA in 1987, required manual input of sunspots and magnetic field properties to distinguish flare classes [8]. However, with the abundance of magnetic field data collected by recent NASA missions, the focus has shifted towards data science-based approaches over purely theoretical modeling.

These data science-based approaches can be broadly categorized into linear and nonlinear statistical models. Depending on the type of dataset used, the models are further divided into line-of-sight magnetogram-based and vector magnetogram-based models. Line-of-sight magnetogram data contains only the line-of-sight component of the magnetic field, while full-disk photospheric vector magnetic field data represents solar active regions more comprehensively.

Linear statistical models aimed to identify highly correlated magnetic field features with flare occurrences. For instance, Cui et al. [9] used line-of-sight magnetogram data to establish correlation-based statistical relationships between magnetic field parameters and flare events. Even before the launch of the Solar Dynamics Observatory (SDO), Leka et al. [10] utilized linear discriminant analysis (LDA) to classify flaring events using vector magnetogram data from the Mees Solar Observatory.

On the other hand, nonlinear statistical models employed various machine learning classifiers like logistic regression, decision trees, neural networks, support vector machines (SVM), and more. For instance, Song et al. [11] and Yu et al. [12] used different classifiers on line-of-sight magnetogram-based datasets. Bobra et al. [13] employed SVM on SDO-based vector magnetogram data for flare classification, while Nishizuka et al. [14] compared the performance of kNN, SVM, and Extremely Randomized Tree (ERT) on both line-of-sight and vector magnetograms. Convolutional neural networks (ConvNets) have also been applied to SDO AIA/HMI images for solar flare prediction [15], [16].

Recently, Angryk et al. [2] introduced temporal window-based flare prediction, extending the earlier single timestamp-based models. Their MVTS-based active region dataset records magnetic field data for a preset observation time and uniform sampling rate, with each instance labeled by flare classes that occurred after a given prediction time. Hamdi et al. [17] and Muzaheed et al. [4] presented various MVTS classification approaches, including statistical summarization, decision trees, and LSTM-based deep sequence modeling. Alshammari et al. [18] addressed the future values forecasting of the magnetic field parameters, given past values in the MVTS representations.

The transformer model, introduced by Vaswani et al. [5], offers several strengths, including its ability to capture long-range dependencies, handle parallel computation, and learn contextual relationships without relying on explicit sequential processing e.g., MVTS normalization. In the context of multivariate time series (MVTS) classification, the benefits of the transformer and self-attention mechanism can be utilized due to the sequential nature of the data.

III. Methodology

A. Notations

The solar event instance i is represented by an 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 $mvts_i = \{v_{t_1}, v_{t_2}, ., ., ., v_{t_T}\}$, where $v_{t_i} \in \mathbb{R}^N$ represents a timestamp vector.

B. Data Preprocessing and Normalization

The magnetic field parameter values are recorded in different scales, so we perform z-score normalization. Z-score normalization is a technique used to transform data in such a way that it possesses a mean of zero and a standard deviation of one. By employing this method, we can effectively assess and compare the relative significance of various features within our dataset. Suppose that M number of MVTS instances each with N parameters and T time points are represented by a third-order tensor $\mathcal{X} \in \mathbb{R}^{M \times N \times T}$, where three modes represent events, parameters, and timestamps. We perform *parameter-level* z-normalization as a preprocessing step in the following three steps.

1) We perform mode-2 metrication, i.e., reshaping the tensor so that mode-2 entities (parameter) become the columns of the matrix. The matrix is denoted by $X_{(2)} \in \mathbb{R}^{MT \times N}$. The columns are denoted by $P_1, P_2, \ldots, P_N$.
2) For each column P_j, we perform z-normalization:

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

Here, $x_k^{(j)}$ is the k-th value of the column P_j, where $1 \leq k \leq MT$, $\mu^{(j)}$ is the mean of the column P_j, and $\sigma^{(j)}$ is the standard deviation of the column P_j.
3) We reshape the matrix $X_{(2)} \in \mathbb{R}^{MT \times N}$ back to third-order tensor, $\mathcal{X} \in \mathbb{R}^{M \times N \times T}$.

C. Attention-based MVTS Classification Framework

In this study, we use the attention/transformer-based model to get better performance in classifying the MVTS solar flare dataset. In our model, we create the transformer encoder block as figure 1 shows. In [5] the authors proposed a model architecture called the transformer. The transformer consists of an encoder and a decoder, both of which are composed of multiple layers of self-attention and feed-forward neural networks. The encoder processes the input sequence, such as a sentence in machine translation. It consists of a stack of identical layers, where each layer has two sub-layers:

- Self-attention layer: this layer functions by enabling each timestamp in the input time series to focus on all other time steps within the same sequence. This process enables the layer to capture temporal dependencies between individual timestamps and produce context-aware representations for each timestamp.
- Feed-forward neural network layer: after self-attention, a feed-forward neural network layer is applied to each timestamp representation independently. It introduces non-linearity and enables the model to incorporate additional information.

Algorithm 1 MVTS Transformer Encoder

function TRANSFORMER_ENCODER($inputs$, $head\_size$, num_heads, ff_dim)
  $x \leftarrow$ LAYERNORMALIZATION($inputs, \epsilon = 1e-6$)
  $x \leftarrow$ MULTIHEADATTENTION($x, x, key\_dim = head\_size, num\_heads = num\_heads$)
  $res \leftarrow x + inputs$
  $x \leftarrow$ LAYERNORMALIZATION($res, \epsilon = 1e-6$)
  $x \leftarrow$ CONV1D($x, filters = ff\_dim, kernel\_size = 1, activation = "relu"$)
  $x \leftarrow$ CONV1D($x, filters = inputs.shape[-1], kernel\_size = 1$)
  return $x + res$
end function

Algorithm 2 Build MVTS Transformer Model

function BUILD_TRANSFORMER_MODEL($input\_shape$, $head\_size, num\_heads, ff\_dim$, $num\_transformer\_blocks$, mlp_units)
  $n\_classes \leftarrow$ LENGTH($unique\_y\_train$)
  $inputs \leftarrow$ INPUT($shape = input\_shape$)
  $x \leftarrow inputs$
  for $i \leftarrow 1$ **to** $num\_transformer\_blocks$ **do**
    $x \leftarrow$ TRANSFORMER_ENCODER(x, $head\_size$, num_heads, ff_dim)
  end for
  $x \leftarrow$ GLOBALAVERAGEPOOLING1D(x, $data\_format = "channels\_first"$)
  for dim **in** mlp_units **do**
    $x \leftarrow$ DENSE($x, dim, activation = "relu"$)
  end for
  $outputs \leftarrow$ DENSE($x, n\_classes, activation = "softmax"$)
  return MODEL($inputs, outputs$)
end function

In our model, we utilize the transformer encoder block and use the benefits of the multi-head attention architecture which is a crucial component of the transformer model. It allows the model to focus on different parts of the input sequence simultaneously, enhancing its ability to capture complex temporal dependencies and extract relevant features. By using multiple attention heads, the model can learn different representations and attend to different aspects of the input data in parallel. In the context of MVTS data classification, multi-head attention offers several advantages:

- Enhanced representational capacity: by attending to different parts of the input sequence simultaneously, multi-head attention allows the model to capture both local and global dependencies effectively. This enables the model to learn complex patterns within the time series data, leading to improved classification performance.
- Robustness to variable-length sequences: MVTS data often consists of sequences with varying lengths. Multi-head attention can handle variable-length sequences efficiently by assigning different attention weights to different parts of the input. This flexibility enables the model to adapt to sequences of different lengths without compromising its classification accuracy.

The multi-head attention mechanism is a crucial component of the transformer model, allowing for the simultaneous capture of different aspects of the input sequence. It involves the computation of multiple attention heads in parallel, enabling the model to effectively process diverse information. The equations governing the multi-head attention are as follows:

- Scaled Dot-Product Attention:
  $Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$
  Here, Q, K, and V denote input matrices representing queries, keys, and values, respectively. The dimension of the key and query vectors is denoted by d_k. The attention mechanism computes the weighted sum of values based on the similarity between queries (Q) and keys (K). Linear transformations are applied to the queries, keys, and values before calculating attention weights through the dot-product operation. The $softmax$ function normalizes these weights, and the resulting weights are used to weigh the corresponding values (V) to produce the final output.
- Multi-Head Attention:
  $MultiHead(Q, K, V) = Concatenate(head_1, head_2, .., head_h)W^O$
  where $head_i$ = Attention(QW_i^Q, KW_i^K, VW_i^V). The matrices W_i^Q, W_i^K, and W_i^V represent learnable linear transformation matrices specific to the i-th attention head, while W^O is the linear transformation matrix applied to the concatenated heads. In this step, the input matrices Q, K, and V are linearly transformed separately for each attention head. The attention function is then applied to these transformed inputs to obtain the attention outputs for each head. These attention heads are again concatenated and transformed by matrix W^O to produce the final output of the multi-head attention layer [5].

Our model diagram, as shown in Figure 1, can be described as follows. The inputs undergo a series of Transformer Encoder Blocks. Each Transformer Encoder Block comprises a normalization and attention step, followed by a feed-forward step. For each Transformer Encoder Block, the input data is passed through a Layer Normalization (Encoder) block to normalize the inputs. The normalized inputs are then fed into the MultiHead Attention layer, which applies self-attention to capture dependencies between different parts of the input sequence. The output of the MultiHead Attention layer is combined with the original inputs using a Residual Sum operation, preserving the original information. The result is further passed through another Layer Normalization (Encoder) block. The output of the Layer Normalization block is fed into a 1D Convolutional layer with ReLU activation, enabling the capture of local patterns and non-linear relationships in the data. This output then passes through another 1D Convolutional layer. The output of the second 1D Convolutional layer is once again combined with the previous output using a Residual Sum operation. The final output of the transformer encoder function is obtained by summing the previous output with the input data, representing the transformed representation of the inputs. The Transformer Encoder Blocks are repeated multiple times according to the specified parameter.

After the last Transformer Encoder Block, the output is fed into a Global Average Pooling 1D layer to aggregate the features across the time dimension. Subsequently, the output of the Global Average Pooling 1D layer is passed through a series of Dense layers with $ReLU$ activation, as determined by the mlp_units parameter. The final Dense layer generates the model's outputs, with the number of units corresponding to the number of output classes, and employs the $softmax$ activation function. These outputs represent the predictions made by the model. Algorithm 1 operates as follows:

1) Layer Normalization: the tensor representation of MVTS instances is first normalized along each feature dimension by passing it through a layer normalization layer.
2) Self-Attention: the normalized tensor is then fed into a multi-head attention layer, where a self-attention mechanism is applied. Each attention head attends to different parts of the input sequence and learns to capture distinct relationships between time steps. The output of the attention layer retains the same shape as the input.
3) Residual Connection: the output of the multi-head attention layer is element-wise added to the original input tensor (inputs). This residual connection facilitates the direct flow of gradients from the input to the output, easing the learning process for the model.
4) Feed-forward: the result of the residual connection is passed through another layer normalization layer.
5) Convolutional Layer: a 1D convolutional layer with ff_dim filters and kernel size 1 is applied to the normalized tensor. This layer acts as a feed-forward neural network layer, applying non-linear transformations independently to each position in the sequence.
6) Second Convolutional Layer: another 1D convolutional layer with inputs. shape[-1] filters and kernel size 1 is applied to the result obtained from the previous layer.
7) Residual Connection: the output of the second convolutional layer is element-wise added to the result obtained from the first residual connection layer.
8) Final Output: the sum of the previous residual connection and the original input tensor (inputs) is returned as the final output.

Algorithm 2 incorporates several parameters, each described as follows: $input\_shape$ specifies the shape of the input data, $head\_size$ determines the size of each attention head in the transformer, num_heads denotes the number of attention heads in the transformer, ff_dim represents the dimension of the feed-forward network in the transformer, $num\_transformer\_blocks$ indicates the number of transformer blocks to be stacked, and mlp_units is a list of integers specifying the number of units in each MLP layer. Within the algorithm, it first determines the number of classes ($n\_classes$) based on the unique labels present in the training data. It then defines the input layer and sets it as the current layer, denoted as x. The algorithm proceeds by applying the transformer encoder block through the $transformer\_encoder$ function. After the transformer encoder blocks, a global average pooling layer is applied to reduce the spatial dimensions of the data. Subsequently, a series of MLP layers are implemented as specified by the mlp_units parameter, with each layer employing $ReLU$ activation. Finally, an output layer is added with $n\_classes$ units and a $softmax$ activation function for classification.

IV. Experiments

In this section, we present our experimental findings, where we compare the performance of our model with six other MVTS-based flare prediction baselines using a benchmark dataset. We implemented our Attention/Transformer-based MVTS classifier using TensorFlow, and the source code of our model, along with the experimental dataset, is available in our GitHub repository. [1]

A. Dataset Description

For our experiments, we utilized the benchmark dataset for MVTS-based solar flare prediction published by Angryk et al. [2]. This dataset consists of multiple MVTS instances, with each instance comprising 25-time series of active region magnetic field parameters (a comprehensive list can be found in Table 1). The time series instances are recorded at 12-minute intervals, spanning a total duration of 12 hours (60-time steps). The dataset is characterized by having 60 observation points (T) and 25 parameters (N). Our experimental dataset consists of 1,540 MVTS instances, which are evenly distributed across four flare classes: X, M, BC, and Q. Here, "Q" represents flare-quiet events, and "BC" represents a mixture of B and C class events.

B. Baseline Models

We evaluated our model with six other baselines.

1) **Flattened vector method (FLT):** this is a naive method, where each 60 × 25 MVTS instance is flattened into a 1, 500- dimensional vector.
2) **Vector of last timestamp (LTV):** this method was introduced by Bobra et al [13], 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 (25 dimensional) to train the classifier.
3) **Time series summarization-based MVTS representation (TS-SUM):** this method, proposed by Hamdi et al. [17] 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 an 8 × 25-dimensional vector space, which is used for training the downstream classifier.
4) **Long-short term memory (LSTM):** this LSTM-based approach was proposed by Muzaheed et. al. [4]. Each MVTS instance was considered as a T -length sequence of $x^{<t>} \in \mathbb{R}^N$ timestamp vectors. After sequentially feeding the LSTM model with each timestamp vector, the last hidden representation was considered as the MVTS representation. In our experiments, we set the number of cell state and hidden state dimensions to 64, the number of training epochs to 500, and the learning rate in stochastic gradient descent to 0.01.
5) **Recurrent Neural Network (RNN):** as the fifth baseline, we replace LSTM cells of the model of [4] with standard RNN cells. We use the number of RNN hidden dimensions as 128, the number of training epochs as 1,000, and the learning rate in stochastic gradient descent as 0.01.
6) **Random Convolutional Kernel Transform (ROCKET):** ROCKET was shown as the best-performing algorithm in the MVTS classification benchmarking study by Ruiz et al [19], which included 26 MVTS datasets of the UEA archive [20]. ROCKET uses a large number of random convolution kernels along with a linear classifier, where each kernel is applied to each univariate time series instance. In line with the experimental setting of Ruiz et al. [19], we set the number of kernels in ROCKET to 10,000.

The first three baselines involve embedding followed by classification methods. We use a logistic regression classifier with L2 regularization for classification. In all the baseline experiments, we split the dataset into train and test sets using the stratified holdout method, with two-thirds of the data used for training and validation, and one-third for testing.

C. Multiclass classification performance

Table I presents the classification performances of the Transformer-based MVTS classifier compared to several baseline methods. In order to provide a comprehensive evaluation, we report accuracy, precision, recall, and F1 scores for each class. The experiments were conducted using five different train/test sets, which were sampled using stratified holdout, and we report the mean and standard deviation of the results.

[1] https://github.com/Kalshammari/tramsformer-based.git

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

Measures	FLT	LTV	TS-SUM	RNN	LSTM	ROCKET	Transformer
Accuracy	0.26 ± 0.012	0.32 ± 0.02	0.61 ± 0.091	0.43 ± 0.025	0.63 ± 0.03	0.74 ± 0.02	**0.83 ± 0.026**
Precision (X)	0.23 ± 0.024	0.34 ± 0.041	0.71 ± 0.054	0.53 ± 0.031	0.76 ± 0.028	0.92 ± 0.03	**0.95 ± 0.023**
Recall (X)	0.26 ± 0.053	0.39 ± 0.043	0.77 ± 0.024	0.63 ± 0.028	0.95 ± 0.023	0.98 ± 0.01	**0.98 ± 0.008**
F1 (X)	0.24 ± 0.032	0.36 ± 0.04	0.74 ± 0.034	0.58 ± 0.019	0.84 ± 0.014	0.95 ± 0.02	**0.97 ± 0.013**
Precision (M)	0.25 ± 0.012	0.32 ± 0.033	0.52 ± 0.031	0.41 ± 0.014	0.59 ± 0.018	0.66 ± 0.04	**0.82 ± 0.051**
Recall (M)	0.26 ± 0.023	0.33 ± 0.061	0.55 ± 0.022	0.40 ± 0.03	0.54 ± 0.014	0.7 ± 0.03	**0.85 ± 0.067**
F1 (M)	0.26 ± 0.026	0.33 ± 0.042	0.53 ± 0.023	0.41 ± 0.029	0.57 ± 0.02	0.68 ± 0.02	**0.83 ± 0.026**
Precision (BC)	0.23 ± 0.044	0.26 ± 0.024	0.45 ± 0.033	0.28 ± 0.031	0.50 ± 0.013	0.58 ± 0.02	**0.71 ± 0.055**
Recall (BC)	0.24 ± 0.053	0.21 ± 0.02	0.47 ± 0.014	0.26 ± 0.021	0.41 ± 0.023	0.57 ± 0.05	**0.70 ± 0.066**
F1 (BC)	0.24 ± 0.041	0.23 ± 0.024	0.46 ± 0.041	0.27 ± 0.031	0.45 ± 0.031	0.57 ± 0.03	**0.70 ± 0.053**
Precision (Q)	0.32 ± 0.034	0.34 ± 0.044	0.58 ± 0.045	0.48 ± 0.024	0.60 ± 0.024	0.81 ± 0.04	**0.85 ± 0.056**
Recall (Q)	0.25 ± 0.042	0.36 ± 0.071	0.66 ± 0.034	0.41 ± 0.042	0.68 ± 0.023	0.72 ± 0.03	**0.78 ± 0.048**
F1 (Q)	0.28 ± 0.014	0.35 ± 0.013	0.62 ± 0.043	0.45 ± 0.032	0.64 ± 0.024	0.77 ±0.03	**0.81 ± 0.033**

The results clearly demonstrate that the Transformer-based MVTS classifier outperforms all other baselines across all performance measures. When considering the overall evaluation, ROCKET achieves the second-best performance, followed by the LSTM model in third place. Notably, the Transformer-based MVTS classifier achieves an accuracy of 20% which is higher than the LSTM model.

Among the shallow ML models, TS-SUM performs better than the FLT and LTV models. Overall, the exceptional performances of TS-SUM, RNN, LSTM, ROCKET, and our Transformer-based MVTS classifier emphasize the importance of time series representations in understanding solar events.

D. Binary classification performance

In the context of data-driven flare prediction, binary classification plays a significant role in distinguishing major flaring events from minor flaring events or flare quiet events. In this experiment, we focus on classifying X and M class MVTS instances as flaring events, while considering all other instances (Q and BC) as non-flaring events. The figure depicts the mean binary classification performances of all models over five different train/test samples. Evaluation metrics such as accuracy, precision, recall, and F1 scores are used for both the flaring and non-flaring classes.

The results clearly demonstrate that the Transformer-based MVTS model outperforms all other baseline models, and achieves an average improvement of approximately 8% compared to the second-best performing ROCKET algorithm across all performance metrics. These findings highlight the superior performance of our model in binary classification and multi-class classification. This consistency reinforces the efficacy and reliability of our Transformer-based model in accurately predicting flaring events.

E. Classification varying training set size

To investigate the adaptability of our model to larger training datasets, we conducted experiments by varying the size of the training set. The training set size was adjusted from 10% to 90% of the total dataset size, while the remaining

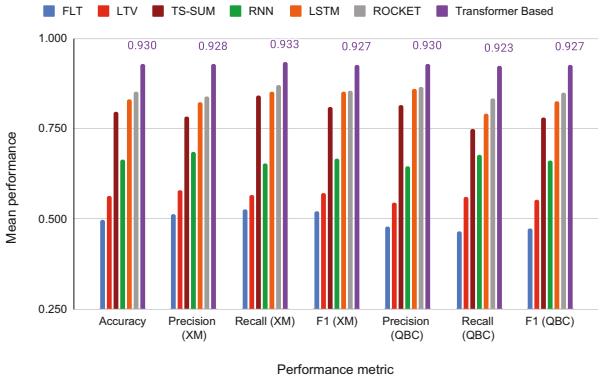

Fig. 2: Binary classification performance of all baselines.

instances were used for testing. Stratified train/test sampling was performed, and the classification performance of the classifiers was evaluated five times using distinct samples of training and test sets.

In Figure 3, we present the mean accuracy values and in Figure 4 we present the mean F1 (X class) values obtained from five runs. Across all training set sizes, our transformer-based MVTS classifier consistently outperformed the other baselines. Notably, the transformer-based MVTS model achieved a classification accuracy of 75% using only 20% of the training data, surpassing the performance of the third-best performing LSTM model, which required 90% of the training data to achieve a similar high level of performance. We observed consistent improvement patterns in deep learning and kernel-based methods, including our transformer-based model, ROCKET, LSTM, and RNN.

This observation suggests that with sufficiently large datasets, deep learning models have the potential to outperform traditional classifiers or embedding methods by a significant margin. These findings underscore the superiority of Transformer models when working with large datasets.

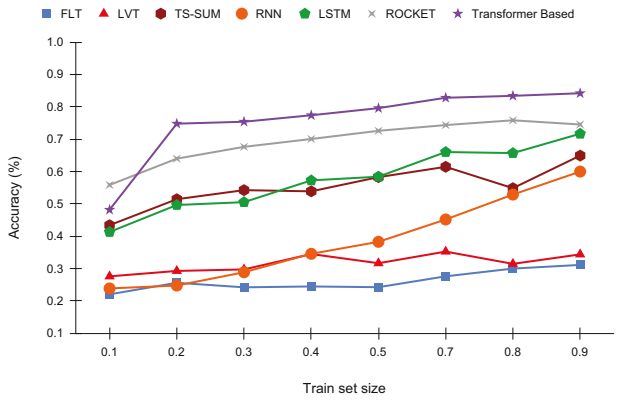

Fig. 3: Multi-class classification accuracy with increasing training data.

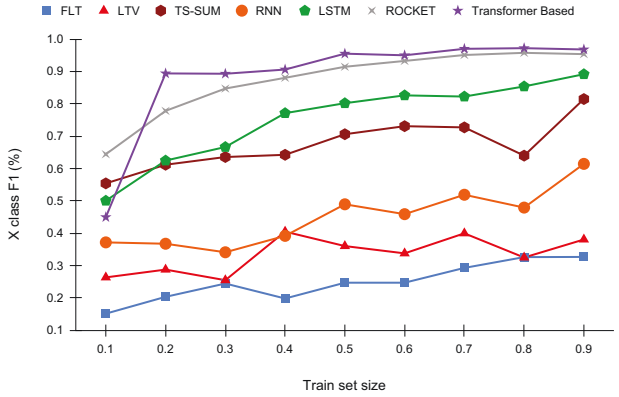

Fig. 4: F1 for X class with increasing training data

F. t-SNE Embedding performance

Visualizing high-dimensional data in 2D or 3D space using techniques like t-SNE is a well-established method for assessing the effectiveness of learned representations. In order to evaluate the quality of the learned MVTS representations, we present a visualization of the t-SNE transformed MVTS representations extracted from the final layer of the Transformer-based model. All instances are projected onto a t-SNE-reduced 2D space (see Figure 6). We employed a stratified holdout strategy for pre-training the model.

The resulting 2D projection clearly demonstrates distinct clustering of the MVTS instances. The t-SNE scatter plot provides meaningful insights, as it allows us to easily distinguish patterns among the four classes. Flare-quiet events (Q) and minor flaring events (B and C) exhibit relatively similar characteristics. On the other hand, X and M class flares show significant dissimilarity from the other classes. Additionally, we observe that certain flare-quiet events share similarities with minor flaring events, while some minor flares display characteristics similar to M-class flares. The characteristics

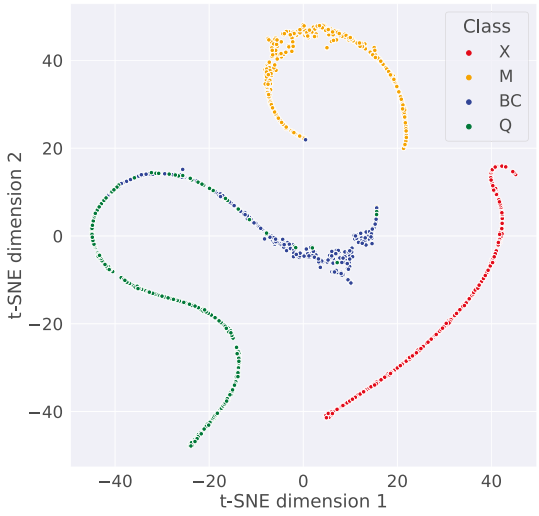

Fig. 5: t-SNE embedding of Transformer-based generated representations of all MVTS instances in the dataset

of X-class flares are distinct, with no observed similarity in instances from other classes.

By visualizing the t-SNE transformed representations, we gain valuable insights into the distinguishable patterns and similarities among the different classes of MVTS instances. This analysis allows for a deeper understanding of the learned representations and sheds light on the distinct features and characteristics of flaring events.

G. Ablation Study of the Transformer-base MVTS Classification Mode

To gain a better understanding of the contributions and effectiveness of the different layers in our model, we conducted several experiments to evaluate the significance of various aspects. Firstly, we assessed the importance of the self-attention mechanism by removing it from the model architecture and comparing the results. The removal of the attention mechanism led to a noticeable drop in accuracy, from 83% to 71%. This outcome highlights the significant role played by the Multi-Head Attention layer in capturing relevant patterns and relationships within the MVTS data.

Secondly, we examined the impact of layer normalization by removing the layer normalization layers from the model. This resulted in a decrease in accuracy from 83% to 77%. This finding underscores the importance of layer normalization in maintaining the model's performance and stability.

Lastly, we investigated the effect of the 1D convolutional layers. When these layers were removed from the model, there was a significant drop in accuracy from 83% to 71%. This result clearly demonstrates the crucial role played by the 1D convolutional layers in capturing important temporal features and contributing to the overall performance of the model.

Overall, the ablation study provided valuable insights into the contributions of different layers in our model. The

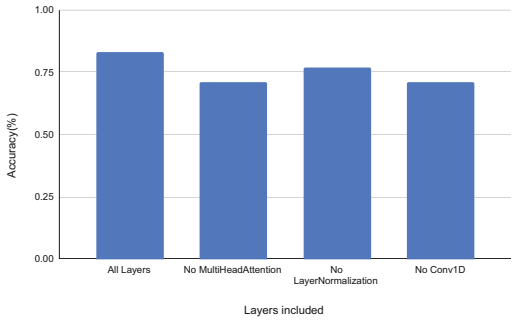

Fig. 6: Ablation Study: Revealing the Contributions of Model Components in MVTS Classification of Solar Flares.

significant decrease in accuracy upon removing the attention mechanism, layer normalization, and 1D convolutional layers highlights their importance in capturing relevant patterns, maintaining stability, and extracting essential temporal features. These findings underscore the effectiveness and significance of each layer in our model architecture.

V. Conclusion

In this work, we presented an end-to-end transformer-based flare prediction model that leverages the self-attention model for the classification of multivariate time series (MVTS) instances. Our study presents a novel approach that utilizes the strengths of the transformer model and self-attention mechanism for MVTS classification. Through an end-to-end learning process, the proposed model effectively captures the temporal relationships within MVTS instances, including higher-order inter-variable relationships and local and global temporal changes. Through the integration of attention/transformer-based techniques, our experiments on the solar flare prediction dataset demonstrate the superior performance of our model in multi-class MVTS classification, achieving an impressive accuracy of 83%. The results demonstrate the potential of our approach in providing more comprehensive and accurate predictions in the field of solar physics and space weather forecasting. This contribution holds promise for improving the overall accuracy and reliability of space weather forecasting. For future research, we plan to utilize LIME (Local Interpretable Model-Agnostic Explanations) interpretability [21] to understand the attention mechanisms in transformer models for MVTS analysis. Additionally, we aim to integrate the Graph Attention Network (GAT) [22] to construct functional networks from MVTS instances, to allow the model to learn both local and global sequences, and capture complex dependencies.

Acknowledgment

This project has been supported in part by funding from CISE and GEO directorates under NSF awards 2301397, 2305781, 2240022, and 2204363.

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?: The economic impact of space weather," *Risk Analysis*, vol. 37, 02 2017.

[2] 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*, vol. 7, no. 1, pp. 1–13, 2020.

[3] S. M. Hamdi, B. Aydin, S. F. Boubrahimi, R. Angryk, L. C. Krishnamurthy, and R. Morris, "Biomarker detection from fmri-based complete functional connectivity networks," in *2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*. IEEE, 2018, pp. 17–24.

[4] A. A. M. Muzaheed, S. M. Hamdi, and S. F. Boubrahimi, "Sequence model-based end-to-end solar flare classification from multivariate time series data," in *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2021, pp. 435–440.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[7] X. Zheng, C. Zhang, and P. C. Woodland, "Adapting gpt, gpt-2 and bert language models for speech recognition," 2021.

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

[9] 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, p. 45–59, 2006.

[10] 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.

[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, p. 101–125, 2009.

[12] 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, p. 91–105, 2009.

[13] 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.

[14] 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," *Astrophysical Journal*, vol. 835, no. 2, p. 156, 2017.

[15] X. Li, Y. Zheng, X. Wang, and L. Wang, "Predicting solar flares using a novel deep convolutional neural network," *The Astrophysical Journal*, vol. 891, no. 1, p. 10, 2020.

[16] Y. Zheng, X. Li, and X. Wang, "Solar flare prediction with the hybrid deep convolutional neural network," *The Astrophysical Journal*, vol. 885, no. 1, p. 73, 2019.

[17] 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.

[18] K. Alshammari, S. M. Hamdi, A. A. M. Muzaheed, and S. F. Boubrahimi, "Forecasting multivariate time series of the magnetic field parameters of the solar events," *CIKM workshop for Applied Machine Learning Methods for Time Series Forecasting (AMLTS)*, 2022.

[19] A. Pasos Ruiz, M. Flynn, J. Large, M. Middlehurst, and A. Bagnall, "The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowledge Discovery*, vol. 35, no. 2, pp. 401–449, 2021.

[20] A. J. Bagnall, H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. J. Keogh, "The uea multivariate time series classification archive, 2018," *arXiv preprint arXiv:1811.00075*, 2018.

[21] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?" explaining the predictions of any classifier," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1135–1144, 2016.

[22] P. Velivckovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2018.