Solar flare prediction using multivariate time series decision trees

Conference Paper · December 2017

DOI: 10.1109/BigData.2017.8258216

4 authors:

Ruizhe Ma
Georgia State University
17 PUBLICATIONS **54** CITATIONS
SEE PROFILE

Soukaina Filali Boubrahimi
Georgia State University
23 PUBLICATIONS **53** CITATIONS
SEE PROFILE

Shah Muhammad Hamdi
Georgia State University
9 PUBLICATIONS **15** CITATIONS
SEE PROFILE

Rafal A. Angryk
Georgia State University
180 PUBLICATIONS **1,080** CITATIONS
SEE PROFILE

Solar Flare Prediction using Multivariate Time Series Decision Trees

Ruizhe Ma
Department of Computer Science
Georgia State University
Atlanta, USA
rma1@student.gsu.edu

Soukaina Filali Boubrahimi
Department of Computer Science
Georgia State University
Atlanta, USA
sfilaliboubrahimi1@student.gsu.edu

Shah Muhammad Hamdi
Department of Computer Science
Georgia State University
Atlanta, USA
shamdi1@student.gsu.edu

Rafal A. Angryk
Department of Computer Science
Georgia State University
Atlanta, USA
rangryk@cs.gsu.edu

***Abstract*—Space Weather is of rising importance in scientific discipline that describes the way in which the Sun and space impact a myriad of activities down on Earth as well as the safety of the space crew members on board of the space stations. Consequently, it is imperative to better quantify the risk of future space weather events. Most of the flare prediction models in literature use physical parameters of the potentially flaring active regions during a limited interval to gain insights on whether a flare will happen or not. This limits our perception of how an event evolves for an extended duration across multiple parameters. In this paper we followed a data-driven approach to address the problem of flare prediction from a multivariate time series analysis perspective and attempt to cluster potential flaring active regions by applying Distance Density clustering on individual parameters and further organize the clustering results into a multivariate time series decision tree. We compared different data extraction priors and spans, and ranked the importance for different parameters through univariate clustering. To the best of our knowledge, this is the first attempt to predict solar flares using a tree structure.**

Keywords*—Solar Flare, univariate clustering, time series decision trees.

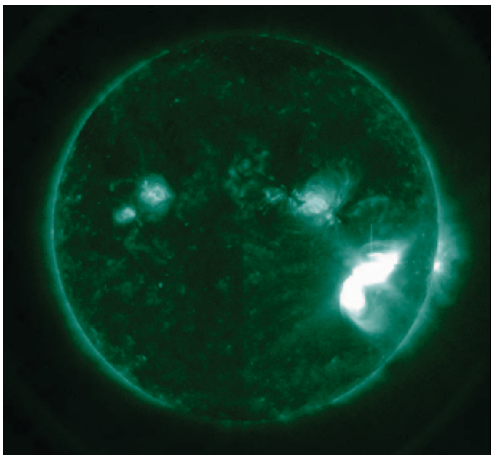

Figure 1: Example of the most intense (X9.3) flare in history as of 2017, that happened on 07-09-2017 (also known as the "solar monster")

I. Introduction

With the advancement in technology, space weather and the risks associated with it becoming more prominent. One of the latest executive orders related to space weather studies issued from the former president of the United States, urge scientists to direct attention to elaborate response plans to severe space weather conditions [1]. Solar flares and Coronal Mass Ejections (CMEs) have a direct impact on the Earth's magnetic field that can lead to geomagnetic storms. Some of the notable space weather events include the "Carrington Event" in 1859 [2], the Quebec blackout in 1989 that caused the failure of the power grid and damage to the electricity transmission system, and lately the Solar Storm of 2012 event that was not directed towards Earth but would have generated a severe geomagnetic storm, which serves as a reminder of our current vulnerability in the case of severe space weather conditions.

Different types of space weather events may have different impacts on Earth as well as to the space crew members and satellites that are in the magnetosphere [3]. For example, Solar Energetic Particles (SEP) can penetrate instruments on satellites and may be a possible threat of magnetic saturation which can eventually lead to electrical failures. Coronal Mass Ejections (CMEs) can induce extra currents in the ground which can deteriorate power grid operations down on the Earth. Additionally, X-rays heat the Earths outer atmosphere, causing it to expand and can be a disturbance to satellites. As a chain effect, space weather will also impact people who rely on those technologies. In the past CMEs were believed to be caused by solar flares, later this was found to be untrue, not every flare is accompanied with CMEs, and not every CME happens during a flare. Due to the potential damage from solar flares and CMEs, they are both actively studied by researchers, the main focus of this paper are solar flares, which can have an impact on the Earths ionosphere due to the electromagnetic radiation that are stopped by our atmosphere.

A number of physics-based model exist for predicting the flares, in this work, we took a data-driven perspective to the problem and more specifically from a multivariate time series perspective. Similar to Bobra et al. [4], we use past observations of an active region that were detected and tracked by the Space-weather HMI (Helioseismic and Magnetic Imager) Active Region Patches (SHARP) to predict its future flaring activity.

Our novelty comes from the use of multivariate time series analysis, which based on our time series cluster approach [5], we generated a decision tree used for classification purposes. Our goal is to build a binary classifier that should be able

978-1-5386-2715-0/17/$31.00 ©2017 IEEE

to distinguish whether an active region produce a flare within a particular interval of time. Previous studies in time series classification have focused on feature data where a particular parameter of a solar event is described by summaries such as: average, maximum, minimum, number of peaks, time to frequency domains transforms coefficients (i.e: Discrete Fourier Transform, Mexican wavelet transforms, etc.). Our investigation focus on the analysis of time series data and providing organization of data through clustering that could be more intuitive for human experts. Since time series data typically contain more details, we hope to gain more insight on which factors contribute more to a solar flare. Through clustering time series data we can base our findings on the detailed process that an event occurs, then based on the clustering properties of parameters we choose the most promising ones, namely the parameters which have a cleaner separation between flares and non-flares. The clustering results are aggregated and inherited by a decision tree structure, which is also illustrated with averaged time series of the corresponding parameter for additional human-friendly apprehension.

The rest of this paper is organized as follows: Section II discusses the previous work, existing methodologies, and applied datasets. Section III introduce in detail how the clustering algorithm and decision tree works. Section IV shows the experimental results, and Section V concludes this paper as well as proposing future works.

II. Background and Related Work

Most of the existing flare prediction studies use magnetic field parameters to characterize active regions [6], [7], [8], [9]. A number of active region parameters that are relevant to the flare prediction problem have been determined in literature. Schrijver et al. use the magnetic field topology parameters [6]. Fisher uses the integrated Lorentz force exerted by active regions [7]. Hagyard et al. use the shear angles properties [10]. Those parameters were designed to study the photospheric and coronal magnetic field relations when a flare happens. However, the relationship between these two aforementioned parameters is still not well understood [4]. A number of non-linear machine-learning algorithms were applied to the problem of solar flare prediction that uses either line-of-sight magnetogram or the full-disk photospheric SDO/HMI vector magnetograms. The algorithms include *k*-Nearest Neighbor [11], Support Vector Machine combined with *k*-nearest neighbors in [12], logistic regression [13], Artificial Neural Networks [14], [15], Support Vector Machines (SVM) [4], and Radial Basis Function Networks (RBFN) [16]. The main focus of these works is on the parameterization of the active regions.

With the increase of computing power and people's understanding of the importance of time series data, more studies on flare predictions are starting to focus on time series analysis. Recently, two studies dealt with time series of AR characteristics derived from HMI data. Giorgi et al. showed situations where fractal dimensions and multifractal parameters indicate flaring activities in ARs [17]. Kontogiannis et al. studied the evolution of total and maximum unsigned non-neutralized currents, and part of their conclusion stated that no single flare predictor suffices to categorically forecast solar flares [18].

In some of our previous works, the comparison of the Distance Density Clustering approach has been extensively investigated and compared with *k*-means [5], as well as Hierarchical Agglomerative Clustering (HAC) [19]. Therefore in this paper our emphasis will be implementing and experimenting with the effectiveness of our clustering approach generated decision tree when applied towards flare classification, and not the comparison against other approaches. Additionally, since one of our main focus is the model's interpretability, hard to interpret models such as neural networks or Hidden Markov Models will not be considered at this time.

In this paper, we approach the problem from a machine learning point of view. The novelty of this work comes in twofolds, on one hand, we use time series data as opposed to active region parameters aggregated over a limited period of time. The use of the active region parameters *as-is*, eliminates the feature generation step and also avoid information loss from over summarizing the time series data. On the other hand, by using a decision tree to aggregate clustering results, multiple parameters can be utilized simultaneously, as well as providing a human-friendly and easy to interpret model. The main effort of this work was focused on maximizing the prediction accuracy while keeping in mind the interpretability of the approach and results.

In the following section, we will talk about the source of the data that we will be using and how the flare and non-flare classes are extracted. Then we will introduce the applied distance measure, the clustering methodologies.

A. Data

A machine learning model analyzes historical data to project future predictions. Here previous events refer to active regions that led to a flare, which constitutes the flare class, and active regions that did not lead to any flare constitutes the non-flare class. To produce such labels, the flare catalog, that was prepared by NOAA, is consulted to find the parent active regions where the flare was initiated. As soon as a solar flare is detected from the GOES X-ray Sensor instrument (XRS), it is reported to the flare catalog and is usually paired with its parent active region. The NOAA flare catalog is a widely accepted catalog due to the continuity of the GOES missions, which started in 1974 with the launch of SMS-1 satellite and continued to today with the GOES-15. There are 5 categories of flares that are detected by the GOES satellites. The less invasive ones are class A and B, whose X-ray flux values are lesser than 10^{-6}. The non-flare class are active regions that did not lead to any flares during its lifetime of at class C and higher. Class C, M and X flares are the ones whose X-ray flux values are greater than 10^{-6}, 10^{-5} and 10^{-4} $Watts/m^2$ respectively [4], and are considered as flares in this paper.

Active regions are systematically detected and reported by the HMI team that developed a high-level data pipeline which performs this task [20], the pipeline uses the Solar Dynamics Observatory (SDO) images shown in Fig. 2. Fig. 2(a) is taken by the HMI instrument, which is used to detect an active region in the full-disk image data also known as HMI Active Region Patches (HARPs), shown in Fig. 2(b). A HARP is a bounding rectangle structure at the size scale of the containing solar active region. The last step corresponds to extracting

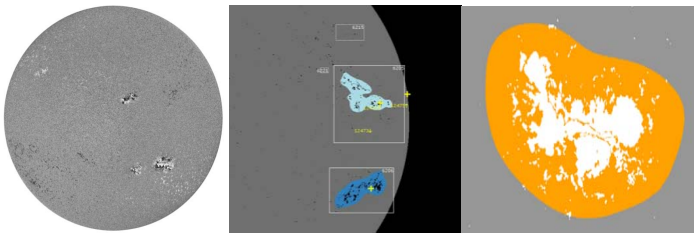

Figure 2: Illustration of the feature extraction phases steps. (a) Shows an illustration of an image taken on board of the SDO which shows the vector magnetic field that is further used to detect the active regions in (b) that are masked in (c) before computing the magnetic field related features.[21]

Table I: Active Region parameters used

ID	Tag	Description
P01	USFLUX	Total unsigned flux
P02	MEANGAM	Mean angle of field from radial
P03	MEANGBT	Mean gradient of total field
P04	MEANGBZ	Mean gradient of vertical field
P05	MEANGBH	Mean gradient of horizontal field
P06	MEANJZD	Mean vertical current density
P07	TOTUSJZ	Total unsigned vertical current
P08	MEANALP	Mean characteristic twist parameter α
P09	MEANJZH	Mean current helicity (B_z contribution)
P10	TOTUSJH	Total unsigned current helicity
P11	ABSNJZH	Absolute value of the net current helicity
P12	SAVNCPP	Sum of the modulus of the net current per polarity
P13	MEANPOT	Mean photospheric magnetic free energy
P14	TOTPOT	Total photospheric magnetic free energy density
P15	MEANSHR	Mean shear angle
P16	SHRGT45	Fraction of Area with Shear $>$ 45 deg

the vector magnetic field maps from the HARP bitmaps as shown in Fig. 2(c). The latter maps along with other magnetic field physical properties are called Space-weather HMI Active Region Patches (SHARP). The 16 parameters we used in this paper are shown in Table I. The HMI data repository is made accessible by the Joint Science Operations Center, which provides continuous measurements of the magnetic field taken from space. The non-flare class is made of active regions that never led to any flare with intensity class greater than class B. More details about how we generated experimental datasets and how we sampled the dataset into training and testing can be found in Section III-A.

B. Distance Measure

Euclidean distance is a simple distance measure that is widely used in the past with good results. However, with the wide-spread use of sequential time series data, Euclidean distance provides less satisfactory results [22], [23], [24], [25]. Using a simple example from speech recognition, different people speaking the same word would convey the exact meaning, however this could be hard to determine using Euclidean distance since the words are likely spoken with different pitch, and at different speeds. For sequential data such as time series data the general shape of data are often of more interest than the specific value.

Dynamic Time Warping (DTW) is an algorithm for measuring the similarity between two temporal sequences which may vary in time or speed. Generally, this is a method that allows computers to find a relative optimal match between two given sequences under certain restrictions. Its advantage is that it allows one-to-many mappings, and this allows one point in one sequence to be mapped to multiple points in the other sequence. Originally, DTW was used in speech recognition, later it was adapted to various real-world data mining problems. Equation 1 and Equation 2 are the distance formulas for Euclidean and DTW distances respectively, where given two time series sequences Q and C, $Q = \{q_1, q_2, ..., q_i, ..., q_n\}$, and $C = \{c_1, c_2, ..., c_j, ..., c_m\}$. When calculating the Euclidean distance, the total distance is the sum of the distance between each of the corresponding one-to-one mappings of q_i and c_i. In the case of DTW distance, an n-by-m distance matrix is first constructed, which contains the distance information between all the elements from the two sequences, often using Euclidean distance. The DTW algorithm then seeks a relatively shorter path through the distance matrix. The warping path is denoted as $W = \{w_1, w_2, ..., w_k, ..., w_K\}$, and while there are exponentially many warping paths, only the path minimizing *Dist_*$\{DTW\}$ is of interest.

$$Dist\_\{Euclidean\} = \sqrt{\sum_{i=1}^{n}(q_i - c_i)^2} \tag{1}$$

$$Dist\_\{DTW\} = min\{\sqrt{\sum_{k=1}^{K} w_k}/K \tag{2}$$

Research continues to show that DTW to be one of the best measures so far for time series data similarity measurements [26], [27], [28], [29], [30], and DTW is generally accepted as a more intuitive distance measure for time series data. As shown in Fig. 3, the mapping results from DTW are often times more natural than Euclidean distance, as it matches the phases, peaks and valleys. Therefore in this paper we will specifically be using DTW as the distance measure for time series data.

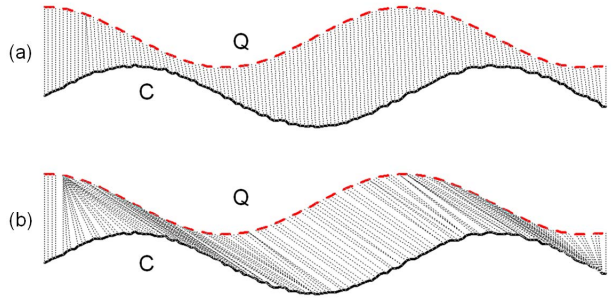

Figure 3: (a) Euclidean Mapping (top) showing one-to-one mapping and (b) DTW Mapping (bottom) showing one-to-many mapping respectively.

C. Clustering

K-means clustering was first introduced 50 years ago [31], and despite the fact that many other cluster algorithms have been proposed, it is still popular with many variations. In essence, *k*-means attempts to minimize the square of total distance between each data instance to other instances in the same cluster. Drawbacks of *k*-means include its poor performance in the presence of outliers, and unsatisfactory results when clustering non-globular data.

Based on the original *k*-means, *k*-medoids clustering was introduced by Kaufman and Rousseeuw [32], where instead of using the mean value to represent a cluster, which could very likely be an artificial data point, it uses existing data points as clustering centers. In the case of time series clusters, we find the sequence that has the shortest DTW distance to the time series average sequences. Compared to the standard *k*-means, *k*-medoids is more robust to noise and outliers. Using a medoid time series that actually occurs to represent the time series cluster is a more data-centric choice.

Although simple to execute, neither *k*-means nor *k*-medoids is known for its high accuracy. While density-based spatial clustering of applications with noise (DBSCAN) [33] can tackle some of the drawbacks of *k*-means and *k*-medoids by using the concept of density rather than distance when clustering instances. The number of clusters does not need to be specified a priori in DBSCAN, which can be useful if the dataset is not labeled. Due to its density clustering properties, it is also more robust in the presence of outliers and typically has higher accuracy than *k*-means and *k*-medoids. DBSCAN is not fully applied to time series data because the original DBSCAN algorithm uses Euclidean distance when measuring the density among the instances, which is an ineffective distance measure for time series data, and more importantly, DBSCAN is very expensive to execute, a problem which is amplified by multivariate time series data. Motivated by DBSCAN, our clustering technique we applied the idea of cluster split on sparse regions based on a distance plot. The difference is in the case of time series data, the density sparse region is theoretical and not factual.

To overcome the fluctuation of results due to random initialization, as is the case with both *k*-means and *k*-medoids, *k*-means++ was proposed by Arthur et al. [34], it is an approximation algorithm that can help to avoid some poor clustering for the original *k*-means. The idea behind the *k*-means++ algorithm is to initialize centroids (medoids) far away from each other, this speeds up the algorithm and provide better cluster results [34]. This is the inspiration behind the initialization step in our clustering method. To avoid random initialization, we picked the furthest time series based on DTW to be the initial clustering seed.

In order to cluster data, there has to be some form of representation portraying a cluster of data, such as mean/medoid in *k*-means/*k*-medoids clustering algorithms. However, averaging time series data is not a trivial problem, as local averaging methods are sensitive to computational order, and traditional averaging using Euclidean distance generate sequences not similar to the time series it is representing. Petitjean et al. proposed a global method to calculate the "average" of time series data, the DTW Barycenter Averaging (DBA) [35]. Barycenter is defined as the center of mass of two or more bodies that orbits each other. In the case of time series data, we can think of each time series sequence as a mass, and the average sequence being the orbiting center. This orbiting center is refined with DTW between time series sequences and initial average template sequence by minimizing the Within Group Sum of Squares (WGSS). WGSS is also referred to as discrepancy distance or inertia. Given a time series set $\mathbb{S} = \{S_1, S_2, ..., S_n\}$, the time series $C = \{c_1, c_2, ..., c_t\}$ is considered an average of $\mathbb{S}$ if it minimizes:

$$WGSS(C) = \sum_{k=1}^{n} dtw(C, S_n)^2 \tag{3}$$

In the original algorithm, a template sequence is selected at random. The main goal of DBA is to minimize each coordinate on the template sequence to other coordinates, thus minimizing the total WGSS. At each iteration the inertia can only decrease, and the process ends when either the algorithm converges, or the maximum number of iterations is reached. The resulting average sequence is the same length as the initial template sequence, and being a global averaging method, it is not sensitive to the order of calculation. The DBA algorithm can be used to represent a cluster of time series data the same way an averaged value can represent a group of discrete data, and also be used for further visualization purposes.

III. Methodology

In this paper we applied univariate time series clustering and multivariate time series decision tree to flare data. The Distance Density cluster methodology [5] is an extension of the combination of distance clustering and density clustering, it is especially geared towards time series data, and has shown promising results [5], [19]. In this section, we will present the Distance Density clustering algorithm and explain how it would be useful for our flare data.

A. Datasets Extraction

Table II: Detailed datasets information with prior $\in \{12, 24\}$ and span $\in \{6, 12, 24\}$

	Dataset Type	Time Period	Data		
			span 6	span 12	span 24
Prior 12	Training Data	2011, 2012, 2013, 2014	6924	6446	5683
	Testing Data	2015, 2016	2383	2185	1818
Prior 24	Training Data	2011, 2012, 2013, 2014	6743	6279	5553
	Testing Data	2015, 2016	2283	2056	1741

We extracted six datasets that corresponds to different observation periods and prior periods. The *prior* signifies the number of hours prior to the flare event occurrence. The prior can be thought of as being an interval of time in the future when our model should be able to predict the occurrence or non-occurrence of a flare. We considered prior periods as either 12 or 24 hours. In other terms we investigated the possibility of predicting a flare from an active region physical

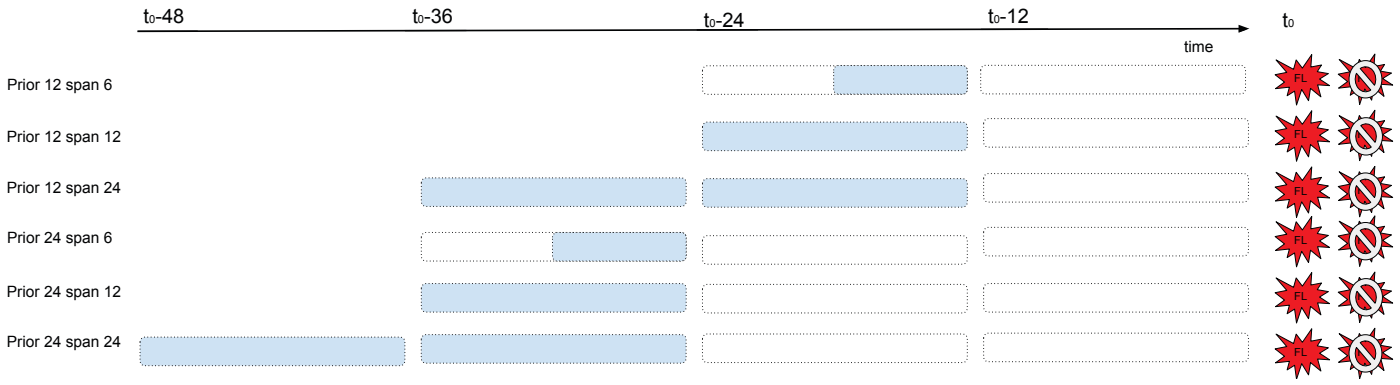

Figure 4: Prior and span combinations considered in this paper, with prior $\in \{12, 24\}$ and span $\in \{6, 12, 24\}$. The blue shaded area represents the sampling period.

characteristics 12 hours and 24 hours before their potential occurrence. The *span* signifies the observation period, in other words, the number of hours we observe the said active region. There is a direct correlation between the span duration and the length of the multivariate time series. In this paper we selected 3 different spans: 6, 12 and 24 hours and 2 different priors: 12 and 24, as shown in Table II. Fig. 4 shows all the span and priors combinations.

B. Data Sampling

In order to validate our model, splitting the data into a training set and a testing set is needed. The most common sampling methodologies include holdout and *k*-fold cross validation. Here, we used a temporal split since we are dealing with solar flares where the temporal dimension is important due to the fluctuating sun activity depending on the solar cycle. The goal of the temporal split is to ensure the robustness of the model by making sure there is no dependency on the sun's activity level in a particular time period. In conclusion, to ensure the validity and applicability, our model will be evaluated not only on *never-seen* testing data but also in data that is temporally non-overlapping with the training data time.

Algorithm 1 Furthest Seed Selection

Require:
$E = \{e_1, ..., e_n\}$ time series events to be clustered
$C_{k-1} = \{c_1, ..., c_{k-1}\}$ set of cluster seeds
$DistMtrx$ as the dissimilarity matrix of the dataset
L_k is the cluster set of events based on the number of groups

for $event_n \in L_i$ **do**
  for $event_m \in L_i$ **do**
    if $(DistMtrx[event_n, event_m] == max(DistMtrx[, event_m]))$ **then**
      $count_n = count_{n+1}$
    end if
  end for
end for
return $f_i \leftarrow event_i \leftarrow max(count_{event}, event \in L_i)$ **as the furthest event in the current cluster**
return $F = f_1, f_2, ..., f_{k-1}$ **as the furthest events for the current seeds**

C. Distance Density Clustering

Unlike *k*-means or *k*-medoids, the Distance Density clustering algorithm is a deterministic process and involves no random steps. The first seed is found by looking for the furthest time series in the dissimilarity matrix, it is done by identifying a furthest time series event for all other time series and finding the one event that has the highest occurrence in being the furthest time series, this process is described in Algorithm 1.

Algorithm 2 New Seed Selection

Require:
$E = \{e_1, ..., e_n\}$ time series events to be clustered
$C_{k-1} = \{c_1, ..., c_{k-1}\}$ set of cluster seeds
F furthest event for current seed
L_k is the cluster set of events based on the number of groups

for $l \in L_{k-1}$ **do**
  $L_{k-1} \leftarrow Cluster(C_{k-1})$
  $ar[1, 2, ..., k-1] = DistSort(L_{k-1})$
  $value[i] \leftarrow max(ar[2] - ar[1], ..., ar[k-1] - ar[k-2])$
  if $ar[n] - ar[n-1] == max(value[i])$ **then**
    $location[i] = n$
  end if
end for
if $i \leftarrow max(value[1, ..., k-1])$ **then**
  $l(i_1, i_2) \leftarrow l(i), (c_{i_1}, c_{i_2}) \leftarrow c_i$
end if

After identifying the first clustering seed, the training dataset is split into two clusters, then we obtain two medoids based on the DBA of each cluster for each cluster. As is described in Algorithm 2 we then form a distance plot for each cluster, with the distance to the furthest event ordered in an incremental fashion. The distance plots shows the distance of the furthest time series event to all the other events in that cluster, this is similar to finding the density split in DBSCAN, only in our case it is an imaginary density due to the high dimensionality of time series events. The largest increase of distance in the distance plot can be seen as a density sparse region where we consider a natural cluster split for time series data. Of the two initial clusters, the one with

a larger distance increase is chosen to be further split into two clusters, thus maintaining an addition of one cluster at each iteration. A complete description of the Distance Density clustering algorithm can be found in Algorithm 3.

Algorithm 3 Distance Density Clustering Algorithm

Require:
$E = \{e_1, ..., e_n\}$ time series events to be clustered
$C_{k-1} = \{c_1, ..., c_{k-1}\}$ set of cluster seeds
L_k is the cluster set of events based on the number of groups

1: **for** $e_i \in E$ **do**
2: $(c'_1, c'_2, ..., c'_k) \leftarrow DBA(c_1, c_2, ..., c_{i_1}, c_{i_2}, ..., c_{k-1})$
3: $UpdateClusterDBA(C_k)$
4: **end for**
5: **return** $C'_k = \{c'_1, ..., c'_k\}$ **as set of cluster seeds**
6: **return** $L_n = \{l(e) \mid = 1, 2, ..., n\}$ **set of cluster labels of** E

D. Time Series Decision Tree

When the probability of different outcomes are known, a decision process can be organized into a tree-like structure. A decision tree consists a root node, various internal nodes and leaf nodes, with the leaf nodes containing the results. Each non-leaf node corresponding to an attribute split, and the path from root to leaf is a sequence of decisions that determines which class each instance belong. An efficient decision tree is one where the concentration of a certain group of data is particularly high in the leaf nodes. Decision trees are highly interpretable models that can provide a visual decision process to users which other more complex classifiers fail to do. Although decision tree is commonly used in data mining with data of discrete attributes, it was not adapted towards time series data.

In this paper we feed the top performing individual parameter clustering results to a decision tree. This transfers hard to interpret univariate clustering results to easy to interpret multivariate decision tree structure. After clustering, decision tree serves as a flare classification application test. Compared to feature extraction, this allows decision trees to inherit some of the original time series information.

IV. Experiments

A. Univariate Clustering

Due to space limitations, all the examples in this paper is shown on the Prior 12 Span 06 dataset, which in addition to achieving high accuracy, also generates a simple and easy to maneuver decision tree. Fig. 5 shows the accuracy curve of the 16 parameters in the Prior 12 Span 06 dataset, each parameter is clustered using the Distance Density clustering method introduced in Section III individually.

In this paper, we are only doing binary clustering, meaning an event is either a flare or a non-flare. Therefore to save training time as well as avoiding over training the model, we only clustered each parameter to 20 clusters. Additionally, since our purpose is to investigate how Distance Density clustering based decision tree works with flare data, no comparison against other methods will be performed in this paper. Here each parameter name and the corresponding area under accuracy curve, which we refer to as Area Under Curve (AUC), of the training data is labeled. Each accuracy curve shows the accuracy, which is labeled on the y-axis, from clustering the time series data from 2 to 20 clusters, which is labeled on the x-axis. The training accuracy curve is a solid black line, and the testing accuracy curve is a dotted blue line, the top 5 performance parameters' accuracy curve are highlighted with a red box.

With testing data excluded from the training step, the consistent accuracy performance between training and testing signifies that the Distance Density clustering algorithm is effective and robust. Among the top five performing parameters, parameter USFLUX, TOTUSJH, and SHRGT45 appeared in all 6 priors and spans, TOTUSJZ appeared in 5 priors and spans, MEANPOT appeared 3 times, and TOTPOT and MEANSHR appeared 2 times. Other than the MEANSHR, all the afore mentioned parameters were also included in the study by Bobra et al. [4]. From the accuracy curves we can infer a knee point, in other words the point where we see the most significant increase in accuracy when cluster number is increased by 1. Since more clusters would lead to longer running time, finding the most benefit of accuracy increase while keeping minimal cluster number is desirable. The knee point for each corresponding parameter is the number of clusters we select to further train the decision tree. More analysis on parameter performance will be discussed after we show the decision trees.

Table III: Truth Table for Performance Evaluation

		Predicted Cluster	
Actual Label		MC	non-MC
	MC	True Positive (TP)	False Negative (FN)
	non-MC	False Positive (FP)	True Negative (TN)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$F - Score = 2 * \frac{precision * recall}{precision + recall} \tag{5}$$

$$precision = \frac{TP}{TP + FP} \tag{6}$$

$$recall = \frac{TP}{TP + FN} \tag{7}$$

$$R = \frac{a + b}{a + b + c + d} \tag{8}$$

The performance are evaluated using Accuracy, F-score, and Rand index, based on the truth table shown in Table III. Accuracy performance is judged by the amount of true positives and true negatives (Eq. 4). F-score (Eq. 5) is the harmonic mean of recall and precision. F-score is more resilient to class imbalance, and is a better measure than accuracy

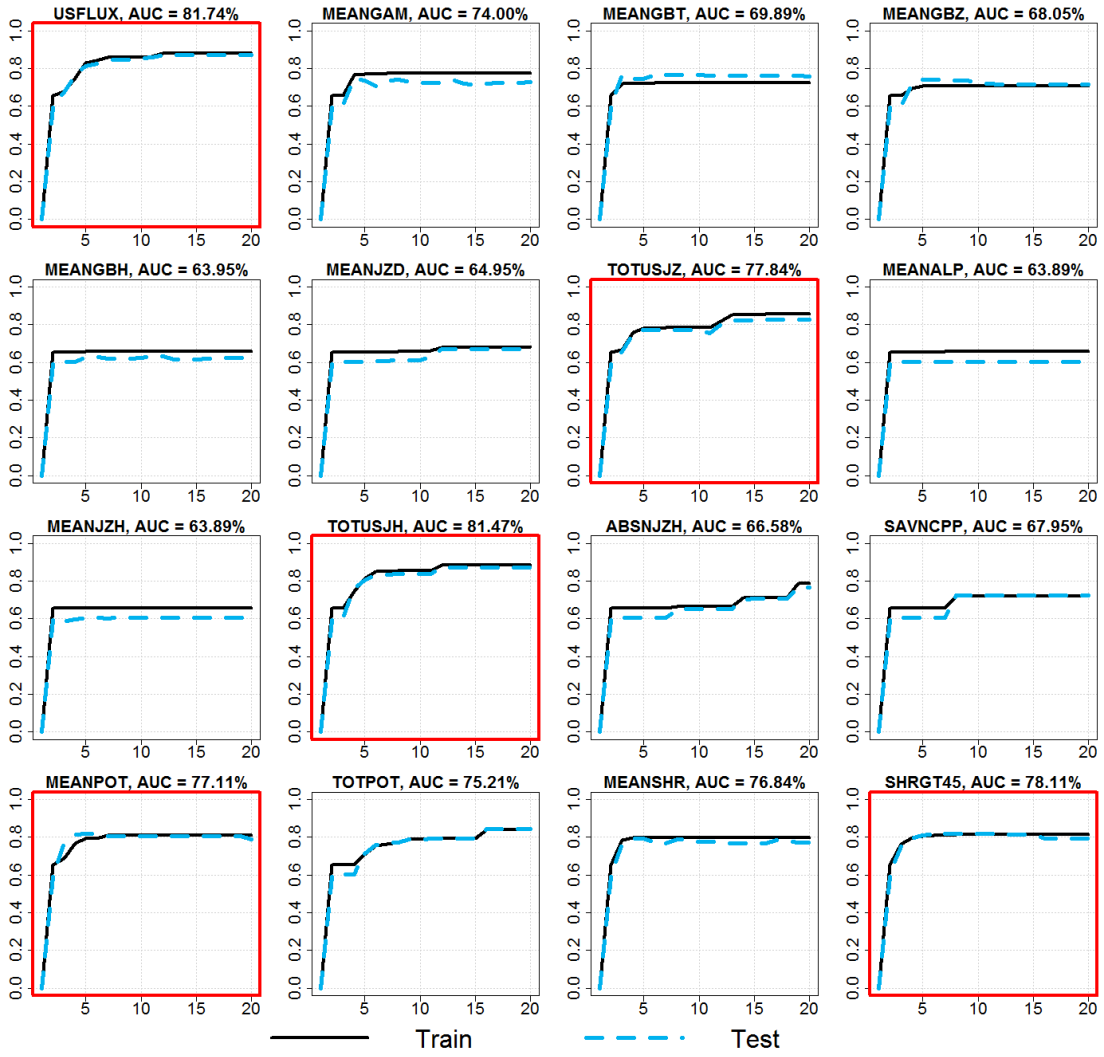

Figure 5: Accuracy curve for 16 parameters on both training and testing datasets of Prior 12 Span 06, with AUC labeled, and top five AUCs highlighted in red boxes

when false positives and false negatives have different cost and consequences. Rand index (Eq. 8) measures the agreements between two data clusterings. Given a set of elements in set S, and two partitions X and Y to compare, a is the number of pairs of elements in S that are in the same subset in X and the same subset in Y, b is the number of pairs of elements in S that are in different subsets in X and different subsets in Y, c is the number of pairs of elements in S that are in the same subset in X and in different subsets in Y, and d is the number of pairs of elements in S that are in different subsets in X and in the same subset in Y.

B. Time Series Decision Tree

The decision tree uses the parameters that can best separate flare and non-flare data, the nodes at the top of the the tree is the parameter that can split the dataset most easily. Meaning that parameters on the higher levels of the tree is in a way, more informative than the parameters on the lower levels of the tree. The decision trees for Prior 12 Span 6, Prior 12 Span 12, Prior 12 Span 24, Prior 24 Span 6, Prior 24 Span 12, Prior 24 Span 24 has three levels, two levels, four levels, three levels, one level, and two levels respectively. Typically, simpler trees avoids overfitting and ensures robustness. Fig. 6 show the decision tree for Prior 12 Span 06 dataset.

Each leaf node has the parameter used for split and the number of events at that node; each leaf node has the parameter used for split, the percentage of events out of the entire training dataset at that node, and the accuracy at that leaf. In all the decision trees as well as Fig. 6 the right-most and left-most leaf nodes corresponds to the purest flares and non-flares. The other leaf nodes has lower purity, which may be a result of only doing binary clustering and not considering all the flare classes. Nodes are colored orange if there are more flare events at that node, and colored blue if there are more non-flare events at that node. The shade of each node indicates the purity which serves as a visual aid and is not a quantified measure here, with

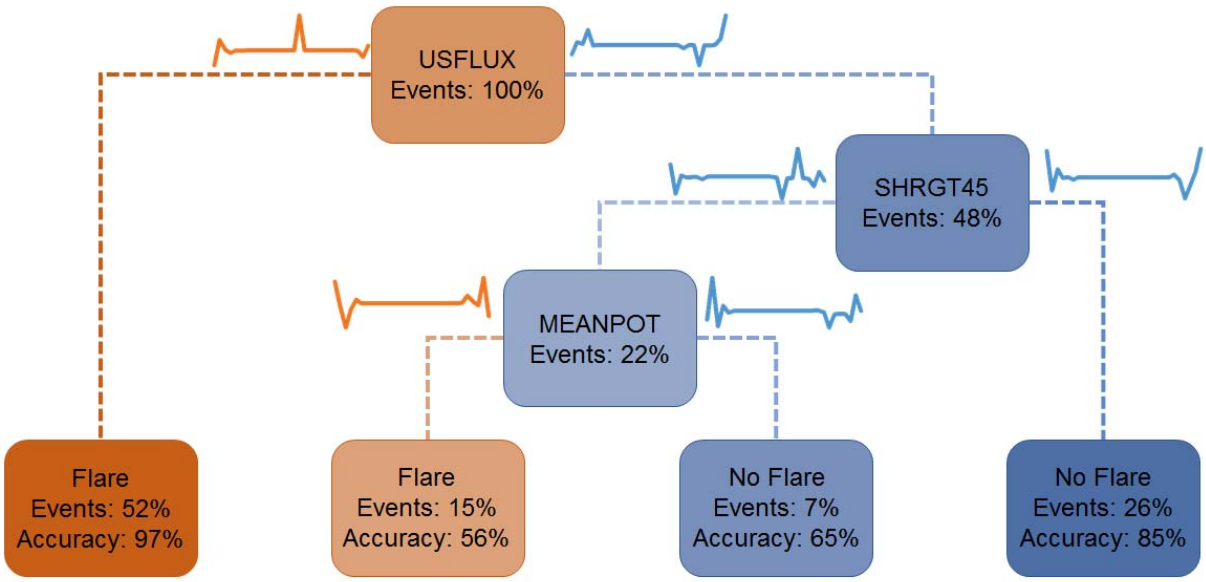

Figure 6: Multivariate decision tree for Prior 12 Span 06 data

darker being more pure, and lighter being less pure.

On each branch of the decision tree is the DBA of the child node cluster on the parent node parameter, providing the user with a more visual interpretation of how the shape of the time series data corresponding to the parameter at that node is utilized. Here all left branches are considered flares, and right branches considered non-flares.

Once the decision tree model is formulated using univariate time series clustering, it is tested by trying to classify unseen flare data. During the testing process, the time series of a new event is matched to the DBA of the corresponding parameter using DTW. Since only the parameters in the decision tree will be checked, event classification will not be time consuming. This can be helpful for future flare labeling, for example, when the initial pattern matching the average time series (DBA) of the parameter within the decision tree is identified, the event would be automatically classified as flare or non-flare. With little to no human involvement, this could improve efficiency and reduce human-errors.

For all priors and spans, either parameter USFLUX or TOTUSJZ are the root node, this corresponds to other studies on flare parameters [4]. Here either USFLUX or TOTUSJZ are able to identify around half of the entire training dataset being flare events at a 97% to 100% accuracy. Fig. 7 shows the overall Accuracy, F-score, and Rand index of the entire decision tree on all 6 priors and spans. Again, the consistency between the results from training and testing dataset proves the effectiveness and robustness of our clustering method and time series decision tree.

Generally, prior window 12 outperforms prior window 24, this is expected, since the closer we are to a flare happening the more likely it is to predict a flare. An accuracy of 85% for a simple structured decision tree on a large dataset is an acceptable accuracy at this stage for binary flare clustering. The reason is that flares are classified according to X-ray flux values, and therefore classes at the two ends of the spectrum, the M, X, and A, B flares are easier to classify. Whereas C class is more borderline, and depending on the specific event could be classified both ways, and would need additional effort for better placement.

V. Conclusion and Future Work

In this paper we applied a Distance Density clustering based, multivariate time series decision tree to flare data prediction. After splitting the original dataset into training and testing, we clustered each individual parameter in the training set. Using the 5 most promising parameters we trained a multivariate time series decision tree. Finally we use the decision tree to classify the testing dataset, and evaluated the performance using accuracy, F-score, and Rand index. The clustering step can be seen as the training section, and the classification step can be seen the testing section in our study.

Based on the experiments we found parameters USFLUX and TOTUSJZ are two of the best parameters across the board, which also concurs with other studies, and Prior 12 Span 6 dataset provides the overall highest accuracy for both training and testing dataset. From our experiments we found the consistent performance between the training and testing datasets, which substantiates the competency of our approach with Distance Density clustering and multivariate time series decision tree.

The major drawback in our current approach is clustering time expense, although once the decision tree is established, the classification of new flare data would take very little time. In the future we would like to expand our work, both toward improving our current clustering techniques for time series, as well as extending beyond binary clustering, which could improve the accuracy and make our method be more applicable in real-world applications. Additionally, now we

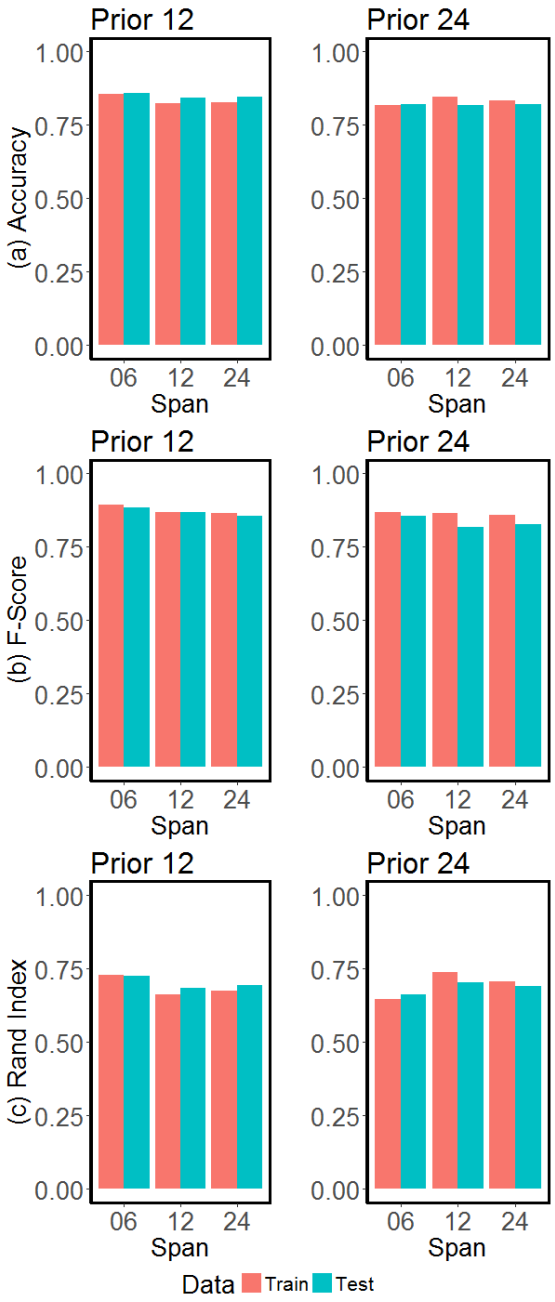

Figure 7: (a) Accuracy bar plot, (b) F-Score bar plot, and (c) Rand Index bar plot of the training and testing data of all priors and spans

have demonstrated the effect of our own clustering and classification techniques, future comparison against other approaches more known for accuracy than interpretability, such as neural networks, would also be interesting. Finding a good balance between accuracy and interpretability would be a challenge.

Acknowledgment

This project has been supported in part by funding from the Division of Advanced Cyber infrastructure within the Directorate for Computer and Information Science and Engineering, the Division of Astronomical Sciences within the Directorate for Mathematical and Physical Sciences, and the Division of Atmospheric and Geospace Sciences within the Directorate for Geosciences, under NSF award #1443061. It was also supported in part by funding from the Heliophysics Living With a Star Science Program, under NASA award #NNX15AF39G.

The SDO data is courtesy of NASA/SDO (https://sdo.gsfc.nasa.gov/) and the Atmospheric Imaging Assembly (AIA), Extreme Ultraviolet Variability Experiment (EVE), and Helioseismic and Magnetic Imager (HMI) science teams.

References

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

[2] L. W. Townsend, D. L. Stephens, J. Hoff, E. Zapp, H. Moussa, T. Miller, C. Campbell, and T. Nichols, "The carrington event: possible doses to crews in space from a comparable event," *Advances in Space Research*, vol. 38, no. 2, pp. 226–231, 2006.

[3] S. S. Board, N. R. Council *et al.*, *Severe space weather events: Understanding societal and economic impacts: A workshop report*. National Academies Press, 2009.

[4] 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.

[5] R. Ma and R. Angryk, "Distance and density clustering on time series data," 2017, In press.

[6] C. J. Schrijver, "A characteristic magnetic field pattern associated with all major solar flares and its use in flare forecasting," *The Astrophysical Journal Letters*, vol. 655, no. 2, p. L117, 2007.

[7] G. H. Fisher, D. J. Bercik, B. T. Welsch, and H. S. Hudson, "Global forces in eruptive solar flares: The lorentz force acting on the solar atmosphere and the solar interior," in *Solar Flare Magnetic Fields and Plasmas*. Springer, 2011, pp. 59–76.

[8] D. A. Falconer, R. L. Moore, A. F. Barghouty, and I. Khazanov, "Prior flaring as a complement to free magnetic energy for forecasting solar eruptions," *The Astrophysical Journal*, vol. 757, no. 1, p. 32, 2012.

[9] H. P. M. R. Bloomfield, D.S. and Gallagher, "Solar monitor's flare prediction system's (fps) probabilities are calculated using noaa space weather prediction center data combined over 1969-1976 and 1988-1996," *The Astrophysical Journal Letters, 747, L41*, 2007.

[10] G. A. Gary and M. Hagyard, "Transformation of vector magnetograms and the problems associated with the effects of perspective and the azimuthal ambiguity," *Solar physics*, vol. 126, no. 1, pp. 21–36, 1990.

[11] R. Li, H.-N. Wang, H. He, and Y.-M. Cui, "Support vector machine combined with k-nearest neighbors for solar flare forecasting," *Chinese Journal of Astronomy and Astrophysics*, vol. 7, no. 3, p. 441, 2007.

[12] R. Li, H.-N. Wang, H. He, Y.-M. Cui, and Z.-L. Du, "Support vector machine combined with k-nearest neighbors for solar flare forecasting," *Chinese Journal of Astronomy and Astrophysics*, vol. 7, no. 3, p. 441, 2007.

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

[14] R. Qahwaji and T. Colak, "Automatic short-term solar flare prediction using machine learning and sunspot associations," *Solar Physics*, vol. 241, no. 1, pp. 195–211, 2007.

[15] O. W. Ahmed, R. Qahwaji, T. Colak, P. A. Higgins, P. T. Gallagher, and D. S. Bloomfield, "Solar flare prediction using advanced feature extraction, machine learning, and feature selection," *Solar Physics*, pp. 1–19, 2013.

[16] R. Qahwaji and T. Colak, "Automatic short-term solar flare prediction using machine learning and sunspot associations," *Solar Physics*, vol. 241, no. 1, pp. 195–211, 2007.

[17] F. Giorgi, I. Ermolli, P. Romano, M. Stangalini, F. Zuccarello, and S. Criscuoli, "The signature of flare activity in multifractal measurements of active regions observed by sdo/hmi," *Solar Physics*, vol. 290, no. 2, pp. 507–525, 2015.

[18] I. Kontogiannis, M. K. Georgoulis, S.-H. Park, and J. A. Guerra, "Non-neutralized electric currents in solar active regions and flare productivity," *Solar Physics*, vol. 292, no. 11, p. 159, 2017.

[19] R. Ma, R. Angryk, and P. Riley, "Coronal mass ejection data clustering and visualization of decision trees," 2017, In press.

[20] M. G. Bobra, X. Sun, J. T. Hoeksema, M. Turmon, Y. Liu, K. Hayashi, G. Barnes, and K. Leka, "The helioseismic and magnetic imager (hmi) vector magnetic field pipeline: Sharps–space-weather hmi active region patches," *Solar Physics*, vol. 289, no. 9, pp. 3549–3578, 2014.

[21] J. T. Hoeksema, Y. Liu, K. Hayashi, X. Sun, J. Schou, S. Couvidat, A. Norton, M. Bobra, R. Centeno, K. Leka *et al.*, "The helioseismic and magnetic imager (hmi) vector magnetic field pipeline: Overview and performance," *Solar Physics*, vol. 289, no. 9, pp. 3483–3530, 2014.

[22] H. Sakoe and S. Chiba, "A dynamic programming approach to continuous speech recognition," in *Proceedings of the seventh international congress on acoustics*, vol. 3. Budapest, Hungary, 1971, pp. 65–69.

[23] C. Myers and L. Rabiner, "A level building dynamic time warping algorithm for connected word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 2, pp. 284–297, 1981.

[24] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in *KDD workshop*, vol. 10, no. 16. Seattle, WA, 1994, pp. 359–370.

[25] M. E. Munich and P. Perona, "Continuous dynamic time warping for translation-invariant curve alignment with applications to signature verification," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 1. IEEE, 1999, pp. 108–115.

[26] E. Keogh, "Exact indexing of dynamic time warping," in *Proceedings of the 28th international conference on Very Large Data Bases*. VLDB Endowment, 2002, pp. 406–417.

[27] C. A. Ratanamahatana and E. Keogh, "Three myths about dynamic time warping data mining," in *Proceedings of the 2005 SIAM International Conference on Data Mining*. SIAM, 2005, pp. 506–510.

[28] T. Giorgino *et al.*, "Computing and visualizing dynamic time warping alignments in r: the dtw package," *Journal of statistical Software*, vol. 31, no. 7, pp. 1–24, 2009.

[29] Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu, "Weighted dynamic time warping for time series classification," *Pattern Recognition*, vol. 44, no. 9, pp. 2231–2240, 2011.

[30] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh, "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 262–270.

[31] G. Nagy, "State of the art in pattern recognition," *Proceedings of the IEEE*, vol. 56, no. 5, pp. 836–863, 1968.

[32] L. Kaufman and P. Rousseeuw, *Clustering by means of medoids*. North-Holland, 1987.

[33] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[34] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[35] F. Petitjean, A. Ketterlin, and P. Gançarski, "A global averaging method for dynamic time warping, with applications to clustering," *Pattern Recognition*, vol. 44, no. 3, pp. 678–693, 2011.