Shapelet-based Temporal Association Rule Mining for Multivariate Time Series Classification

Omar Bahri, Peiyu Li, Soukaïna Filali Boubrahimi, Shah Muhammad Hamdi
Department of Computer Science, Utah State University, Logan, UT 84322
Email: {omar.bahri, peiyu.li, soukaina.boubrahimi, s.hamdi}@usu.edu

2022 IEEE International Conference on Big Data (Big Data) | 978-1-6654-8045-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/BigData55660.2022.10020478

***Abstract*—The rapid upsurge of numerical sources of information and the growth of storage capacities in recent years has resulted in the collection of massive time series datasets. Inspired by association rule mining and other rule discovery algorithms, several approaches have been proposed in the literature to discover temporal association rules from time series data. These methods place interpretability at the top of their priorities and aim to provide domain experts with relevant and qualitative rules. In this paper, we aim to fill the gap between temporal association rule mining and time series classification tasks to increase the interpretability of current classification methods. We propose rule transform (RT), a novel algorithm for multivariate time series classification (MTSC) that generates discriminative temporal rules for the sake of classification. RT generates a new feature space that represents the support of the mined temporal rules which can easily be qualitatively interpreted by domain experts. The algorithm uses Allen's Interval Algebra to extract the most prominent temporal rules from a given dataset. To our knowledge, this is the first effort to use shapelets as a unit for temporal rule mining studies for the purpose of classification. We evaluate our algorithm on the UEA archive of multivariate time series. Results show that RT produces accuracies superior to state-of-the-art time series classification algorithms with the additional advantage of interpretability.**

***Index Terms*—time series mining, multivariate time series mining, shapelets, temporal association rules**

I. Introduction

The exponential increase of time series data in terms of use and availability makes it necessary for the data mining research community to multiply the efforts to improve state-of-the-art algorithms and develop new ones [1], [2]. In the context of time series classification (TSC), while most of the efforts have been directed towards the univariate case, a limited number of algorithms have been proposed for multivariate time series classification (MTSC) [3]. TSC can be applied in a multitude of domains including aerospace, astronomy, healthcare, and finance [4]–[9]. In a recent comprehensive study benchmarking different univariate time series classification approaches and algorithms by Bagnall et al. [10], shapelet-based classification algorithms were found to be among the most powerful algorithms, in addition to being highly interpretable. In particular, the shapelet transform (ST) algorithm [11] had the second-best overall performance. The only algorithm that performed better was based on the ensemble learning technique. However, ensemble learning techniques are usually hard to interpret. In a similar study focusing on MTSC by Pasos Ruiz et al. [3], more recent algorithms were compared on the UEA archive [12]. However, the only algorithms that achieved a higher performance compared to ST were either based on ensemble learning, deep learning, or domain transformations. Nevertheless, compared to ST, these techniques are not very interpretable.

Inspired by association rule mining and other rule discovery algorithms, several approaches to extract temporal association rules from time series data have been proposed in the literature [13]–[21] These temporal relationships can then be interpreted by domain experts in order to achieve a better understanding of the problem at hand, in a purely qualitative way. An example of a temporal relation that Das et al. introduced in their paper is "A stock which follows a 2.5-week declining pattern will likely incur a short sharp fall within 4 weeks" [13]. In this work, we aim to fill the gap between rule discovery and classification in time series mining and to increase the interpretability of the current classification method. We propose rule transform (RT), an algorithm that combines shapelet-based representation learning and temporal rule finding. Using ST, we extract the most important shapelets from the time series, which we consider as our time intervals, i.e. the antecedents and subsequents of our rules. Then, we combine them using Allen's Interval [22] relationships to create temporal rules and select the most important ones based on their support. Finally, we transform the time series dataset using this set of rules and use it for classification. The motivation for this work builds on the idea that these same temporal rules which can help domain experts draw qualitative conclusions from the data can be used for classification purposes. We demonstrate its success by evaluating RT on the UEA archive of multivariate time series datasets [23].

The rest of this paper is organized as follows: In Section II, we discuss previous works in the context of shapelet-based time series classification and temporal association rules mining; in Section III, we provide technical background; in Section IV, we describe the RT algorithm; in Section V we introduce the UEA MTSC archive and discuss some implementation details; in Section VI, we evaluate the experiments; and in Section VII, we conclude with a summary.

II. Related Work

A shapelet is defined as a characteristic discriminatory, phase-independent subsequence that happens frequently in a

time series. The first shapelet-based algorithm was introduced by Ye et al. [24]. It consisted in finding all possible shapelets and using them to construct a decision tree. Rakthanmano et al. [25] introduced Fast Shapelets (FS) that improves upon the original shapelet algorithm by discretizing and approximating the shapelets using Symbolic Aggregate Approximation (SAX) [26], resulting in a significant speedup. Grabocka et al. [27] introduced Learning Time Series Shapelets (LTS). The main novelty of LTS consists in the use of gradient descent for mining shapelets. This optimization approach allows learning shapelets that are not subsequences of the original time series dataset. Fang et al. [28] introduced Efficient learning Interpretable Shapelets (ELIS), an algorithm that improves on LTS by discovering shapelets from a Piecewise Aggregate Approximation (PAA) [29] word space and developing a mechanism for computing the optimal number of shapelets. Lines et al. [30] introduced the ST algorithm which was later improved by Hills et al. [31]. ST finds all possible shapelets of given lengths and keeps the best ones in terms of their closeness to the original time series and of their discriminatory power. In addition, ST presents an advantage over the previous approaches: it separates shapelet discovery and classification. Therefore, different classifiers can be trained independently in the feature space generated by ST. In two recent studies, several time series classification methods were benchmarked including the aforementioned shapelet-based classifiers. ST proved to be one the best algorithms, and the first choice if full, inherent interpretability is required [3], [10]. Therefore, ST has been used as the shapelet mining component in other domains such as data augmentation [32] and counterfactual explanations [33], [34].

Equally important to the classification task, time series rule discovery has received less focus where most of the research efforts have been inspired by frequent pattern mining (e.g., market basket analysis). In the context of this paper, we are interested in the intersection of rule discovery and classification studies. We name the process *temporal association rule mining*. A temporal rule consists of an antecedent interval A and a subsequent interval B. In its simplest form, the relationship between the two is $precedes$, i.e. $A \rightarrow B$, meaning that A happens before B. In some works, a time lag T is introduced to the $precedes$ rule [13], [15], [16], [18], [19], which suggests the rule to become if A happens, it is very likely that B will happen within T. In addition, subsequences constraints have been added in order to include several subsequents and/or antecedents within the same rule [16] or to create sequential and cyclic rules [19]. The first step in rule discovery using time series data is to discretize the data. This can be either achieved using a sliding window to extract the time intervals and applying a clustering technique [13], [14], [16], [19], or using a PAA [35] approach [18]. Once the time series have been divided into discrete time intervals, the temporal rules are formed and pruned based on some modification of the Apriori [36], [37] algorithm [14], [15], [17]–[19] or other association rule mining techniques based on support and confidence [13], [16]. In addition, Kam and Fu [38] and Hoppner [17] used Allen's Interval relationships [22] to extend the types of rules between an antecedent and a subsequent time interval.

In this work, we propose to generalize all the proposed approaches by broadening the scope of possible interval relationships. Using temporal algebra, we extract the most prominent temporal rules from a given dataset and use them for the classification task.

TABLE I: Allen's Interval relationships

precedes	meets	overlaps	finished by	contains	starts	equals
p	m	o	F	D	s	
preceded by	met by	overlapped by	finishes	during	started by	
P	M	O	f	d	S	e

III. Background

In this section, we introduce ST and Allen's Interval Algebra, as two important elements of our approach

A. Shapelet Transform

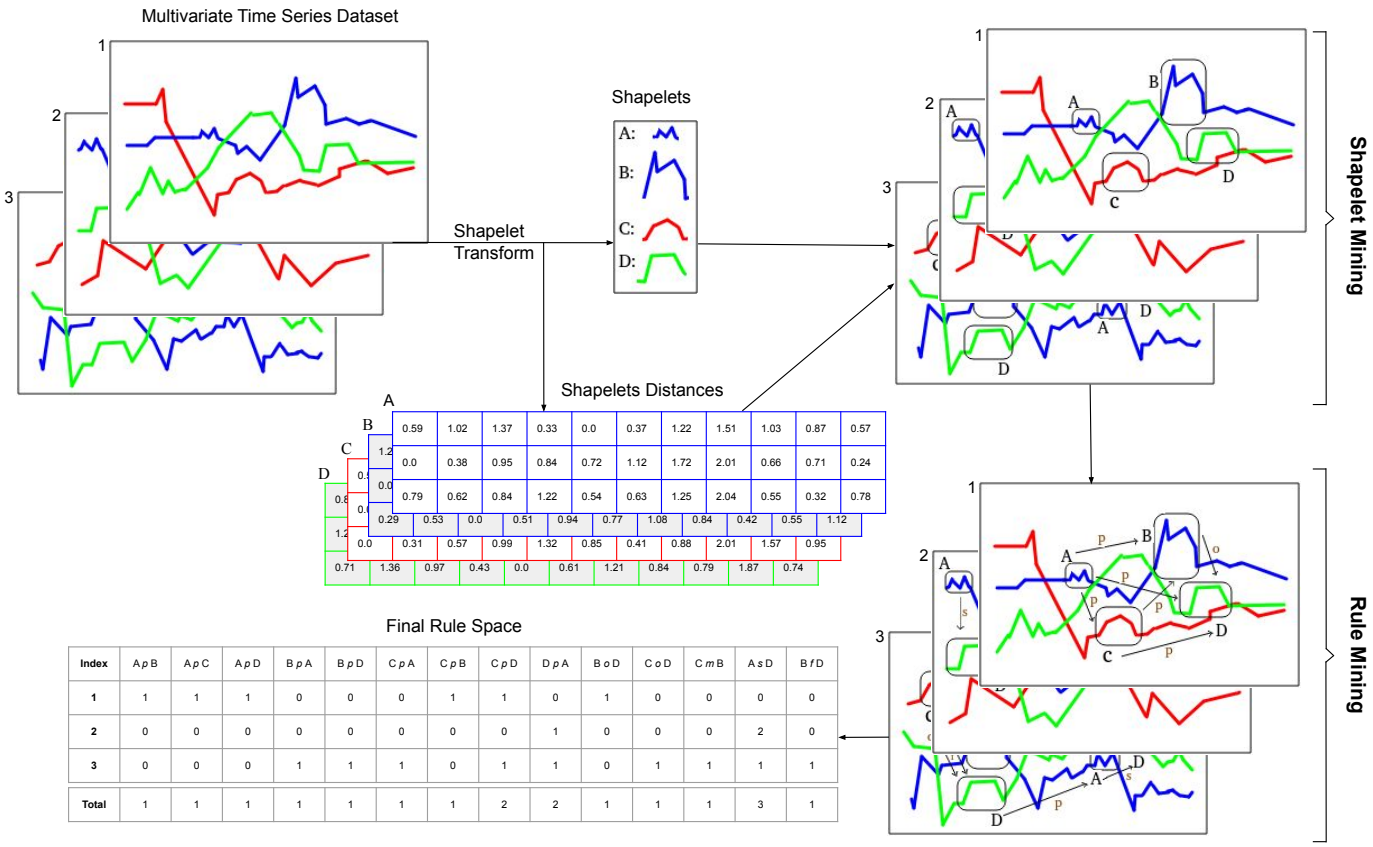
ST was introduced by Lines et al. [30] and later improved by Hills et al. [31], and by Bostrom and Bagnall [11]. In ST, shapelet discovery is performed independently from the classification task. The process starts by finding all possible candidate shapelets of predefined lengths for each time series. Then, the distance between each shapelet S_i and the other time series is computed by sliding it along the series and retrieving the minimum Euclidian distance between S_i and all subsequences w of similar length. The sliding window function is defined in Equation 1, where W is the set of all subsequences of the same length as S.

$$sDist(S,T) = min_{w \in W}(dist(S,w)) \tag{1}$$

The next step is to select the shapelets with the highest discriminative power based on the information gain of each shapelet. Overlapping shapelets are discarded and the dataset is transformed to the new shapelet feature space, where each time series is represented by its distance to the best shapelets. Finally, any machine learning classifier can be used to perform the classification based on the shapelet space.

B. Allen's Interval Algebra

Allen's Interval Algebra is a system for reasoning about temporal relations. It was introduced by James F. Allen [22]. The algebra is built on a set of 13 relationships between time intervals. These relationships are (1) distinct, meaning that no pair of antecedent and subsequent time intervals can be related by more than one "relation"; (2) exhaustive, meaning that the temporal relationship between any pair of antecedent and subsequent intervals can be described by one relation; and (3) qualitative, as no quantitative attribute is considered in their definition. Table I shows the relationships and their operator symbols (e.g., p for $precedes$ and P for $preceded\ by$).



Index	A p B	A p C	A p D	B p A	B p D	C p A	C p B	C p D	D p A	B o D	C o D	C m B	A s D	B f D
1	1	1	1	0	0	0	1	1	0	1	0	0	0	0
2	0	0	0	0	0	0	0	0	1	0	0	0	2	0
3	0	0	0	1	1	1	0	1	1	0	1	1	1	1
Total	1	1	1	1	1	1	1	2	2	1	1	1	3	1

Fig. 1: Rule Transform on a toy dataset

IV. Rule Transform

A. The Proposed Algortihm

RT aims to fill the gap between temporal association rule mining and time series classification. Since ST has proved to be one of the most efficient time series classification algorithms [10], and that shapelets represent nothing but time intervals, we designed an algorithm that relies on shapelets as building blocks for constructing rules. To our knowledge, this is the first effort to use shapelets as a unit for temporal rule mining. By definition, shapelets are phase-independent. Our idea is to introduce a notion of phase dependence by extracting relevant information from their relationships across multiple dimensions.

The main outline of the RT algorithm is described in Algorithm 1. The first step is to apply the ST algorithm in order to extract the best shapelets. This is done independently for each dimension of the multivariate time series (step 1). During ST, when computing the distance between each shapelet and dataset sample, a sliding window is used and the distances between each shapelet and all subsequences of the same length are computed (see Equation 1). We save all these distances and use them in the following steps. In step 2, the occurrences of each shapelet are found by selecting the closest subsequences —which lie under a user-defined threshold—, based on the distances computed during ST. This process might result in overlapping occurrences of the same shapelet, which are in fact only one occurrence extended by a single time step to one or both sides. Therefore, we remove the overlapping occurrences and only keep the one with the shortest distance to the original shapelet. This process is described in Algorithm 2. In step 3, all possible rules between the shapelets across all dimensions are constructed using Allen's Interval relationships. Since the relationships are symmetric, it suffices to consider seven of them, namely $A\ precedes\ B$, $A\ meets\ B$, $A\ overlaps\ B$, $A\ finishes\ B$, $A\ starts\ B$, $A\ during\ B$ and $A\ equals\ B$. Next, rules containing overlapping shapelets are dropped and the support of each rule at each dataset sample is computed as described in Algorithm 3. We define the support of a rule as the number of times it occurs in each dataset sample. Finally, the dataset is transformed to the new rule space, where each data sample is described by the support of the mined rules. After the transformation, the resulting dataset can be used to fit any standard machine learning classifier.

In Figure 1, the three steps involved in RT are illustrated and applied to a toy dataset with three samples and three dimensions. The final rule space table shows the dataset being transformed into the new rule feature space. The last row of the table represents the aggregate supports of the rules, i.e. the total number of times a rule appears throughout all the dataset samples. For example, the first sample in Figure

Algorithm 1 Rule Transform algorithm

Input: Dataset $\mathcal{D}$ of n samples and D dimensions, shapelet similarity threshold t_{sh}

Output: Dataset $\mathcal{D}$ transformed into the Rules Space

Step 1: Perform Shapelet Transform on each dimension
for d in D
    $shapelets_d$, $sh\_occurrences_d = ST(\mathcal{D}_d)$
Step 2: Select the closest occurrences of shapelets and remove overlapping ones
$sh\_locations =$ ▷ **Alg.2**
$Get\_shapelets\_occurences(\mathcal{D}, sh\_occurrences)$
Step 3: Get the supports of all rules at each sample
$all\_sups = Count\_supports(sh\_locations)$ ▷ **Alg.3**
return all_sups

1 shows the existence of four shapelets connected with six temporal relationships recorded in the final rule space table. The intuition behind our proposed RT miner is that if temporal rules can provide domain experts with qualitative insights on the data, they should also help a machine learning classifier distinguish between different classes. Going back to Figure 1 and assuming that each sample belongs to a different class, the A *precedes* B rule will help identify the class of the first sample, given that it does not occur in the second and third samples, although shapelet A is present under those.

Algorithm 2 Get_shapelets_occurences

Input: Dataset $\mathcal{D}$ of n samples and D dimensions, list of occurences of all shapelets $sh\_occurrences$, shapelet similarity threshold t_{sh}

Output: List $sh\_locations$ holding the location of each shapelet occurrence throughout all dataset samples

Create empty list $sh\_locations$ to hold the location of each shapelet occurrence throughout all $\mathcal{D}$ samples.
for d in D
    for each shapelet s
        $(idx_0,start_0,end_0,dist_0) \leftarrow sh\_occurrences_{d,s,0}$
        for $(idx,start,end,dist)$ in $sh\_occurrences_{d,s}$
            if $dist < t_{sh}$
                if $(idx,start,end) \cap (idx_0,start_0,end_0)$
                    if $dist < dist_0$
                        $sh\_locations_{d,s}.drop(-1)$
                        $sh\_locations_{d,s}.append(idx,start,end)$
                    else
                        continue
                    end if
                else
                    $sh\_locations_{d,s}.append(idx,start,end)$
                end if
            end if
            $(idx_0,start_0,end_0,dist_0) \leftarrow (idx,start,end,dist)$

Algorithm 3 Count_supports

Input: List SL holding the location of each shapelet occurrence throughout all dataset samples

Output: List all_sups holding the support of each rule at each dataset sample

Create empty lists P, O, D, M, S, F and E to hold the supports of A *precedes* B rules, A *overlaps* B rules, A *during* B rules, A *meets* B rules, A *starts* B rules, A *finishes* B rules and A *equals* B rules respectively, at each sample.
for each pair of shapelets (A, B)
    for each (i_A, s_A, e_A), (i_B, s_B, e_B), in $range\,(SL_A,\ SL_B)$
        if $i_A == i_B$
            if $s_B > s_A$
                $P_{i_A,d_A,s_A,d_B,s_B} \leftarrow P_{i_A,d_A,s_A,d_B,s_B} + 1$
            else if $s_A < s_B$ & $e_A > s_B$ & $e_A < e_B$
                $O_{i_A,d_A,s_A,d_B,s_B} \leftarrow O_{i_A,d_A,s_A,d_B,s_B} + 1$
            else if $s_A > s_B$ & $e_A < e_B$
                $D_{i_A,d_A,s_A,d_B,s_B} \leftarrow D_{i_A,d_A,s_A,d_B,s_B} + 1$
            else if $s_A < s_B$ & $e_A == s_B$ & $e_A < e_B$
                $M_{i_A,d_A,s_A,d_B,s_B} \leftarrow M_{i_A,d_A,s_A,d_B,s_B} + 1$
            else if $s_A == s_B$ & $e_A < e_B$
                $S_{i_A,d_A,s_A,d_B,s_B} \leftarrow S_{i_A,d_A,s_A,d_B,s_B} + 1$
            else if $s_A > s_B$ & $e_A == e_B$
                $F_{i_A,d_A,s_A,d_B,s_B} \leftarrow F_{i_A,d_A,s_A,d_B,s_B} + 1$
            else if $s_A == s_B$ & $e_A == e_B$ & $(d_A != d_B \| s_A != s_B)$
                $E_{i_A,d_A,s_A,d_B,s_B} \leftarrow E_{i_A,d_A,s_A,d_B,s_B} + 1$
            end if
        end if
$all\_sups = concat(P, O, D, M, S, F, E)$
return all_sups

B. Time Complexity

For a dataset $\mathcal{D}$ of n samples and m time steps, the time complexity of the ST algorithm is $\mathcal{O}(n^2m^4)$ [31]. Therefore, for D dimensions, the time complexity becomes $\mathcal{O}(Dn^2m^4)$. Concerning RT (Algorithm 1), the complexity of step 1 is the same as ST. For step 2, the complexity is $\mathcal{O}(Dn_{sh}nm)$, where n_{sh} is the number of shapelets extracted per dimension. Step 3, is the most time consuming step of the rule mining phase with a complexity of $\mathcal{O}(D^2n_{sh}^2a_{sh}^2n)$. In general, the number of dimensions D is lower than the number of dataset samples n. Therefore, $D^2 < Dn$. Also, since the number of shapelets in each dimension n_{sh} and the number of shapelets per dataset sample a_{sh} are both lower than the time length of the dataset m, then $n_{sh}^2a_{sh}^2 < m^4$. Thus, $\mathcal{O}(D^2n_{sh}^2a_{sh}^2n) < \mathcal{O}(Dn^2m^4)$, which means that the complexity of RT is upper bounded by that of ST.

C. Time Contracting

Since the time complexity of ST is $\mathcal{O}(n^2m^4)$, the running time can become very long for large datasets. Therefore, Bostrom and Bagnall [39] introduced a time contracted version of ST and proved that it does not significantly hurt the

performance of the algorithm. The main idea is to sample the shapelet space instead of going through all possible shapelets, and it is achieved by simply selecting random shapelets from the dataset until the user-defined time contract (limit) runs out. In the following experiments, we run the contracted version of ST, and adapt it to the multivariate case by dividing the time contract by the number of dimensions in the dataset. In case the number of dimensions is higher than the number of minutes in the time contract, a subset of dimensions is dropped from the dataset until the two numbers match. In addition, in order to time-bound the entire RT algorithm, we extended the same idea to the rule mining phase. In that case, instead of counting the supports of rules formed by all shapelet pairs, we sort the shapelets at each dimension by their score and mine the rules formed by the best pairs for as long as the time contract allows.

D. Shapelets Lengths Selection

To run ST, the lengths of the shapelets to be mined have to be predetermined. In order to automate this process, Lines et al. [30] proposed a length parameter approximation algorithm. The algorithm mines 10 shapelets from 10 randomly selected dataset samples. This procedure is repeated 10 times until a total of 100 shapelets is formed. Then, these shapelets are sorted by length, and the lengths of the 25th and the 75th are respectively considered as the minimum and maximum shapelet lengths. We have implemented this algorithm, with two minor modifications. First, we adapt it to the multivariate case by randomly selecting a dimension from which to select each 10 dataset samples. Then, to keep the runtime of the ST algorithm within the time bound, we use the contracted version of ST for each of the 10 runs.

E. Shapelet Clustering

The shapelet mining phase results in multiple similar, nearly duplicate shapelets. During classification, this should not cause an issue as long as the classification model can resolve correlations. However, when visualizing the rules for qualitative insights, grouping similar shapelets can improve interpretability. Therefore, we implement the post hoc shapelet clustering procedure proposed by Hills et al. [31], with minor modifications to adapt it to the multivariate time series. For each dimension, clusters containing one shapelet are formed. Then, the distances between all clusters are computed and stored in a n × n matrix, where n is the number of shapelets. Finally, we perform a recursive step where the two closest clusters are merged until the total number of clusters desired is reached. The distance between two clusters is computed as the average distance between their respective elements. As we will discuss in Section VI, while this improves the visual analysis of the rule space, it might cause performance loss.

F. Rule Selection

1) Fisher Score: The main drawback of the rule space created by RT is its high dimensionality. To reduce the number of rules used for classification and to improve the interpretability of the algorithm, we perform the Fisher score-based feature selection method [40], [41]. Moreover, we find that rule selection results in important performance gains, as we discuss in Section VI. We also show that reducing the feature space by a factor of 100 does not affect the performance.

2) Support Level: In addition, we propose an alternative rule selection method that includes computing the aggregate rule supports and discarding the set of rules with the lowest values. We define the aggregate support of a rule as the total number of times it occurs in all dataset samples. Therefore, it can be computed by summing up the supports of each rule as shown in the *Total* row in the final rule space (Figure 1).

V. Experimental Setup

A. Datasets

We evaluate the performance of RT and compare it to ST and LTS on the UEA MTSC archive [23]. The archive contains 30 multivariate time series datasets representing different domains, such as human activity recognition, motion classification, ECGclassification, EEG/MEG, audio spectra classification, and others. In addition, the datasets have a high variance in terms of dimensionality (from 2 to 1,345), time series length (from 8 to 3,000), and size (from 27 to 10,992). Four of the datasets in the archive contain series of different lengths. To avoid preprocessing issues during our benchmarking, we discard these four datasets from the study. The remaining 26 datasets are described in Table II.

B. Baselines

- *FS*: Constructs a SAX word (discrete approximation of shapelets) dictionary for each shapelet length from the time series dataset, performs dimensionality reduction using random projections, and builds frequency count histograms for each class. The best SAX words are selected according to their discriminatory power and remapped to the original shapelets [25].
- *LTS*: Shapelet discovery is performed as an optimization problem using stochastic gradient descent and a regularized logistic loss function. The final learned, optimal shapelets do not necessarily exist in the original dataset [27].
- *ELIS*: Shapelets are first discovered from the PAA space, ranked based on their TFIDF scores, and remapped to the original space. Then, they are adjusted using logistic regression and stochastic gradient descent. The length of shapelets and their total numbers do not need to be set as input parameters [28].
- *ST* [11], [30], [31]: See Section III.A.

C. Implementation Details

We used the sktime [42] implementation of ST, and introduced a slight modification to the ST implementation to extract the indices of the occurrences of each shapelet along with their distances as described in Section IV. For LTS, we used

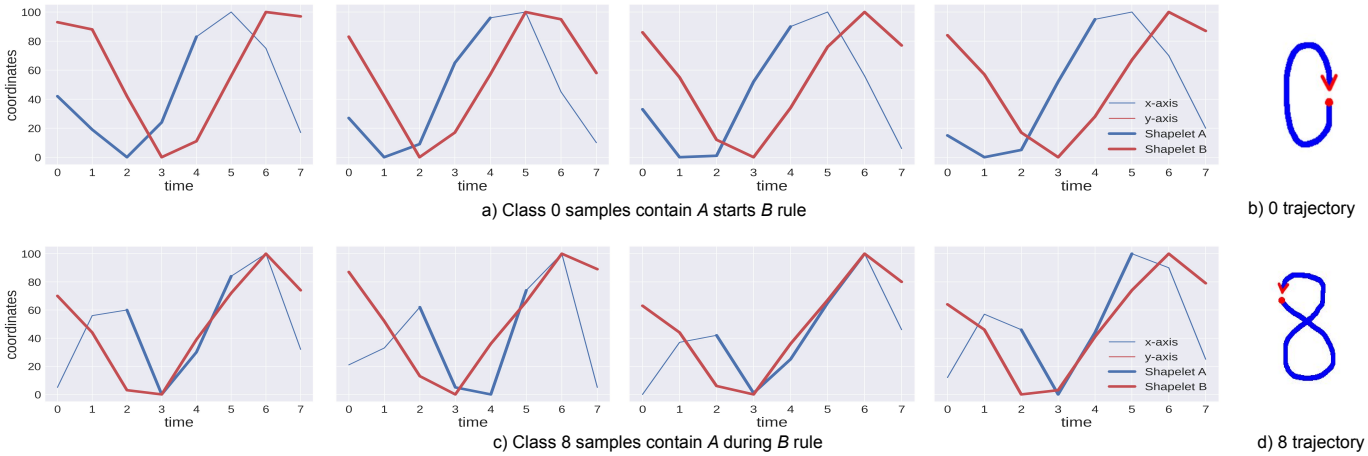


Fig. 2: Rule interpretability example from the PenDigits dataset.
All the shapelets that makeup class 0 are also contained in class 8.
The A starts B rule is present only in samples with label 0.

the implementation provided in the pyts, a Python Package for Time Series Classification [43], and for FS and ELIS, we use the code provided in the original papers [25], [28]. The code for RT is available in our GitHub repository[1] and in the project web page[2].

Given that FS, LTS, ELIS, and ST were proposed for univariate time series datasets, we adapted them to the multivariate case as follows. For ST, we apply the transformation independently to each dimension of the dataset to extract the shapelets. Then, we concatenate the resulting shapelets to form a larger feature space. Since FS, LTS, and ELIS include a built-in classifier (decision tree for FS and logistic regression for LTS and ELIS), we fit a model to each dimension of the dataset and use a majority vote to decide the final output prediction.

VI. Experimental Results

A. Rule Space Visualization

One of the main advantages of using temporal rules is their inherent interpretability. As we discussed previously, temporal rules have been used in the literature in a qualitative way only. By visualizing them, domain experts can achieve a better understanding of the problem and perhaps discover new trends and patterns. In RT, each rule of the feature space can be visualized and interpreted by domain experts. In what follows, we show a simple example from the PenDigits dataset [44] (see description in Table II). The PenDigits dataset is designed for handwritten digit classification. Each sample represents a digit (from 0 to 9) and contains two variables representing the x and y coordinates of the trajectory of the pen-tip when writing the digit. The dataset was constructed by 44 writers, and each instance was spatially resampled to 8 time steps. By looking at the plots of class 0 samples (Figure 2.a) and class 8 samples (Figure 2.c), we can see that all the shapelets that makeup class 0 are also contained in class 8, which means that a shapelet-based classifier might not be able to identify class 0 elements. On the other hand, the A starts B rule is present only in samples with the label 0. So, it can be used to discriminate class 0. In addition, we can notice that (1) the two numbers are mostly written according to the trajectories in Figure 2.b and 2.d, where the red dot represents the beginning (where the pen-tip touches the paper) and the red arrow represents the end (where the pen tip is drawn off), (2) A represents the lower arcs of 0 and 8 on the x-axis, and (3) B represents the entire numbers on the y-axis. Therefore, in the case of class 0 samples, A starts from the very beginning of the trajectory (i.e. at the same time as B), whereas it starts later in the case of 8. Moreover, the lower arc in 8 is shorter, and therefore takes less time to trace. This explains why A starts B occurs under class 0 only.

B. Performance Evaluation

We used the default train-test split for all datasets. In Table II, the accuracy of RT with different levels of rule selection using the Fisher-based method is assessed against that of FS, LTS, ELIS, and ST. For each dataset, the highest accuracy is printed in bold. The last two rows show the number of wins of each classifier and their average ranks. The time contract was fixed at 6 hours —for RT, it is broken down into 4 hours for shapelet mining and 2 hours for rule mining— and the downstream classifier used for ST and RT is the random forest (number of trees: 500). The results using the support threshold rule selection were slightly inferior. Due to the space limit, we include them in the project web page[2]. The grayed cells under FS, LTS, and ELIS represent the datasets that did not complete under the given time limit. When it comes to smaller datasets, FS and LS were the fastest of the five algorithms. However, they faced difficulties when training on larger datasets, especially on long time series instances (some of which we ran for 150+ hours without any results). Although this highly affected their overall performance, it is

[1]https://github.com/omarbahri/RuleTransform
[2]https://sites.google.com/view/ruletransform/home

TABLE II: UEA datasets description and performance comparison of FS, LTS, ELIS, ST, and RT with different rule selection thresholds on the archive

Dataset	Train Size	Test Size	Dims	TS Length	# Classes	FS	LTS	ELIS	ST	RT (by % of rules selected)						
										100%	50%	20%	10%	5%	1%	0.1%
ArticularyWordRecognition	275	300	9	144	25	88.67%	94.33%		80.67%	**99.33%**	**99.33%**	99.00%	99.00%	**99.33%**	98.67%	98.33%
AtrialFibrillation	15	15	2	640	3	33.33%	**40.00%**	26.67%	20.00%	20.00%	33.33%	26.67%	20.00%	26.67%	26.67%	13.33%
BasicMotions	40	40	6	100	4	90.00%	**100.00%**	25.00%	50.00%	**100.00%**	97.50%	**100.00%**	**100.00%**	97.50%	**100.00%**	97.50%
Cricket	108	72	6	1197	12				62.50%	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	98.61%
DuckDuckGeese	50	50	1345	270	5	**28.00%**	20.00%	20.00%	18.00%	18.00%	16.00%	18.00%	20.00%	12.00%	20.00%	18.00%
EigenWorms	128	131	6	17,984	5				53.44%	58.78%	59.54%	**60.31%**	**60.31%**	55.73%	52.67%	45.80%
Epilepsy	137	138	3	206	4	89.86%	92.75%	51.45%	84.78%	**99.28%**	98.55%	98.55%	97.83%	97.10%	96.38%	90.58%
EthanolConcentration	261	263	3	1751	4				47.15%	39.54%	38.40%	41.44%	43.73%	41.83%	48.67%	**58.56%**
ERing	30	270	4	65	6	**86.67%**	**86.67%**	84.44%	40.74%	81.85%	81.11%	83.70%	82.96%	80.74%	78.89%	52.22%
FaceDetection	5890	3524	144	62	2	**58.09%**	48.95%		51.19%	50.74%	52.07%	52.67%	50.94%	50.91%	50.77%	50.20%
FingerMovements	316	100	28	50	2	48.00%	51.00%	49.00%	53.00%	**59.00%**	54.00%	57.00%	54.00%	52.00%	52.00%	51.00%
HandMovementDirection	160	74	10	400	4	36.49%			**45.95%**	33.78%	25.68%	40.54%	39.19%	**45.95%**	35.14%	39.19%
Handwriting	150	850	3	152	26	16.94%	26.59%		5.53%	26.94%	26.35%	26.24%	27.76%	30.47%	**31.41%**	29.06%
Heartbeat	204	206	61	405	2	72.68%	72.20%		72.20%	72.68%	**73.17%**	**73.17%**	**73.17%**	73.66%	72.68%	72.20%
Libras	180	180	2	45	15	62.22%	11.11%	15.56%	61.11%	85.56%	85.00%	**86.67%**	85.56%	84.44%	77.78%	67.78%
LSST	2459	2466	6	36	14	**49.35%**		5.03%	33.90%	33.62%	34.43%	36.62%	38.00%	39.29%	40.31%	41.24%
MotorImagery	278	100	64	3000	2				50.00%	50.00%	48.00%	**51.00%**	44.00%	46.00%	**51.00%**	44.00%
NATOPS	180	180	24	51	6	75.00%	80.00%	73.33%	51.11%	**87.78%**	88.33%	87.22%	88.33%	83.89%	83.33%	82.22%
PenDigits	7494	3498	2	8	10		25.10%	10.41%	**93.48%**	62.09%	62.09%	62.01%	61.18%	52.12%	24.41%	24.41%
PEMS-SF	267	173	963	144	7	**85.55%**	11.56%		53.18%	36.42%	36.99%	35.84%	35.26%	33.53%	36.42%	32.37%
PhonemeSpectra	3315	3353	11	217	39		2.56%		8.59%	17.86%	**19.18%**	19.12%	18.25%	18.31%	14.29%	6.71%
RacketSports	151	152	6	30	4	76.32%	67.11%	28.29%	60.53%	84.87%	85.53%	84.87%	86.84%	87.50%	87.50%	84.87%
SelfRegulationSCP1	268	293	6	896	2	78.83%	73.72%		74.40%	85.67%	86.69%	86.35%	**87.37%**	86.69%	87.03%	86.69%
SelfRegulationSCP2	200	180	7	1152	2	48.89%	44.44%		55.00%	47.78%	**56.67%**	53.89%	48.89%	48.89%	52.22%	53.89%
StandWalkJump	12	15	4	2500	3	**46.67%**	33.33%		**46.67%**	40.00%	40.00%	33.33%	40.00%	40.00%	33.33%	33.33%
UWaveGestureLibrary	120	320	3	315	8	65.31%			59.38%	**87.50%**	86.56%	**87.50%**	86.88%	86.56%	86.25%	81.25%
Average Rank						6.54	7.58	9.23	6.92	4.42	3.92	**3.54**	3.62	4.46	4.54	6.46
Wins/Ties						6	3	0	3	**7**	5	**7**	5	3	4	1

worth noting that FS achieved good results on several datasets. ELIS was able to complete 11 datasets only and had the worse performance. The best performance was produced by RT with 20% of the rule space. Another important observation is that the performance of RT remains more or less stable while reducing the feature space all the way down to 1% of the total number of rules. Next, we select the RT column with the best performance from Table II (20% of the rule space) and compare it to FS, LTS, ELIS, and ST. The results of the comparison in Figure 3 show that the discriminatory power of the rule feature space created by RT allows it to outperform the other algorithms by an even larger margin (13 wins vs. 6 for FS, 4 for LTS, and 5 for ST).

C. Ablation Study

In this section, we assess the effect of the different components of RT on its overall performance. First, we consider the most basic version of the algorithm, i.e. without shapelet length selection, rule selection, and clustering. In this case, three different shapelet lengths are selected at 25%, 50%, and 75% of the time series length. Then, we include the shapelet length selection procedure. Next, we add rule selection using the support threshold method and using the Fisher-score-based method. Finally, we include shapelet clustering. We compute the average accuracy of the 5 versions over the datasets of the UEA archive and compare them in Table III. The shapelet lengths selection procedure had the highest impact. The support threshold and Fisher-score rule selection approaches improved the performance as well, with a similar effect. On the other hand, clustering resulted in an overall loss in performance. However, as discussed in Section IV.E, it fulfills its main purpose of enhancing the interpretability of the temporal rules.

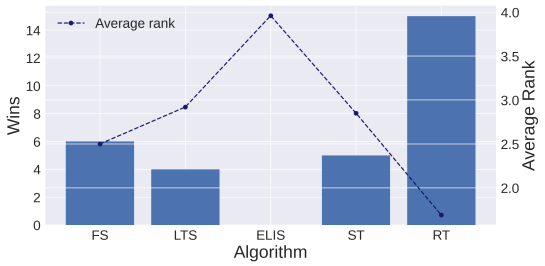


Fig. 3: Performance Comparison in the UEA MTSC Archive (RT with 20% of rule space)

D. Clustering Ratio Tuning

Clustering shapelets before rule mining improves the visual analysis of the rule space by grouping similar shapelets. In general, this might come at the cost of a slight decrease in performance. However, some datasets might also benefit from clustering at the quantitative level. In the previous section, two datasets from the UEA archive —ERing and DuckDuckGeese— showed signs of improvement with clustering. Therefore, we decided to further tune their clustering ratio. The results in Table IV and Table V show significant performance gains for both datasets.

TABLE III: Ablation study: assessing the effect of each RT component on the overall performance

RT Version	Average Accuracy (%)	Average Rank
Basic	58.05	2.92
Shapelet Lengths Selection	61.60	2.71
Shapelet Lengths Selection + Support Rule Selection	62.13	**2.33**
Shapelet Lengths Selection + Fisher Rule Selection	**62.42**	**2.33**
Shapelet Lengths Selection + Clustering	59.66	2.67

TABLE IV: ERing Clustering Ratio Tuning

Clustering Ratio (%)	No Clustering	5	10	15
Accuracy (%)	83.70	83.70	**92.59**	88.15

E. Case Study

1) Solar Flare Dataset: We perform a case study on a real-life solar flare-based multivariate time series dataset produced by a research team from Georgia State University [45]. Solar flares are defined as rare and sporadic massive eruptions of electromagnetic radiation from the sun's surface that can highly affect astronauts, space infrastructure, the atmosphere, and ground surfaces. Each sample in the dataset is made of 33 dimensions representing active region magnetic field parameters recorded at 12 minutes intervals for a duration of 12 hours (60 time steps in total), and labeled according to the largest solar flare that occurred in the next 12 hours (5 different classes: X, M, C, B, and Q). The original data is collected in the form of solar vector magnetograms, captured by the Helioseismic Magnetic Imager (HMI) [46]–[48] onboard NASA's Solar Dynamics Observatory (SDO) [49]. Since the dataset suffers from an important imbalance problem, we are using a balanced version where the B and C classes have been merged into a B/C class, and the dataset has been undersampled. The final dataset contains 1354 samples, distributed across 4 classes as shown in Table VI.

2) Performance Evaluation: First, we compare the performance of FS, LTS, ELIS, ST, and RT. The algorithms were trained using a 5-fold cross-validation, with a 6 hours time contract for each fold. ELIS did not complete in time. Table VII shows that RT achieves the highest mean accuracy. In addition, it has the second smallest standard deviation, which demonstrates the robustness of the algorithm.

3) Effect of Time Bounding: In the previous sections, we assumed that time bounding RT does not significantly hurt the performance of the algorithm. We verify this assumption by running RT without time contract. The accuracy was 74.30%, close to the 72.82% accuracy of the contracted runs.

TABLE V: DuckDuckGeese Clustering Ratio Tuning

Clustering Ratio (%)	No Clustering	10	15	20
Accuracy (%)	20.00	24.00	**36.00**	28.00

TABLE VI: Solar flare dataset class distribution

Class	**X**	**M**	**B/C**	**Q**	**All**
Number of elements	303	350	356	345	1354

TABLE VII: Performance Comparison on the Solar Flare Dataset

Algorithm	FS	LTS	ST	RT
Accuracy (%)	46.40	43.50	44.39	**72.82**
Standard Deviation	**1.60**	2.21	2.79	1.77

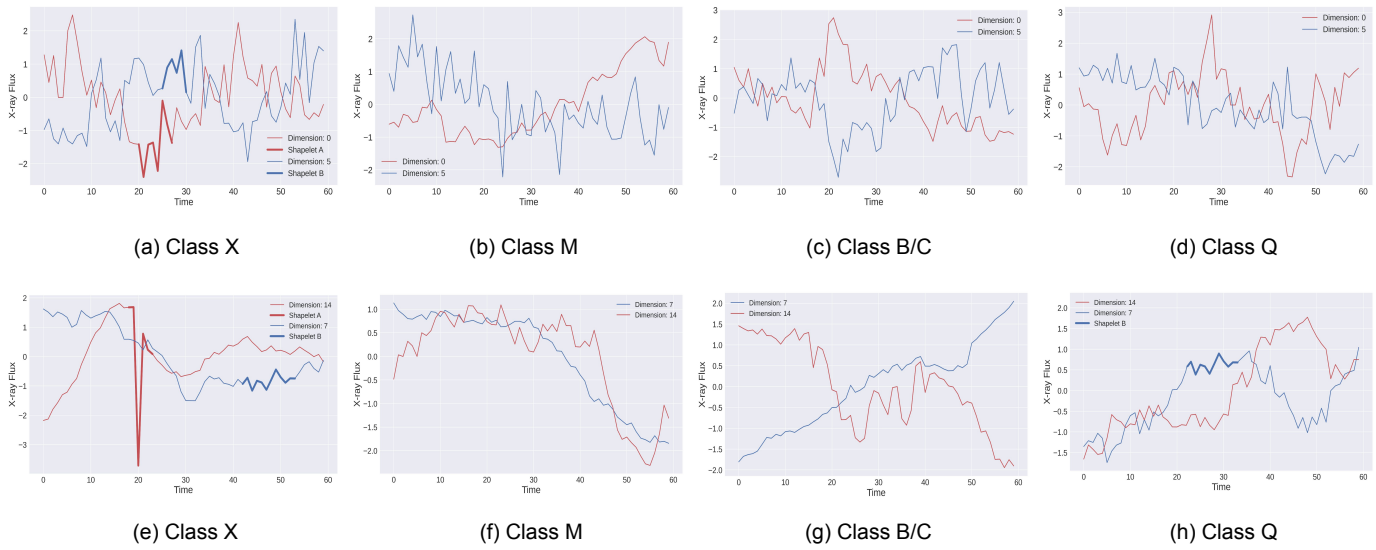
4) Rule Space Analysis:

a) Qualitative Analysis: As discussed in Section VI.A, visualizing the temporal rules extracted by RT can provide significant insights to domain experts and helps understand the decisions of the classifier. In particular, it is highly interesting to examine rules that happen uniquely under one output class. We extract such rules by considering the per-class distribution of the aggregate supports and selecting the ones with the largest distance from a random probability distribution. Figure 4 shows an $A\ overlaps\ B$ and an $A\ precedes\ B$ rule which happen uniquely under the X class. Furthermore, shapelet A in the $A\ precedes\ B$ happens also under class Q, which highlights the importance of the rule in discriminating the two classes, and in this case, in preventing damage from high-risk solar flare events.

b) Quantitative Analysis: When performing rule selection using fisher scores or simply thresholding the aggregate rule supports as described in Section IV.F, we noticed that the performance of RT is not significantly affected until the rule space is decreased by a factor of more than 100. In this section, we explore the rule space extracted from the solar flare dataset. We calculate the distributions of the rule supports by the aggregate support threshold and display the results in Figure 5. The boxplots show that dropping rules with low support increases the variation of the data at the level of the IQR. Also, lowering the support threshold increases the median value of the aggregate distribution. This confirms our previous results and proves the importance of rule selection.

VII. Conclusion

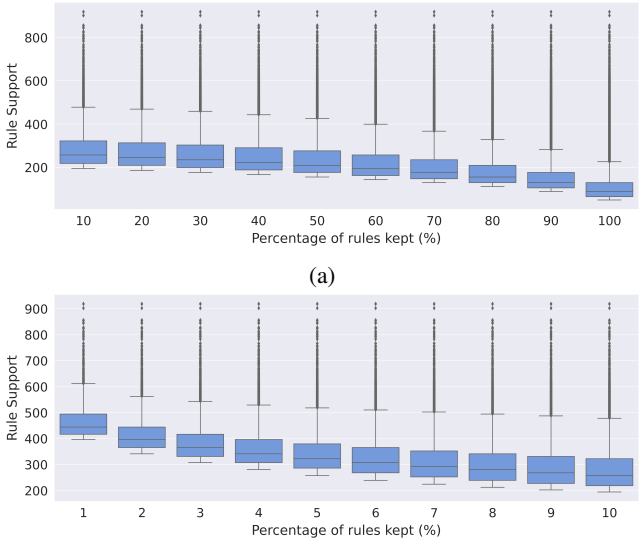
In this work, we proposed RT, an MTSC algorithm that extends a state-of-the-art shapelet-based classifier with temporal rules. By this, we aim to fill the gap between temporal association rule mining and TSC and to increase the interpretability of current classification techniques. The ideas behind this approach are: (1) using ST to extract the most discriminative shapelets from the time series dataset, (2) mining rules using these shapelets and based on Allen's Interval relationships, and (3) transforming the dataset to the new rule feature space and use it to perform classification. To the best of our knowledge, this is the first effort to use shapelets as the building blocks for temporal rule mining. We tested the proposed approach on the UEA MTSC archive and evaluated its performance against Fast Shapelets (FS), Learning Time Series Shapelets



(a) Class X (b) Class M (c) Class B/C (d) Class Q

(e) Class X (f) Class M (g) Class B/C (h) Class Q

Fig. 4: The A *overlaps* B rule in (a) occurs under the X class only, and is not present under any of the M (b), B/C (c) and Q (d) classes. The A *precedes* B rule in (e) occurs under the X class only, and is not present under any of the M (f), B/C (g) and Q (h) classes, although shapelet B is detected in class Q.



(a)

(b)

Fig. 5: Rule support distribution by support threshold: (a) from 100% to 10% (b) from 10% to 1%

(LTS), Efficient learning Interpretable Shapelets (ELIS), and Shapelet Transform (ST). The results show that our approach produces higher accuracies, with the additional interpretability edge. As a first effort to combine rule mining and classification tasks, RT holds significant promises as an interpretable model. Visualizing the temporal rules that happen exclusively under one class (or a subset of classes) will provide useful insights to domain experts.

REFERENCES

[1] E. J. Keogh, "Mining Time Series Data," in *Wiley StatsRef: Statistics Reference Online*. John Wiley & Sons, Ltd, sep 2014.

[2] R. Wu and E. J. Keogh, "Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress," sep 2020.

[3] 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, pp. 401–449, 2021.

[4] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and Information Systems*, vol. 7, no. 3, pp. 358–386, mar 2005.

[5] S. F. Boubrahimi, B. Aydin, P. Martens, and R. Angryk, "On the prediction of ¿100 MeV solar energetic particle events using GOES satellite data," in *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017*, vol. 2018-Janua. Institute of Electrical and Electronics Engineers Inc., jul 2017, pp. 2533–2542.

[6] E. Keogh and S. Kasetty, "On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration," in *Data Mining and Knowledge Discovery*, vol. 7, no. 4. Springer, oct 2003, pp. 349–371.

[7] P. Schäfer and U. Leser, "Fast and accurate time series classification with WEASEL," in *International Conference on Information and Knowledge Management, Proceedings*, vol. Part F1318. Association for Computing Machinery, nov 2017, pp. 637–646.

[8] S. Li, Y. Li, and Y. Fu, "Multi-view time series classification: A discriminative bilinear projection approach," in *International Conference on Information and Knowledge Management, Proceedings*, vol. 24-28-Octo. Association for Computing Machinery, oct 2016, pp. 989–998.

[9] C. Huang, X. Wu, X. Zhang, S. Lin, and N. V. Chawla, "Deep prototypical networks for imbalanced time series classification under data scarcity," in *International Conference on Information and Knowledge Management, Proceedings*, vol. 19. Association for Computing Machinery, nov 2019, pp. 2141–2144.

[10] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowledge Discovery*, vol. 31, no. 3, pp. 606–660, may 2017.

[11] A. Bostrom and A. Bagnall, "Binary shapelet transform for multiclass time series classification," in *Lecture Notes in Computer Science (includ-*

ing subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9263. Springer Verlag, 2015, pp. 257–269.

[12] A. B. Hoang, A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh, "The UEA multivariate time series classification archive, 2018," 2018.

[13] G. Das, K.-I. Lin, H. Mannila, G. Renganathan, and P. Smyth, "Rule Discovery from Time Series," in *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, KDD'98, Ed. New York, NY: AAAI Press, 1998, pp. 16–22.

[14] G. N. Pradhan and B. Prabhakaran, "Association Rule Mining in Multiple, Multidimensional Time Series Medical Data," *Journal of Healthcare Informatics Research*, vol. 1, no. 1, pp. 92–118, jun 2017.

[15] H. Nam, K. Y. Lee, and D. Lee, "Identification of temporal association rules from time-series microarray data sets," in *BMC Bioinformatics*, vol. 10, no. SUPPL. 3, mar 2009.

[16] S. K. Harms, J. Deogun, and T. Tadesse, "Discovering sequential association rules with constraints and time lags in multiple sequences," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2366 LNAI. Springer Verlag, 2002, pp. 432–441.

[17] F. Höppner, "Learning Temporal Rules from State Sequences," 2001.

[18] Y. Zhao and T.-t. Zhang, "Discovery of Temporal Association Rules in Multivariate Time Series," *DEStech Transactions on Computer Science and Engineering*, no. mmsta, mar 2018.

[19] T. Schluter and S. Conrad, "About the analysis of time series with temporal association rule mining," in *IEEE SSCI 2011: Symposium Series on Computational Intelligence - CIDM 2011: 2011 IEEE Symposium on Computational Intelligence and Data Mining*, 2011, pp. 325–332.

[20] M. H. Namaki, Y. Wu, Q. Song, P. Lin, and T. Ge, "Discovering Graph Temporal Association Rules," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 2017.

[21] U. Vespier, S. Nijssen, and A. Knobbe, "Mining characteristic multi-scale motifs in sensor-based time series," in *International Conference on Information and Knowledge Management, Proceedings*, 2013, pp. 2393–2398.

[22] J. F. Allen, "Maintaining Knowledge about Temporal Intervals," *Communications of the ACM*, vol. 26, no. 11, pp. 832–843, nov 1983.

[23] A. G. Bostrom, "Shapelet Transforms for Univariate and Multivariate Time Series Classification," 2018.

[24] L. Ye and E. Keogh, "Time series shapelets: A novel technique that allows accurate, interpretable and fast classification," *Data Mining and Knowledge Discovery*, vol. 22, no. 1-2, pp. 149–182, jan 2011.

[25] T. Rakthanmanon and E. Keogh, "Fast shapelets: A scalable algorithm for discovering time series shapelets," in *Proceedings of the 2013 SIAM International Conference on Data Mining, SDM 2013*. Siam Society, 2013, pp. 668–676.

[26] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, DMKD '03*. New York, New York, USA: ACM Press, 2003, pp. 2–11.

[27] J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme, "Learning time-series shapelets," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, aug 2014, pp. 392–401.

[28] Z. Fang, P. Wang, and W. Wang, "Efficient learning interpretable shapelets for accurate time series classification," *Proceedings - IEEE 34th International Conference on Data Engineering, ICDE 2018*, pp. 497–508, oct 2018.

[29] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases," *Knowledge and Information Systems*, vol. 3, no. 3, pp. 263–286, aug 2001.

[30] J. Lines, L. M. Davis, J. Hills, and A. Bagnall, "A shapelet transform for time series classification," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, New York, USA: ACM Press, 2012, pp. 289–297.

[31] J. Hills, J. Lines, E. Baranauskas, J. Mapp, and A. Bagnall, "Classification of time series by shapelet transformation," *Data Mining and Knowledge Discovery*, vol. 28, no. 4, pp. 851–881, may 2014.

[32] P. Li, S. F. Boubrahimi, and S. M. Hamdi, "Shapelets-based Data Augmentation for Time Series Classification," in *Proceedings - 20th IEEE International Conference on Machine Learning and Applications, ICMLA 2021*. Institute of Electrical and Electronics Engineers Inc., 2021, pp. 1373–1378.

[33] O. Bahri, S. F. Boubrahimi, and S. M. Hamdi, "Shapelet-Based Counterfactual Explanations for Multivariate Time Series," aug 2022, in ACM SIGKDD Workshop on Mining and Learning from Time Series (KDD-MiLeTS 2022).

[34] P. Li, S. F. Boubrahimi, and S. M. Hamdi, "Motif-guided Time Series Counterfactual Explanations," nov 2022, in 2-nd Workshop on Explainable and Ethical AI - ICPR 2022.

[35] K. Chakrabarti, E. Keogh, S. Mehrotra, and M. Pazzani, "Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases," *ACM Transactions on Database Systems*, vol. 27, no. 2, pp. 188–228, jun 2002.

[36] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," Tech. Rep., 1993.

[37] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proceedings - International Conference on Data Engineering*. IEEE, 1995, pp. 3–14.

[38] P. S. Kam and A. W. C. Fu, "Discovering temporal patterns for interval-based events," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1874. Springer Verlag, 2000, pp. 317–326.

[39] A. Bostrom and A. Bagnall, "Binary shapelet transform for multiclass time series classification," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10420 LNCS, pp. 24–46, dec 2017.

[40] X. He, D. Cai, and P. Niyogi, "Laplacian Score for Feature Selection," 1957.

[41] R. O. Duda, P. E. P. E. Hart, and D. G. Stork, *Pattern classification*, 2nd ed. Wiley, 1973.

[42] M. Löning, A. Bagnall, S. Ganesh, V. Kazakov, J. Lines, and F. J. Király, "sktime: A Unified Interface for Machine Learning with Time Series," sep 2019.

[43] J. Faouzi and H. Janati, "pyts: A python package for time series classification," *Journal of Machine Learning Research*, vol. 21, no. 46, pp. 1–6, 2020. [Online]. Available: http://jmlr.org/papers/v21/19-763.html

[44] E. Alpaydin and E. Alpaydin, "Combining multiple representations for pen-based handwritten digit recognition," *ELEKTRIK: TURKISH JOURNAL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCES*, vol. 9, p. 2001, 2001.

[45] 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, M. A. Schuh, and M. K. Georgoulis, "Multivariate time series dataset for space weather data analytics," *Scientific Data*, vol. 7, no. 1, pp. 1–13, dec 2020.

[46] J. Schou, P. H. Scherrer, R. I. Bush, R. Wachter, S. Couvidat, M. C. Rabello-Soares, R. S. Bogart, J. T. Hoeksema, Y. Liu, T. L. Duvall, D. J. Akin, B. A. Allard, J. W. Miles, R. Rairden, R. A. Shine, T. D. Tarbell, A. M. Title, C. J. Wolfson, D. F. Elmore, A. A. Norton, and S. Tomczyk, "Design and Ground Calibration of the Helioseismic and Magnetic Imager (HMI) Instrument on the Solar Dynamics Observatory (SDO)," *Solar Physics*, vol. 275, no. 1-2, pp. 229–259, jan 2012.

[47] M. G. Bobra, X. Sun, J. T. Hoeksema, M. Turmon, Y. Liu, K. Hayashi, G. Barnes, and K. D. 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.

[48] 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, M. A. Schuh, and M. K. Georgoulis, "Multivariate time series dataset for space weather data analytics," *Scientific Data*, vol. 7, no. 1, p. 227, 2020.

[49] W. D. Pesnell, B. J. Thompson, and P. C. Chamberlin, "The Solar Dynamics Observatory (SDO)," *Solar Physics*, vol. 275, no. 1-2, pp. 3–15, jan 2012.