

A multi-scale smoothing kernel for measuring time-series similarity

A. Troncoso^a, M. Arias^b, J. C. Riquelme^c

^a*Department of Computer Science, Pablo de Olavide University of Seville, Spain, ali@upo.es*

^b*Department of Computer Science, Universitat Politècnica de Catalunya, Spain, marias@lsi.upc.edu*

^c*Department of Computer Science, University of Seville, Spain, riquelme@us.es*

Abstract

In this paper a kernel for time-series data is introduced so that it can be used for any data mining task that relies on a similarity or distance metric. The main idea of our kernel is that it should recognize as highly similar time-series that are essentially the same but may be slightly perturbed from each other: for example, if one series is shifted with respect to the other or if it slightly misaligned. Namely, our kernel tries to focus on the shape of the time-series and ignores small perturbations such as misalignments or shifts. First, a recursive formulation of the kernel directly based on its definition is proposed. Then it is shown how to efficiently compute the kernel using an equivalent matrix-based formulation. To validate the proposed kernel three experiments have been carried out. As an initial step, several synthetic datasets have been generated from UCR time-series repository and the KDD challenge of 2007 with the purpose of validating the kernel-derived distance over shifted time-series. Also, the kernel has been applied to the original UCR time-series to analyze its potential in time-series classification in conjunction with Support Vector Machines. Finally, two real-world applications related to ozone concentration in atmosphere and electricity demand have been considered.

Keywords: kernel, similarity, distance, time-series classification

1. Introduction

Time-series analysis is an important problem with application in domains as diverse as engineering, medicine, astronomy or finance [11, 29]. In particular, the problem of time-series classification and prediction is attracting a lot of attention among researchers. One of the most successful and popular methods for classification and prediction are kernel-based methods suchlike as support vector machines (SVM) [26, 12, 35, 25]. Despite their popularity, there seem to be only a handful of kernels designed for time-series. This paper tries to fill this gap, and proposes a kernel exclusively designed for time-series. Moreover, using a standard trick, we are able to convert our kernel into a distance for time-series, therefore allowing us to use our kernel in distance-based algorithms as well.

A crucial aspect when dealing with time-series is to find a good measure, either a kernel similarity or a distance, that captures the essence of the time-series according to the domain of application.

For example, Euclidean distance between time-

series is commonly used due to its computational efficiency; however, it is very brittle and small shifts in one time-series can result in huge changes in the Euclidean distance. Therefore, more sophisticated distances have been devised and designed to be more robust to small fluctuations of the input time-series. Notably, Dynamic Time Warping (DTW) [30] is held as the state-of-the-art method for comparing the similarity among time-series. The DTW is very powerful in the sense that it can deal optimally with contractions, expansions and shifts in time-series in addition to being able to handle time-series of different lengths. Unfortunately, computing the DTW distance is prohibitively costly for many practical applications [33]. Moreover, it cannot be used to define a positive definite kernel since it violates the triangle inequality [3].

Therefore, researchers are coming up with distances for time-series that approximate the DTW at lower computational costs either by adding global path constraints [30, 36] or by reducing the number of instances e.g. in nearest neighbor classification

[35].

In this paper we introduce a new kernel, called MUlti-Scale Smoothing Kernel (MUSS). The basic idea behind our kernel is to take into account many smoothed versions of the time-series and compute the similarity of the time-series as the aggregation of the similarities of the multiple smoothed versions of the original time-series. The underlying idea is that by smoothing the original time-series we will get rid of slight perturbations, and so the basic trends will become apparent and more easily detected. The main strength of this kernel is the integration of multiple time-scales, that is, at a high level, the MUSS kernel is a combination of linear kernels obtained by using several smoothed versions over different scales from the original time-series. In a sense, the kernel-derived distance that is proposed here tries to fix the brittleness of Euclidean distance without incurring in the high computational costs of DTW. Moreover, our kernel can easily be adapted to deal with multidimensional time series by considering multi-variate versions of the point-wise distance between time-series. In addition, we can derive a distance metric from the kernel definition that satisfies the triangle inequality.

The main goal of the proposed kernel is to recognize as similar time-series that may be slightly perturbed from one another. Namely, it tries to focus on the shape of the time-series and not so much on the details. It is conceivable that small errors in measurement or delivery of data may result in slight shifts or misalignments of time-series. Consequently, any data that is sent through complicated machinery can suffer from this type of misalignment as for example astronomic data, and could benefit from our kernel.

In this work, two ways of computing the kernel are presented: a recursive formulation and an equivalent matrix-based formulation. To evaluate the proposed kernel three experiments have been carried out. As an initial step, several synthetic datasets have been generated from UCR time-series repository [20] and the KDD challenge of 2007 [19], with the purpose of validating our kernel-derived distance over shifted time-series. In particular, a comparison with DTW and Euclidean distances shows that our kernel-derived distance outperforms the Euclidean distance and is competitive with respect to the DTW distance while having a much lower computational cost. The DTW distance is designed to deal with misalignments and shifts optimally. Therefore, our objective is not to beat the

DTW, but to approach its performance without incurring its high computational cost. On the other hand, the Euclidean distance has been considered as baseline distance. In the second experiment, the proposed kernel has been applied to the original UCR time-series [20] to analyze its potential in time-series classification using an SVM. In this case, the proposed kernel shows a remarkable performance when comparing with a kernel based on DTW [10] and a linear kernel. Finally, two real-world applications related to ozone concentration in atmosphere and electricity demand have been considered to show the performance of the MUSS kernel over very specific datasets. In this case, an accuracy ranging from 97% to 99% has been obtained.

The paper is structured as follows. Section 2 presents the most relevant related works found in literature. Section 3 describes our time-series kernel and its corresponding derived distance. The experimental results are presented in Sections 4, 5 and 6. Finally, Section 7 concludes with a summary of our main contributions and possible directions for future work.

2. Related work

Similarity and distance measures for time-series are a crucial ingredient in solving time-series classification and forecasting problems [29, 15]. For this reason, many distances have been proposed. For example, [1] defines a distance between two time-series representing the convexities/concavities of two shape contours. In [4] the authors modify the Euclidean distance with a correction factor based on the complexity of the input time-series.

The success and popularity of Support Vector Machines has motivated researchers to design kernels that capture similarity between time-series and sequences. For example, [32] define a kernel for the particular task of handwritten character recognition. In this work, the authors approximate each time-series by a linear combination of piecewise polynomial functions and the kernel is based on the product of the coefficients and functions that form part of this approximation. Another family of kernels for time-series based on Echo State Network [23, 27] with a deterministic reservoir architecture is proposed in [6]. Their kernel is defined by the gaussian kernel with the L_2 distance between the corresponding readouts for each time-series from the same reservoir.

It is well-known that the DTW distance is not a distance in a strict sense as it does not satisfy the triangle inequality and, therefore, it can not be used to define a positive definite kernel [3]. Despite this disadvantage, many variants of DTW and definitions of kernels based on DTW have been recently proposed in the literature. As an example, [17] use a gaussian kernel and the DTW distance with a special support vector machine, which has the ability to handle non positive-definite kernel matrices.

Another example of the use of (a weighted variant of) DTW for time-series classification, this time based on nearest neighbors is [18]. The weights penalize instances with higher phase difference between a reference point and a testing point with the purposes of minimizing the distortion caused by outliers.

More recently, non linear kernels have been proposed for time-series classification. In [10] a new kernel based on the DTW distance is defined by global alignments (GA-DTW). In particular, the kernel is defined as the sum of the exponential function of the distances for all possible alignments. However, this kernel has a high computational cost and similar constraints on alignments to that of [30] are presented to speed-up the computation in [8]. The same author presents another kernel based on the idea that similar time-series should be fit well by the same models [9]. The author used autoregressive models and thus the name of autoregressive kernel. In particular, these two global alignment-based and autoregressive kernels defined in [8] and [9] have been recently used in machine olfaction applications in conjunction with SVM [33]. An extension of SVM based on nonlinear dynamical systems theory is presented in [16]; here it is shown that these non-linear methods perform better and faster than the DTW distance-based methods. However, linear kernels may still be preferred over their more accurate non-linear counterparts due to their interpretability, computational efficiency and the lack of metaparameters that need tuning.

A kernel for periodic time-series arising in the field of astronomy is presented in [34]. This kernel is similar to a global alignment kernel as it consists of the sum of the exponential function of the inner products for all possible shifts of a time series instead of alignments.

Finally, another kernel for time-series is proposed in [22]. In particular, the time series are represented with a summarizing smooth curve in a Hilbert space and the learning method of the kernel is based on

Gaussian processes.

In the last years, several approaches have been proposed to combine multiple kernels instead of using a single kernel. In [28] a combination of kernels for long-term time-series forecasting is presented. In particular, a kernel that takes into account the seasonality of the time-series to improve the performance of the predictor is combined with the well-known gaussian or rational quadratic kernel. A detailed description can be found in [14].

Due to the fact that annotation of class labels in time-series is very expensive, researchers are exploring the semi-supervised methodology to the problem of time-series classification. The main strategies within this line of work are the extension of well-known semi-supervised techniques for static data classification to time-series problems [21], and the definition of new distances for time-series that work well in semi-supervised classification [7].

3. Kernel Description

This Section presents the notation used in this paper and also provides the definitions underlying the proposed kernel.

3.1. Preliminaries

Definition 1. Time-series. A time-series X is a set of temporally sorted sequence of real values. In this work, $X = \{x_1, \dots, x_N\}$, where N is the length of the time-series.

Definition 2. Subsequence time-series. A subsequence of length k of a time-series $X = \{x_1, \dots, x_N\}$ is a time-series $X_j^k = \{x_j, x_{j+1}, \dots, x_{j+k-1}\}$ for $1 \leq j \leq N - k + 1$.

Definition 3. $\{k, j\}$ -Order partial sum. A $\{k, j\}$ -order partial sum of a time-series X , $s_{k,j}^X$, is the sum of the values of the X_j^k subsequence time-series of length k . That is:

$$s_{k,j}^X = \sum_{i=0}^{k-1} x_{j+i} = x_j + x_{j+1} + \dots + x_{j+k-1}.$$

Definition 4. k -Order partial sum time-series. A k -order partial sum time-series is a time-series S_k^X whose values are $s_{k,j}^X$ for $1 \leq j \leq N - k + 1$, that is, the sum of all the values of the subsequences of length k of the time-series X .

$$S_k^X = \{s_{k,1}^X, s_{k,2}^X, \dots, s_{k,N-k+1}^X\}.$$

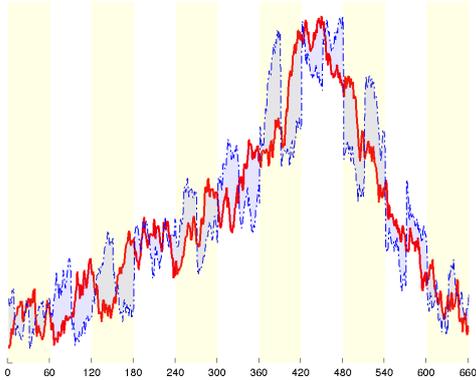


Figure 1: Example of two similar time-series. These time-series represent measurements taken each second over an 11 minute period. One of the time-series (thin, dotted blue) has been obtained from the other (thick, red continuous) as follows: within each minute block, the values of the original (red) time-series have been circularly shifted with an offset of 30 measurements. The Euclidean distance, as illustrated by the shaded regions between these two time-series, is high. On the other hand our kernel should recognize both time-series as very similar. Plot best seen in color.

For example, the $\{k, j\}$ -order partial sums and the k -order partial sum time-series for the $X = \{3, 2, 4, 1\}$ time-series are:

$$\begin{aligned}
 s_{2,1}^X &= 3 + 2 = 5 & S_2^X &= \{5, 6, 5\} \\
 s_{2,2}^X &= 2 + 4 = 6 \\
 s_{2,3}^X &= 4 + 1 = 5 \\
 s_{3,1}^X &= 3 + 2 + 4 = 9 & S_3^X &= \{9, 7\} \\
 s_{3,2}^X &= 2 + 4 + 1 = 7 \\
 s_{4,1}^X &= 3 + 2 + 4 + 1 = 10 & S_4^X &= \{10\}
 \end{aligned}$$

3.2. Motivation

The main motivation in the definition of the MUSS kernel proposed in this paper is to obtain a similarity measure for time-series that yields high values when two time-series X and Y have the same shape but may be slightly perturbed from each other. As an illustration, Figure 1 shows two time-series that are in fact similar in the sense that they follow the same trend, however, the Euclidean distance between the two time-series is high and so it fails at detecting the similarity. As a consequence, using the Euclidean distance in distance-based classification algorithms may lead to poor results. The purpose of this work is to propose a kernel that

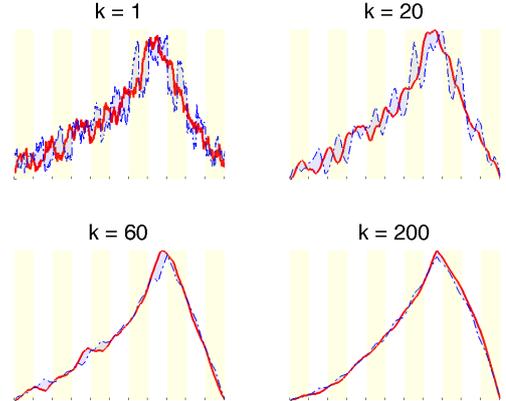


Figure 2: Example of k -order partial sum time-series for $k = 1, 20, 60, 200$ on the time-series of Figure 1. The reader should notice that as k increases, the smoothed versions of the original time-series become more and more similar. Our kernel is computed as the sum of the similarities over all possible k . We chose explicitly to give all k equal weight since a priori there is no reason why one should pay more attention to one particular scale. Plot best seen in color.

yields high similarity for time-series that have similar shapes with a reasonable time complexity.

Figure 2 shows the different time-series composed by the values of the partial sums of the X and Y time-series for orders $k \in \{1, 20, 60, 200\}$. Note that when the order is $k = 1$ the time-series are the original X and Y time-series. It can be readily observed that the greater the order of partial sums is, the smoother the time-series become. In fact, the partial sums are essentially unnormalized window-average smoothing over the original series.

The MUSS kernel proposed here is obtained by adding the inner products of partial sum time-series over all possible orders. It is not necessary to discover the best alignment between two time-series, in contrast to the DTW distance, as all partial sums will be included in the kernel definition. Moreover, as it is shown in the Section 4 the MUSS kernel can be calculated very fast when compared to the DTW distance since finding an optimal alignment is not necessary.

3.3. Definition of the kernel

Let X and Y be two time-series of length N . Let U^X and U^Y be two upper triangular matrices defined as:

$$U^X = [U_1^X, \dots, U_N^X] \quad (1)$$

$$U^Y = [U_1^Y, \dots, U_N^Y] \quad (2)$$

where U_i^X and U_i^Y are the i -th rows of the matrices U^X and U^Y , respectively, which are defined by the elements u_{ij}^X and u_{ij}^Y as follows:

$$u_{ij}^X = \begin{cases} s_{i,j}^X & \text{if } 1 \leq j \leq N - i + 1 \\ 0 & \text{if } j > N - i + 1 \end{cases} \quad (3)$$

$$u_{ij}^Y = \begin{cases} s_{i,j}^Y & \text{if } 1 \leq j \leq N - i + 1 \\ 0 & \text{if } j > N - i + 1 \end{cases} \quad (4)$$

Finally, the MUSS kernel is defined as the sum of the scalar products among the rows of the U^X and U^Y matrices. That is,

$$\text{Kernel}(X, Y) = \sum_{i=1}^N \langle U_i^X, U_i^Y \rangle \quad (5)$$

where U_i^X and U_i^Y are defined by Equations (1)–(4) and $\langle \cdot, \cdot \rangle$ is the scalar product of two vectors in \mathbb{R}^N . It is obvious that the function defined by Equation (5) is indeed a kernel as it can be represented by an inner product in the high-dimensional feature space $\phi(\cdot)$ defined as follows:

$$\text{Kernel}(X, Y) = \langle \phi(X), \phi(Y) \rangle \quad (6)$$

where

$$\begin{aligned} \phi: \mathbb{R}^N &\longrightarrow \mathbb{R}^{N^2} \\ X &\longrightarrow \phi(X) = (U_1^X, \dots, U_N^X) \end{aligned}$$

Next, we show an illustrative example for the time-series $X = \{3, 2, 4, 1\}$ and $Y = \{1, -1, 0, 2\}$. Firstly, the U^X and U^Y matrices comprising the partial sums of the X and Y time-series have to be computed. It is necessary only to estimate the partial sums for all the orders greater or equal than 2 as the 1 order partial sums are the own values of the time-series. The 2-order partial sums for X and Y are $S_2^X = \{5, 6, 5\}$ and $S_2^Y = \{0, -1, 2\}$, respectively. Analogously, the 3 and 4 order partial sums are $S_3^X = \{9, 7\}$, $S_3^Y = \{0, 1\}$, $S_4^X = \{10\}$ and $S_4^Y = \{2\}$. Therefore, the matrices are:

$$U^X = \begin{bmatrix} 3 & 2 & 4 & 1 \\ 5 & 6 & 5 & 0 \\ 9 & 7 & 0 & 0 \\ 10 & 0 & 0 & 0 \end{bmatrix} \quad U^Y = \begin{bmatrix} 1 & -1 & 0 & 2 \\ 0 & -1 & 2 & 0 \\ 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{bmatrix}$$

The second step consists in calculating the scalar products of the rows of the U^X matrix and the corresponding rows of the U^Y matrix. That is,

$$\begin{aligned} \langle U_1^X, U_1^Y \rangle &= 3 \cdot 1 + 2 \cdot (-1) + 4 \cdot 0 + 1 \cdot 2 = 3 \\ \langle U_2^X, U_2^Y \rangle &= 5 \cdot 0 + 6 \cdot (-1) + 5 \cdot 2 = 4 \\ \langle U_3^X, U_3^Y \rangle &= 9 \cdot 0 + 7 \cdot 1 = 7 \\ \langle U_4^X, U_4^Y \rangle &= 10 \cdot 2 = 20 \end{aligned}$$

where U_i^X and U_i^Y are the i rows of the U^X and U^Y matrices, respectively.

Finally, the MUSS kernel is defined as the sum of the above-mentioned scalar products. Therefore,

$$\text{Kernel}(X, Y) = (3 + 4 + 7 + 20) = 34$$

3.4. Speeding-up the computation

The MUSS kernel can be computed following from the definition given above. However, an exhaustive analysis provides two different schemes to compute the kernel at a far lower computational cost. In this Subsection these two schemes are described in detail.

3.4.1. Recursive Scheme

This scheme is based on the fact that the i -th row of the U^X matrix can be recursively obtained from the $(i - 1)$ -th row and the original values of the time-series X . Continuing the previous example, the following relationships hold:

$$\begin{aligned} U_{2,1}^X &= 5 &= U_{1,1}^X + x_2 &= 3 + 2 \\ U_{2,2}^X &= 6 &= U_{1,2}^X + x_3 &= 2 + 4 \\ U_{2,3}^X &= 5 &= U_{1,3}^X + x_4 &= 4 + 1 \\ U_{3,1}^X &= 9 &= U_{2,1}^X + x_3 &= 5 + 4 \\ U_{3,2}^X &= 7 &= U_{2,2}^X + x_4 &= 6 + 1 \\ U_{4,1}^X &= 10 &= U_{3,1}^X + x_4 &= 9 + 1 \end{aligned}$$

In general, the (i, j) -element of the U^X matrix can be obtained from the recursive formula as follows:

$$U_{i,j}^X = U_{i-1,j}^X + x_{j+i-1}$$

3.4.2. Matrix-based scheme

This scheme is based on the observation that each row U_i^X of the matrix U^X can be obtained by multiplying a matrix M_i by the row vector composed of the values of the time-series. That is,

$$U_i^X = (x_1, \dots, x_N) \cdot M_i \quad (7)$$

where $M_i = [A_i|0]$ is the $N \times N$ matrix capturing the partial sums of order i and the $A_i = \{a_{lj}^i\}_{1 \leq l, j \leq N}$ matrix is defined by: $a_{lj}^i = 1$ for all l, j such that $1 \leq l \leq i$ and $1 \leq j \leq N - i + 1$, and 0 otherwise.

The matrix M_i has two well-differentiated blocks, a first block, A_i , comprising a diagonal band matrix of width i composed of 1's and two null triangular matrices of dimensions $(N-i) \times (N-i)$ and a second block, composed of a null matrix of dimensions $N \times (i-1)$. It can be observed that the structure of the M_i matrix is mainly due to the definition of the i -order partial sums as the diagonal band of width i represents the i -order partial sums for the different subsequence time-series of length i . On the other hand, the null matrix of dimension $N \times (i-1)$ is because of the columns of the U^X matrix are equal to 0 when $j > N - i + 1$.

We define:

$$\begin{aligned} \text{Kernel}(X, Y) &= \sum_{i=1}^N \langle U_i^X, U_i^Y \rangle \\ &= \sum_{i=1}^N U_i^X \cdot (U_i^Y)^t \\ &= \sum_{i=1}^N (\bar{x} \cdot M_i) \cdot (\bar{y} \cdot M_i)^t \\ &= \sum_{i=1}^N \bar{x} \cdot (M_i \cdot M_i^t) \cdot \bar{y}^t \\ &= \bar{x} \cdot \left(\sum_{i=1}^N M_i \cdot M_i^t \right) \cdot \bar{y}^t \end{aligned}$$

where \bar{x} and \bar{y}^t are the row and column vectors composed by the values of the time-series X and Y , respectively, and $(\cdot)^t$ denotes the transpose. Let Z the matrix defined by

$$Z = \sum_{i=1}^N M_i \cdot M_i^t \quad (8)$$

and thus $\text{Kernel}(X, Y) = \bar{x} \cdot Z \cdot \bar{y}^t$.

It can be easily seen that the Z matrix is symmetric since it is the sum of symmetric matrices. Moreover, it is positive semidefinite because it defines an inner product (cf. Eq. 6). On the other hand, this matrix only depends on the length of

the time-series N . Thus, the matrix can be pre-computed just once when computing the pair-wise kernel of a whole dataset providing a reduction of the computing time. Most of kernels for time series proposed in the literature can not be represented by a matrix, which is not depending on the values of the time series [6, 10, 34]. Moreover, the structure of the $Z = \{z_{ij}\}_{1 \leq i, j \leq N}$ matrix can be obtained by unwrapping Eq. 8:

$$z_{ij} = \begin{cases} i \cdot (N - (j - 1)) & \text{if } j \geq i \\ j \cdot (N - (i - 1)) & \text{if } i > j \end{cases}$$

Continuing with the previous example, it can be observed that, for all $i = 1, \dots, 4$:

$$U_i^X = (3, 2, 4, 1) \cdot M_i$$

where M_1 is the identity matrix of dimension 4×4 and the remaining matrices and the resulting Z are:

$$\begin{aligned} M_2 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} & M_3 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \\ M_4 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} & Z &= \begin{bmatrix} 4 & 3 & 2 & 1 \\ 3 & 6 & 4 & 2 \\ 2 & 4 & 6 & 3 \\ 1 & 2 & 3 & 4 \end{bmatrix} \end{aligned}$$

It should be noted that a distance metric can be obtained from any positive definite kernel Ker using the standard transformation described in the following equation [31]:

$$d(u, v) = Ker(u, u) + Ker(v, v) - 2 \cdot Ker(u, v).$$

Therefore, when this work refers to the MUSS kernel as a distance it really means the derived distance from the kernel.

4. Results

This section presents the results obtained by the application of the MUSS kernel to forty datasets for measuring the similarity in shifted time-series. Section 4.1 provides a detailed description of all datasets used in the experiments. In Section 4.2 the kernel has been applied to forty time-series to validate its potential for separating classes in shifted time series. Finally, a statistical analysis of the results is reported in Section 4.3.

4.1. Description of datasets

The MUSS kernel has been initially tested on several synthetic datasets from the UCR time-series repository [20] and from data from the 2007 KDD Challenge [19]. In particular, a synthetic dataset derived from each available dataset in the aforementioned repositories has been created in the following way: for each different class existing in a dataset, a time-series belonging to this class is chosen uniformly at random. Then, shifted time-series copies with varying offsets are created from the selected chosen time-series having the same label as the original time-series the copies were created from.

Time-series lengths in our datasets range from 60 to 637, with the average and the median being 282.1 and 272.5, respectively. Computation times are highly sensitive to the time-series length, especially for the DTW algorithm, which is quadratic in this parameter. Relevant information about these datasets is summarized in Tables 1 and 2. The class labels for the test KDD datasets are not available.

Table 1: Data sets from UCR time-series repository.

Data sets from UCR	Num. Instances	Num. Classes	Length
50 Words	450	50	270
Adiac	296	37	176
Beef	45	5	470
CBF	21	3	128
Coffee	18	2	286
ECG	14	2	96
Fish	63	7	463
Face (All)	112	14	131
Face (Four)	36	4	350
Gun-Point	16	2	150
Lighting-2	20	2	637
Lighting-7	63	7	319
OSU Leaf	54	6	427
Olive Oil	40	4	570
Swedish Leaf	105	15	128
Trace	36	4	275
Two Patterns	28	4	128
Synth. Control	36	6	60
Wafer	16	2	152
Yoga	18	2	426

4.2. Validation

A statistic based on pair-wise distances has been developed to show how well the MUSS kernel is able to separate classes in time-series. Let D be a labeled dataset of M time-series of the same length N . Let $c(X)$ be the class of the time-series $X \in D$. Then, the SM separation measure is defined as follows:

$$SM = (INTRA - INTER)/MAX$$

Table 2: Datasets derived from the 2007 KDD Challenge.

Data sets from KDD 2007	Training Size	Test Size	Num. Classes	Length
kdd01	80	20	3	1024
kdd02	10	20	2	24
kdd03	18	72	4	512
kdd04	18	72	3	512
kdd05	44	33	1	1639
kdd06	55	33	10	1092
kdd07	54	45	4	398
kdd08	70	56	5	99
kdd09	14	35	4	70
kdd10	14	18	2	65
kdd11	14	21	2	82
kdd12	30	60	2	1024
kdd13	36	9	2	345
kdd14	14	28	4	84
kdd15	24	184	4	166
kdd16	16	24	1	136
kdd17	18	27	2	405
kdd18	77	33	6	1882
kdd19	112	56	7	131
kdd20	225	27	5	270

where $INTRA$ and $INTER$ are the average pair-wise distance of time series belonging to the same and to different classes, respectively, and MAX is the maximum pair-wise distance over the whole dataset. Namely, let $A = \{(X, Y) | X, Y \in D, c(X) = c(Y)\}$ and $B = \{(X, Y) | X, Y \in D, c(X) \neq c(Y)\}$. That is, A is the set of pairs of time series that belong to the same class, and B is the set of pairs of time-series that belong to different classes. Then,

$$INTRA = \frac{1}{|A|} \sum_{(X,Y) \in A} d(X, Y)$$

$$INTER = \frac{1}{|B|} \sum_{(X,Y) \in B} d(X, Y)$$

$$MAX = \max_{X, Y \in D} d(X, Y)$$

where d is any distance defined in $\mathbb{R}^N \times \mathbb{R}^N$ and N is the length of the time-series in D . Observe that the higher value for this measure, the better the separation among classes obtained by distance d is. A positive value of the SM measure means that, on average, time-series belonging to the same class are closer than time-series of different classes. A negative value of the SM measure means the opposite, that is, pairs of series of different classes tend to be closer than pairs of the same class. Notice that for negative or values of the SM measure close to 0, any distance-based classifier such as Nearest Neighbors is doomed to fail.

Table 3: Separation measure among classes and computing times when using several distances.

Data sets	Separation Measure			CPU Times (in seconds)		
	Euclidean distance	MUSS distance	DTW distance	Euclidean distance	MUSS distance	DTW distance
kdd01	0.004	0.010	0.129	829.8	2906.6	54550.3
kdd02	0.076	0.128	0.174	0.2	0.3	1.8
kdd03	0.499	0.637	0.718	0.7	40.2	2733.9
kdd04	0.588	0.670	0.388	0.7	41.6	2696.3
kdd05	0.193	0.389	0.869	1.5	1838.0	54971.2
kdd06	0.013	0.020	-0.014	2.1	1187.6	35261.7
kdd07	0.159	0.229	0.056	2.3	68.3	5147.5
kdd08	0.131	0.208	0.396	3.6	10.9	534.6
kdd09	-0.037	-0.041	0.096	0.3	0.7	20.4
kdd10	0.012	0.109	0.335	0.2	0.5	23.6
kdd11	0.017	0.000	0.031	0.3	0.6	25.1
kdd12	0.364	0.387	0.439	1.2	587.2	18344.6
kdd13	-0.028	0.009	-0.018	0.7	24.4	1187.5
kdd14	0.083	0.095	0.208	0.3	0.7	34.0
kdd15	-0.048	-0.007	0.134	2.1	10.2	827.4
kdd16	0.019	0.044	0.040	0.3	1.1	80.0
kdd17	-0.036	0.004	0.158	0.4	10.5	799.0
kdd18	0.157	0.058	0.015	3.7	5993.7	184957.3
kdd19	-0.009	-0.024	0.050	7.9	33.1	2016.0
kdd20	0.089	0.147	0.080	24.2	285.5	25279.4
50 Words	0.155	0.196	0.498	177.4	3653.3	176629.5
Adiac	-0.042	-0.040	-0.027	83.8	387.5	32702.9
Beef	0.696	0.894	0.557	1.8	93.7	5505.9
CBF	0.150	0.311	0.557	0.5	4.9	94.2
Coffee	-0.015	0.111	-0.015	0.3	5.8	316.9
ECG	0.045	0.054	0.126	0.2	3.8	22.8
FISH	-0.004	-0.008	-0.018	10.3	46.4	2690.4
Face (All)	0.007	0.110	0.387	1.1	46.7	1928.6
Face (Four)	-0.005	0.016	0.011	3.4	131.1	10576.4
Gun-Point	0.136	0.113	0.345	0.3	1.2	83.3
Lighting-2	0.126	0.152	0.263	0.4	29.0	1964.2
Lighting-7	0.137	0.268	0.279	3.6	67.1	4767.3
OSU Leaf	0.229	0.346	0.098	1.4	94.0	6396.3
Olive Oil	-0.063	-0.041	-0.027	2.5	100.8	6459.6
Swedish Leaf	0.104	0.144	0.048	9.3	39.2	2286.0
Trace	0.300	0.303	0.091	1.2	3.1	65.1
Two Patterns	0.113	0.165	0.585	1.1	19.2	1195.1
Synth. Control	0.103	0.293	0.580	0.7	3.9	174.5
Wafer	0.126	0.172	0.015	0.3	1.0	72.9
Yoga	-0.000	0.073	-0.005	0.3	11.0	724.1
Average	0.114	0.168	0.216	29.6	444.6	16103.7

Table 3 presents a comparison of the separation statistic and computation time of the following distances: the Euclidean distance, the one derived from the MUSS kernel (which we call MUSS distance), and the DTW distance. The comparison is over the synthetic 40 datasets from UCR and KDD repositories. The computation times have been obtained with a laptop of 8 GB of RAM and an Intel Core i7 processor running at 2.7 GHz. The distance that better separates the existing classes for each dataset is marked in bold style. It can be seen that the average of the separation measure for the MUSS kernel is better than that of the Euclidean distance and similar to that of the DTW distance. However, further statistical analysis shows that the differences between the DTW distance and the MUSS are not significant, while the differences between the Euclidean distance and the MUSS distance are. When looking at the columns for computation times, it is very clear that the Euclidean distance is by far the fastest one to compute, followed by our MUSS distance using (roughly) an order of magnitude extra CPU time. The DTW distance is by far the slowest, needing two more orders of magnitude than our MUSS distance.

We further analyze some of the rows of Table 3, in particular, the rows corresponding to the *FISH* and *KDD-03* datasets. These two rows are representatives of opposite behavior of the SD measure: for *FISH* all distances have negative values in the SM measure, while for *KDD-03* all of them perform quite well. We try to explain why this should be the case in what follows. Remember that in order to generate the synthetic datasets, we chose one representative of each class at random and then created circularly shifted copies from each representative as time-series of the same class. Depending on the chosen representatives, some of the resulting synthetic datasets are thus inconsistent in their labels and therefore all distance measures perform poorly. Figure 3 shows one example of this (poor behavior in *FISH*), and Figure 4 shows a case of good behavior across all distances (*KDD-03*).

Table 4 further summarizes Table 3, where the reader can observe that the behavior of the MUSS distance is similar than that of DTW on average (1.75 versus 1.70), and both outperform the Euclidean distance (1.75 and 1.70 versus 2.55). From this table, it can be noticed that the MUSS kernel reaches the highest rank in 14 datasets, the second position and third positions in 21 and 4 datasets, respectively, the DTW distance reaches the high-

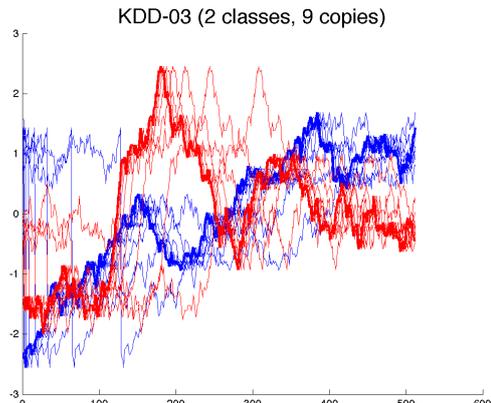


Figure 4: The plot shows in red the copies made out of the first representative for dataset *KDD-03*, and in blue the copies made out of the second representative. In this case, both representatives are in fact quite different and thus the copies are more similar among themselves than with respect to copies of other classes. Therefore, all SM measures are positive. Plot best seen in color.

est rank in 24 datasets, the second position for 4 datasets (*kdd13*, *kdd16*, *Coffee*, *Face (Four)*) and the third position for the 12 remaining datasets, and finally, the euclidean distance obtains the highest ranking just for 2 datasets (*kdd13* and *FISH*), the second position for 14 datasets and the third position for the remaining 24 datasets. The average ranking for each distance is summarized in Table 4. It can be observed that DTW and the MUSS kernel present similar behavior on average.

Table 4: Number of times each distance achieves the first, second and third positions over all datasets and average rank.

Distances	#1st	#2nd	#3rd	Avg. Rank
Euclidean	2	14	24	2.55
MUSS	14	21	5	1.75
DTW	24	4	12	1.70

Table 5 shows the wins matrix for pairs of distances over the 40 datasets. That is, in how many datasets a distance separates better than another distance. It should be read as follows: if row i and column j contains number m , then distance i has beaten distance j a total of m times. It can be noticed that the MUSS distance beats the Euclidean distance in 34 datasets, and beats the DTW distance in 16 datasets. The DTW distance wins in 28 datasets to the Euclidean distance and in 24

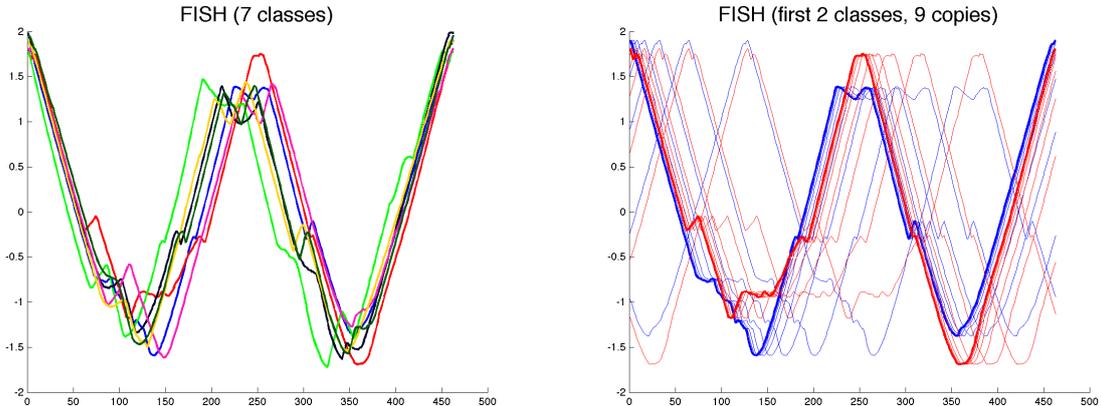


Figure 3: The left plot shows the 7 *representatives* chosen for each of the 7 classes found in the *FISH* dataset. The right plot shows in red the 9 copies made out of the first representative, and in blue the 9 copies made out of the second representative. Since the representatives are, in fact, quite alike, it turns out that the copies are further apart from their representatives than from other copies of other representatives, and therefore any reasonable distance measure is going to have low scores in the SM measure. Plots best seen in color.

datasets to the MUSS kernel, and finally, the Euclidean distance wins in 6 and 12 datasets to the MUSS and DTW distances, respectively.

Table 5: Win matrix for pairs of distances.

Distances	Euclidean	MUSS	DTW
Euclidean	–	6	12
MUSS	34	–	16
DTW	28	24	–

4.3. Statistical analysis

A statistical analysis has been conducted to evaluate the significance of the MUSS kernel, following the non-parametric procedures discussed in [13]. Non-parametric tests have to be selected because the conditions of normality and homoscedasticity are not met. All tests applied in this work have been obtained by using the open-source platform KEEL [2].

Friedman and Iman-Davenport (ID) tests have been applied to assess whether there are global differences in the separation measure obtained by the three distances compared. The p-values obtained by both tests at the level of significance $\alpha = 0.05$ are: $1.12 \cdot 10^{-4}$ for the Friedman test, and $4.25 \cdot 10^{-5}$ for the ID test. As the p-values obtained from both of the tests are lower than the level of significance considered, it can be stated that there exist significant differences among the results obtained by

three distances and a post-hoc statistical analysis is required.

The Holm and Shaffer tests have been applied to perform pair-wise comparisons. Table 6 shows the sorted p-values obtained by all comparisons for two levels of significance ($\alpha = 0.05$ and $\alpha = 0.10$). Both of the tests allow concluding that the MUSS distance is better than the Euclidean distance and similar to DTW distance for both levels of significance, as the two tests reject the hypotheses for the MUSS kernel and DTW distance versus Euclidean distance but do not reject the null hypotheses for the MUSS kernel versus DTW distance.

5. Time-series classification

In our second experiment, we use Support Vector Machines to perform classification using three different kernels: our MUSS kernel, the GA-DTW kernel based on global alignments in [8], and a linear kernel. We compare the prediction accuracy achieved by these three kernels over the 20 datasets of the UCR repository [20]. The results in this section show that our kernel is able to achieve good results for general time-series (not necessarily shifted). We have used the implementation LIB-SVM [5] of SVM due to its wide acceptance in the literature. Code for the GA-DTW kernel has been taken from ¹.

¹<http://www.iip.ist.i.kyoto-u.ac.jp/member/cuturi/Code/DTWkernel.cpp>

Table 6: Holm and Shaffer tests results

i	Distances	z	p	$\frac{\alpha}{i}$ ($\alpha = 0.05$)	$\frac{\alpha}{i}$ ($\alpha = 0.10$)
3	Euclidean vs. DTW	3.35	$7.96 \cdot 10^{-4}$	0.016	0.033
2	Euclidean vs. MUSS	3.02	$2.54 \cdot 10^{-3}$	0.025	0.05
1	MUSS vs. DTW	0.33	0.74	0.05	0.10

Table 7 shows the accuracy of the three methods. The last row presents the average and the standard deviation in brackets for all data sets. It can be appreciated that the MUSS kernel is better than the linear kernel and similar to the GA-DTW kernel having in addition the lowest standard deviation.

On the other hand, Table 8 reports the time (in seconds) needed for the kernel computations. The MUSS kernel proposed here is much faster than the GA-DTW kernel and as fast as the linear kernel. In fact, the GA-DTW kernel is not applicable to high-dimensional real-world problems that require the computation of the kernel for many instances.

Table 7: Accuracy (%) obtained by the SVM classifier when using different kernels.

Data sets	GA-DTW kernel	Linear kernel	MUSS kernel
50 Words	81.54	66.37	65.49
Adiac	50.38	51.66	62.92
Beef	46.67	46.67	70.00
CBF	100.00	88.00	96.00
Coffee	78.57	96.43	92.86
ECG	81.00	82.00	80.00
FISH	95.43	84.57	80.57
Face (All)	82.49	74.73	75.68
Face (Four)	96.59	86.36	88.64
Gun-Point	96.00	88.67	92.67
Lighting-2	65.57	67.21	70.49
Lighting-7	75.34	65.75	65.75
OSU Leaf	52.89	46.69	45.87
Olive Oil	40.00	40.00	63.33
Swedish Leaf	92.80	85.60	80.00
Trace	99.00	75.00	91.00
Two Patterns	97.60	81.08	83.85
Synth. Control	97.33	94.00	96.33
Wafer	80.37	94.79	88.63
Yoga	49.23	65.40	60.20
Average	77.94 (20.17)	74.05 (17.25)	77.51(14.07)

These results show that our MUSS kernel achieves comparable accuracy to the DTW-based kernel but at a much lower computational cost.

6. Real-world applications: ozone concentration in atmosphere and electricity demand classification

Finally, we present results in two real-world applications: atmospheric pollutants (ozone), and electricity demand. The pattern recognition in

Table 8: Computing times in seconds required to classify the datasets when using different kernels.

Data sets	GA-DTW kernel	Linear kernel	MUSS kernel
50 Words	3820.55	1.54	2.86
Adiac	1183.33	0.85	1.45
Beef	52.13	0.03	0.17
CBF	77.81	0.14	0.18
Coffee	16.49	0.02	0.07
ECG	23.57	0.05	0.08
FISH	1654.97	0.38	1.34
Face (All)	3297.67	2.98	4.61
Face (Four)	50.72	0.04	0.10
Gun-Point	33.37	0.05	0.08
Lighting-2	382.17	0.09	0.61
Lighting-7	132.32	0.06	0.23
OSU Leaf	2127.90	0.50	1.50
Olive Oil	75.73	0.03	0.24
Swedish Leaf	1210.00	1.31	1.96
Trace	194.04	0.10	0.29
Two-Patterns	12666.86	11.64	17.76
Synth. Control	84.69	0.27	0.40
Wafer	26289.65	20.26	30.52
Yoga	28925.00	6.20	11.18
Average	4114.95 (426.96)	2.33 (0.25)	3.78 (0.38)

ozone time-series data is an important task as it allows governments to activate alert protocols and implement good environmental policies if high ozone concentration levels are predicted. On the other hand, electricity-producing companies are interested in predicting spikes of demand in order to schedule the energy production to maximize their profit.

Ozone time-series have been retrieved from a meteorological station placed in the outskirts of Seville city (Spain), providing 312 times series composed of 168 hourly records each one. The dataset is classified into two class corresponding to high and low ozone level periods (165 and 147 time-series, respectively). The time-series have been split into a training set of 218 time-series, and a test set of 94 time-series, preserving the proportion between two existing classes.

Electricity demand time-series from 2008 to 2012 for the Spanish electricity market have been collected [24], providing 261 times series composed of 168 hourly records each. Four datasets, called Demand-2, Demand-3, Demand-4 and Demand-5, have been used depending on the necessary number

of classes that an electrical engineering can consider useful to determine the behavior of the demand. The time-series data have been split in training set (182, 182, 181 and 181 time-series for each one of 4 datasets, respectively) and test set (79, 19, 80 and 80, respectively) preserving the proportion among existing classes.

The MUSS kernel has been used to classify the ozone time-series into weeks of high or low ozone concentration and likewise the demand into weeks of different behavior (valley, spikes, etc). More concretely, we have used SVMs in conjunction with the same three kernels as in the previous section.

Table 10 and Table 9 show the percentage of time-series correctly classified in the test sets and the time in seconds obtained by the application of the SVM classifier for all real-world data sets when using several kernels. The last row presents the average and the standard deviation in brackets for all data sets. It can be observed that the MUSS kernel presents the better results in accuracy for all data sets. Moreover, the MUSS kernel is 440 times faster than the GA-DTW kernel approximately and it provides a mean error of classification by 3% and 5% lower than the remaining kernels.

Table 9: Percentage of time-series correctly classified.

Data sets	GA-DTW kernel	Linear kernel	MUSS kernel
Demand-2	94.94	96.20	98.73
Demand-3	89.87	94.94	98.73
Demand-4	93.75	95.00	97.50
Demand-5	92.59	93.83	97.53
Ozono	93.68	97.89	98.95
Average	92.97 (1.72)	95.57 (1.38)	98.29 (0.64)

Table 10: Time in seconds required to classify the test set.

Data sets	GA-DTW kernel	Linear kernel	MUSS kernel
Demand-2	89.83	0.17	0.22
Demand-3	90.21	0.16	0.20
Demand-4	90.04	0.17	0.22
Demand-5	91.15	0.16	0.25
Ozono	143.93	0.21	0.27
Average	101.03 (21.45)	0.18 (0.02)	0.23 (0.02)

7. Conclusions

In this paper we have presented the MUSS kernel for time-series data and its associated distance metric. Initial experiments show promise in detecting similarity between time-series. The MUSS kernel

has been compared to the Euclidean distance as a reference distance and the DTW distance as one of the most competitive distances that exist in the literature. Also, the MUSS kernel has been used in conjunction with the SVM classifier to time-series classification and compared with the GA-DTW kernel [8] and the linear kernel. The kernel has shown to be efficient in separating different classes in time-series from well-known repositories, and also, it has been successfully applied to real-world time-series. In particular, a low error for the classification of the ozone in low and high concentrations in the atmosphere and classification of the electricity demand has been obtained.

For future work, we would like to generalize our MUSS kernel to multivariate time-series and time-series that differ in length. Moreover, we would like to adapt our ideas so that they can be used in a *stream* setting where time-series keep growing unboundedly.

Acknowledgments

This work is partially supported by the Spanish Ministry of Science and Technology contracts TIN2011-27479-C04-03 and TIN2011-28956-C02, by the Generalitat de Catalunya 2009-SGR-1428 (LARCA), by the Junta de Andalucía P12-TIC-1728, by the Pablo de Olavide University APPB813097 and by the EU PASCAL2 Network of Excellence (FP7-ICT-216886). Part of this work was done while A.T. and M.A. were at Columbia University in New York and University of California in San Diego, supported in part by grant BEDGR 2011 of the Generalitat de Catalunya. We are grateful for the fruitful discussions held with Charles Elkan, Eamonn Keogh and Ramón Huerta during A.T. and M.A.’s stay at UCSD. Finally, we also acknowledge constructive comments given by anonymous reviewers that undoubtedly helped improve this paper.

References

- [1] Tomasz Adamek and Noel E. O’Connor. A multiscale representation method for nonrigid shapes with a single closed contour. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(5):742–752, 2004.
- [2] J. Alcalá-Fdez, L. Sánchez, S. García, M. J. del Jesus, S. Ventura S, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fernández, and F. Herrera. KEEL: A software tool to assess evolutionary algorithms for data mining problems. *Soft Computing*, 13(3):307–318, 2009.

- [3] C. Bahlmann, B. Haasdonk, and H. Burkhardt. On-line handwriting recognition with support vector machines—a kernel approach. In *Proceedings of the 8th International Workshop on frontiers in Handwriting Recognition*, 2002.
- [4] Gustavo E. A. P. A. Batista, Xiaoyue Wang, and Eamonn J. Keogh. A complexity-invariant distance measure for time series. In *SIAM International Conference on Data Mining*, 2011.
- [5] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):article no. 27, 2011.
- [6] Huanhuan Chen, Fengzhen Teng, Peter Tino, and Xin Yao. Model-based kernel for efficient time series analysis. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 392–400, 2013.
- [7] Yanping Chen, Bing Yu, Eamonn Keogh, and Gustavo E. A. P. A. Batista. DTW-D: time series semi-supervised learning from a single example. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 383–391, 2013.
- [8] M. Cuturi. Fast global alignment kernels. In *International Conference on Machine Learning*, 2011.
- [9] M. Cuturi and A. Doucet. Autoregressive kernels for time series, 2011. arXiv:1101.0673.
- [10] M. Cuturi, J.-P. Vert, O. Birkenes, and T. Matsui. A kernel for time series based on global alignments. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 413–416, 2007.
- [11] C. Elkan. Deriving TF-IDF as a fisher kernel. In *12th International Conference String Processing and Information Retrieval (SPIRE)*, 2005.
- [12] Haijin Fan and Qing Song. A sparse kernel algorithm for online time series data prediction. *Expert System With Applications*, 40:2174–2181, 2013.
- [13] S. García, A. Fernandez, J. Luengo, and F. Herrera. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180:2044–2064, 2010.
- [14] M. Gonen and E. Alpaydin. Multiple kernels learning. *Journal of Machine Learning Research*, 2011.
- [15] C. Gruber, T. Gruber, and B. Sick. Online signature verification with new time series kernels for support vector machines. *Lecture notes in computer science*, 3832:500–508, 2005.
- [16] Ramón Huerta, Shankar Vembu, Mehmet K. Muezzinoglu, and Alexander Vergara. Dynamical svm for time series classification. *Lecture Notes in Computer Science*, 7476:216–225, 2012.
- [17] Arash Jalalian and Stephan K. Chalup. GDTW-P-SVMs: Variable-length time series analysis using support vector machines. *Neurocomputing*, 99:270–282, 2013.
- [18] Young-Seon Jeong, Myong K. Jeong, and Olufemi A. Omitaomu. Weighted dynamic time warping for time series classification. *Pattern Recognition*, 44:2231–2240, 2011.
- [19] E. Keogh, C. Shelton, and F. Moerchen. Challenge on time series classification in SIGKDD 2007, www.cs.ucr.edu/~eamonn/sigkdd2007timeseries.html, 2011.
- [20] Eamonn Keogh. UCR time series repository, <http://www.cs.ucr.edu/~eamonn/>, 2011.
- [21] Minyoung Kim. Semi-supervised learning of hidden conditional random fields for time-series classification. *Neurocomputing*, 119:339–349, 2013.
- [22] Z. Lu, T.K. Leen, Y. Huang, and D. Erdogmus. A reproducing kernel Hilbert space framework for pairwise time series distances. In *Proceedings of the 25th international conference on Machine learning*, pages 624–631, 2008.
- [23] M. Lukosevicius and H. Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.
- [24] Spanish Electricity Price Market Operator. <http://www.omel.es>.
- [25] Zoltán Prekopsák and Daniel Lemire. Time series classification by class-specific mahalanobis distance measures. *Advances in Data Analysis and Classification*, 6(3), 2012.
- [26] Cédric Richard, José Carlos M. Bermúdez, and Paul Honeine. Online prediction of time series data with kernels. *IEEE Transactions on signal processing*, 57(3):1058–1067, 2009.
- [27] A. Rodan and P. Tiño. Simple deterministically constructed cycle reservoirs with regular jumps. *Neural Computation*, 18(10):1822–1852, 2012.
- [28] Ginés Rubio, Luis Javier Herrera, Héctor Pomares, Ignacio Rojas, and Alberto Guillen. Design of specific-to-problem kernels and use of kernel weighted k-nearest neighbors for time series modeling. *Neurocomputing*, 73:1965–1975, 2010.
- [29] T. P. Runarsson S. Gudmundsson and S. Sigurdsson. Support vector machines and dynamic support vector machines and dynamic time warping for time series. In *International Joint Conference on Neural Networks*, 2008.
- [30] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 26:43–49, 1978.
- [31] B. Scholkopf. The kernel trick for distances. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, pages 301–307, 2000.
- [32] K. R. Sivaramakrishnan, K. Karthik, and C. Bhattacharyya. Kernel for large margin time-series classification. In *Proceedings of the International Joint Conference on Neural Networks*, pages 2746 – 2751, 2007.
- [33] Shankar Vembu, Alexander Vergara, Mehmet K. Muezzinoglu, and Ramón Huerta. On time series features and kernels for machine olfaction. *Sensors and Actuators B: Chemical*, 174:535–546, 2012.
- [34] G. Wachman, R. Khardon, P. Protopapas, and C. R. Alcock. Kernels for periodic time series arising in astronomy. In *European Conference on Machine Learning*, 2009.
- [35] X. Xi, E. Keogh, C. Shelton, L. Wei, and C.A. Ratanamahatana. Fast time series classification using numerosity reduction. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 1033–1040, 2006.
- [36] D. Yu, X. Yu, Q. Hu, J. Liu, and A. Wu. Dynamic time warping constraint learning for large margin nearest neighbor classification. *Information Sciences*, 181(13):535–546, 2011.