
Elastic Motif Segmentation and Alignment of Time Series for Encoding and Classification

Tao-Yi Lee¹
taoyilee@ntu.edu.tw

Yuh-Jye Lee²
yuhjye@math.nctu.edu.tw

Hsing-Kuo Pao³
pao@mail.ntust.edu.tw

You-Hua Lin³
m10415036@mail.ntust.edu.tw

Yi-Ren Yeh⁴
yryeh@nknku.edu.tw

¹NTU IoX Center, National Taiwan University, Taipei, Taiwan

²Department of Applied Mathematics, National Chiao-Tung University, Hsinchu, Taiwan

³Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan

⁴Department of Mathematics, National Kaohsiung Normal University, Kaohsiung, Taiwan

Abstract

Time series (TS) learning tasks usually succeed identification of motifs and data cleaning from dynamics like heartbeats and walking gaits. These signals can be seen as cascade of multiple motifs which have different periodicities. Instead of analyzing TS in the frequency domain, this paper proposes an efficient algorithm for elastically segmenting, shrinking and stretching motifs, which may have different lengths but share the same shape, from TS onto a synthetic time-axis, thereby generating a set of synchronized subsequences ready for feature extraction (i.e., encoding) and classification. The new motif extraction algorithm and its counterpart, naive sliding-window match, are benchmarked on slipper datasets and other ones regarding runtime and efficacy in compressed learning.

1 Introduction

Internet of Things (IoT) has added unprecedented intelligence to the industry[21, 16], medicine[4, 3], in recent years. For example, industrial automation and monitoring now rely on the knowledge of multi-faceted sensory data collected from nodes continuously sampling temperature, pressure and vibration out of a fleet of operating machines. Conventionally, data are acquired and reported back to computational back-ends to perform classification, inference and recording. Unfortunately, this *dummy-leaf-node* model does not scale well if data sanitation and normalization, which may dominate overall budget of computation during data life-cycle, are not off-loaded to end-points. IoT end-points are battery powered, wireless sensing devices with weak computation and little memory. These devices gather information from environment intermittently with low deployment cost, comparing to large-scale sensing techniques like radars. However, the power of radio transmission dominates IoT devices’[17] overall power budget, because, by default, raw data, which includes noise and superfluous information is not compressed for transmission. It is expedient if a light-weight algorithm can be deployed on those end-points to trade power of transmission back with the cost of slightly more preprocessing.

Applying method from [14], we can encode a TS with an *envelope* into a sequence of 1/0/-1 for classification and owner identification. The envelope is equivalently an as raw-response from an observed instance. However, to achieve optimal discriminating performance, the envelope needs to be as thin as possible by averaging same-length, well-synchronized unit pattern. The envelope can encode the TS into a sparse representation which has many zeros. Thus it enables compressed

sensing[6] (CS). CS applies random projection on encoded TS to greatly reduce the amount of transmitted data and has little sacrifice on classification accuracy. So there is an urge to efficiently segment and identify motifs from long TS to build “thin” envelopes efficiently. Moreover, [5] had proved that it is practical to tackle machine learning tasks in compressed domain to reduce computations.

Therefore, *elastic motif segmentation and alignment* (EMSA) is proposed to efficiently identify unit patterns in TS without prior knowledge. The algorithm exploits features of TS to place time-anchors to align, normalize and extract repetitive subsequences. In the experiment, EMSA is compared with sliding window matching (SWM) technique on three datasets, namely accelerator data from walked slippers, opening and closing motion data from cabinet doors and ECG data etc[8].

2 Related Works

As mentioned earlier, SWM (formulated as algorithm 1) is a naive algorithm which compares a segment of times-series (“query”) with another segment (“candidate”) by sliding the query sequence over the candidate, with shrinking and stretching along the way. When the Euclidean distance of two sequences is smaller than a given threshold, the candidate sequence is appended to the output list of motifs. However, SWM does not guarantee phase invariance of the envelope if one long TS is divided into two and processed by SWM independently.

In enumerating motifs, distant-based techniques, such as Symbolic Aggregate Approximation (SAX)[15, 11, 19] and Motif Enumerator (MOEN) [18] are proved to be exact when the variance of unit pattern length is small. However, even allowing additional linear space for caching Euclidean distance of subsequences, distance-based algorithms usually operate under $O(n^3)$ time complexity. In addition to the intractability of computation on large datasets, distance-based algorithms work poorly if motifs are randomly scaled on time axis[20]. Take gene expression motif alignment as an example[1], normal unit patterns with different length may be categorized into independent groups if distances are computed without proper alignments.

Any distortion on time-axis, if not rectified by *dynamic time warping* (DTW)[12], may hamper accurate unit pattern recognition. Relaxing the discriminating threshold may also alleviate the issue with the cost of lower sensitivity on anomalies. Unfortunately, to apply DTW, one must know a priori that two sequences can be meaningfully matched. In [20, 10], DTW’s insufficiency is replaced by explicitly finding optimized uniform scaling distance $d_u(Q, C)$ on query sequence \mathbf{Q} and candidate sequence \mathbf{C} , before distance calculations. [10] uses lower-bounding technique to accelerate scale-and-match iterations, while [20] applied random projection on regular SAX to reduce computations of Euclidean distances: $d_u(\mathbf{Q}, \mathbf{C}) = \min_{m_q \leq s \leq m_c} \sum_{i=1}^s (q_i^s - c_i)^2$

In EMSA, TS is segmented with primary mathematical features of a sequence, before any stretching and shrinking. Clustering of extrema generates valuable insights on the recurring pattern of sequence, therefore speeds up following match-and-rescale with direct insights of the sequence itself.

3 Proposed Elastic Motif Segmentation and Alignment Algorithm

In TS classification, input data is expected to be a set of synchronized and equal-length TS such that the *envelope*[14], which is most probable ranges of a motif, can be generated by averaging the curves. However, real-world events, like walking gaits, usually vary in length. Thus, elastically rescale, and normalization of motifs are the critical steps before envelopes can be generated.

The envelope is necessarily a set of cascaded features encoding different meanings. For example, ECG does have the well-known envelope designated by P-QRS-T. Each segment corresponds to indications on various parts of a heart. Suppose we are extracting a patient’s ECG envelope, even if heartbeats are segmented, without proper alignment of time-anchors (the peaks of P, R, and T; the valleys of Q and S), the envelope may not possess high discriminating power, because segments may be smeared with each other. With interpolation and alignment, EMSA generates thin envelopes, which performs better in anomaly detection and owner signature verification.

Algorithm 1 Sliding window matching algorithm

```
1: procedure SWM( $t, s, d, m, M, r$ )  $\triangleright$  Raw time-series input with time-stamps  $t$  and signal  $s$ 
2:   distThres = d
3:   minLen = m
4:   maxLen = M
5:    $\mu \leftarrow \text{motif}(t, s)$ 
6:    $\mu_{o,pocket} \leftarrow []$ 
7:   score  $\leftarrow -1$ 
8:   for curLen in range(m, M) do
9:     for  $i \leftarrow 0, \lfloor \text{len}(\mu) / \text{curLen} \rfloor - 1$  do
10:       $\mu_o \leftarrow []$   $\triangleright$  Initialize output motif set as empty
11:       $\mu_q \leftarrow$  subsequence  $\mu$  from indexes  $[i \times \text{curLen} : (i+1) \times \text{curLen}]$   $\triangleright$  Get query motif
12:       $\mu_r \leftarrow$  cut away  $\mu_q$  from  $\mu$   $\triangleright$  Remaining TS
13:       $d = []$   $\triangleright$  Keep track of Euclidean distance between query and accepted motifs
14:       $p_d = 0$   $\triangleright$  Keep track of the number of dropped time-samples
15:      repeat
16:        for all  $j$  in  $[\text{len}(\mu_t) \times (100-r)/100 : \text{len}(\mu_q) \times (100+r)/100]$  do
17:           $\mu_c \leftarrow \mu_r[0:j]$ 
18:           $\mu_c \leftarrow$  map  $\mu_c$  to  $\mu_q$  time-steps
19:           $d_e \leftarrow \text{EuclideanDist}(\mu_q, \mu_c) / \text{len}(\mu_q)$ 
20:          if  $d_e < \text{distThres}$  then
21:             $\mu_o = \mu_o$  append  $\mu_c$   $\triangleright$  Accept motif candidate  $\mu_c$  under given condition
22:             $\mu_r \leftarrow$  cut away  $\mu_c$  from  $\mu_r$ 
23:             $d = d$  append  $d_e$ 
24:          else
25:             $\mu_r \leftarrow$  pop away first time sample from  $\mu_r$ 
26:             $p_d ++$ 
27:          end if
28:        end for
29:      until  $\text{len}(\mu_r) \times (100-r)/100 < \text{len}(\mu_t)$ 
30:    end for
31:     $N_o \leftarrow$  total number of extracted motifs in  $\mu_o$ 
32:    score'  $\leftarrow N_o / (\text{mean}(d) \times p_d)$ 
33:    if score'  $>$  score then
34:       $\mu_{o,pocket} \leftarrow \mu_o$ 
35:      score  $\leftarrow$  score'
36:    end if
37:  end for
38:  return  $\mu_{o,pocket}$ 
39: end procedure
```

3.1 Detailed Procedure of EMSA

As illustrated in fig. 1, EMSA can be divided into four parts: (1) identification and clustering of peaks and valleys (2) segmenting and labeling atomic motifs, (3) search of recurring patterns, and (4) rescaling of unit pattern to align peaks and valleys. Optional preprocessing steps, such as low-pass filtering (e.g., moving average) and discretization, can be added on for data sanitization. Algorithm 2 is a detailed formulation of the proposed technique.

3.1.1 Identification and clustering of peaks and valleys

In walking gait (Fig. 3(a)), each peak (A) and valley (B) corresponds to upward and downward leg movements, receptively. Unfortunately, peaks and valleys are not uniform regarding magnitude and interval. Local maximum of TS s at timestamp t_i , defined as $s(t_{i-k}) < s(t_i) > s(t_{i+k}), \forall k \in [i-m, i+m]$ in the range of $[i-m, i+m]$, and minimum are identified and accumulated to generate histogram, as shown in Fig. 3(b) to address these uncertainties. EMSA use all the peaks (most probable values) in maxima, and minima histograms for clustering. Furthermore, clusters in histograms are discriminated at the center of least probable range between extrema groups (e.g.

Algorithm 2 Elastic Motif Segmentation and Alignment Algorithm

```

1: procedure EMSA( $t, s$ ) ▷ Raw time-series data with time-stamps  $t$  and signal  $s$ 
2:    $(p, v) \leftarrow LOCAL\_EXTREMA(t, s)$ 
3:   ▷ Get local peak/valley values  $(p.s, v.s)$  and corresponding timestamps  $(p.t, v.t)$ 
4:    $([b_p, o_p], [b_v, o_v]) \leftarrow HIST([p, v])$  ▷ Histogram bin and occurrences
5:    $[h_p, h_v] \leftarrow HISTCLUST([b_p, o_p], [b_v, o_v])$ 
6:   ▷ Find thresholds of clusters in histogram
7:    $l \leftarrow ATOMLABEL(p, v, h_p, h_v)$  ▷ Convert signal into stream of labels
8:    $u \leftarrow MATCHPAT(l)$ 
9:   ▷ Find most probable unit pattern from atom stream, for example  $u = [AB, A, B]$ 
10:  for all  $u_i$  in  $u$  do
11:    repeat
12:       $m_{r,i} \leftarrow motif(EXPLODE(t, s, l, u_i))$ 
13:       $\mathbf{m}_r \leftarrow \mathbf{m}_r \text{ append } m_{r,i}$ 
14:      ▷ Accumulate motifs in a list, after exploding them from raw time-series
15:       $t \leftarrow t \text{ cut away } t_m$ 
16:       $s \leftarrow s \text{ cut away } s_m$ 
17:    until  $s$  contains no labels
18:     $m_{r,max} \leftarrow \text{nearest integer of maximum length of motif in } \mathbf{m}_r \text{ divided by } len(u_i)$ 
19:     $t_v \leftarrow linspace(0, len(u_i) \times m_{r,max})$  ▷ Generate synthetic time-step
20:    for all  $m$  in  $\mathbf{m}_r$  do ▷ Elastic rescale all motifs and align peaks and valleys
21:      for  $i \leftarrow 0, len(u_i) - 1$  do
22:         $\tau \leftarrow \text{subsequence m.time from index of } i^{th} \text{ label to } (i+1)^{th} \text{ label}$ 
23:         $\mu \leftarrow \text{subsequence m.signal from index of } i^{th} \text{ label to } (i+1)^{th} \text{ label}$ 
24:         $t_i \leftarrow t_v[i \times m_{r,max} : (i+1) \times m_{r,max}]$ 
25:         $r \leftarrow (\max(t_i) - \min(t_i)) / (\max(\tau) - \min(\tau))$ 
26:         $\tau \leftarrow r \times (\tau - \min(\tau))$ 
27:         $s[i] \leftarrow interp1d(\tau, \mu, t_i)$ 
28:      end for
29:       $\mathbf{m}_u \leftarrow \mathbf{m}_u \text{ append motif}(t_v, s[:])$ 
30:    end for
31:  end for
32:  return  $\mathbf{m}_u$ 
33: end procedure

```

Figure 1: Clustering peaks and valleys in histogram to identify motifs

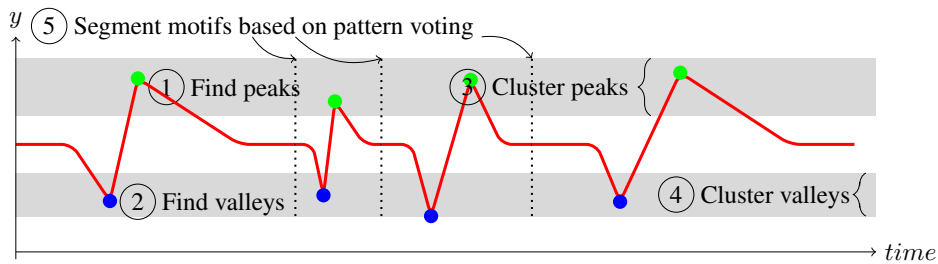


Figure 2: The walking gait accelerometer data example

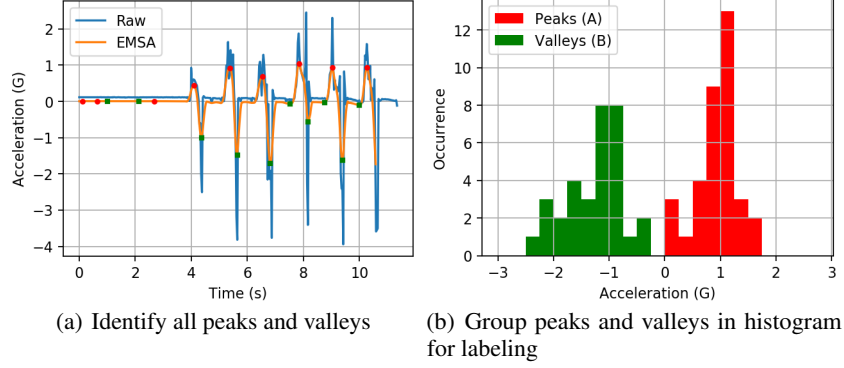
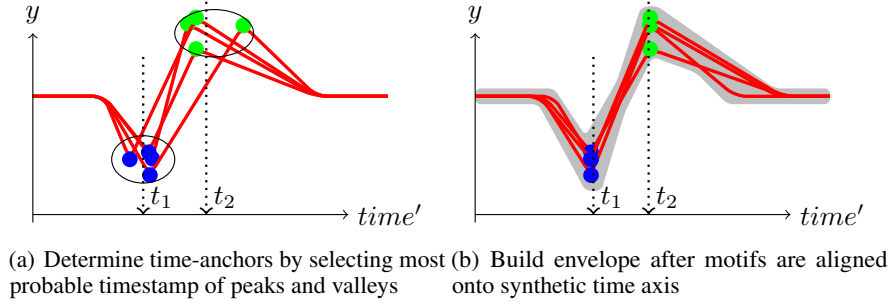


Figure 3: Elastic rescaling of motifs



center of $-0.25 \sim 0.25$ in Fig. 3(b)) by subroutine *HISTCLUST*. Divide-and-conquer can be applied to find local extrema under tight time complexity bound of $\Theta(n \lg n)$ [7].

3.1.2 Segmenting and labeling atomic motifs

Given a set of tuples (p, v, h_m, h_M) containing timestamps of extrema and thresholds splitting extrema in histograms, TS can be labeled by *ATOMLABEL* with characters on each atomic segments. For example, after segmenting and labeling, TS in Fig. 3(a) can be represented by character stream I , which is $\dots(\mathbf{AB})(\mathbf{AB})(\mathbf{AB})(\mathbf{AB})(\mathbf{AB})\dots$ (optional parentheses are included for clarity). *ATOMLABEL* effectively labels the TS with character “A” and “B” one the intervals from peaks and valleys to the right and left until it first met neighboring values in set h_m or h_M .

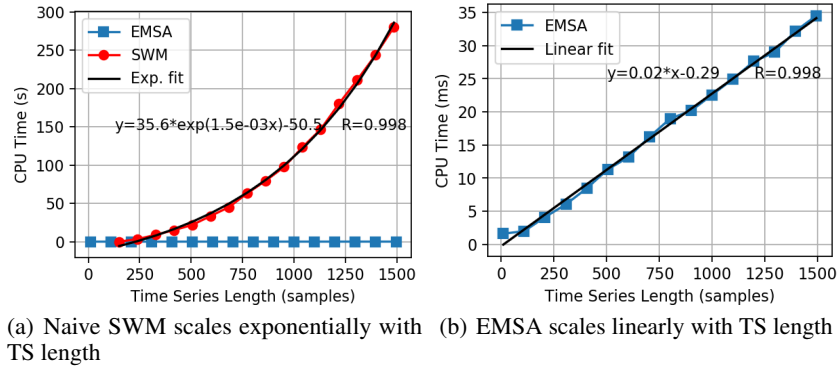
3.1.3 Search for recurring patterns

Original TS can be compressed into a very concise form of a cascaded sequence of scaled variants of unit pattern \mathbf{AB} . However, to robustly identify unit patterns, we use heuristics to select a set of unit patterns which maximally covered TS while producing a minimum number of dividable segments with subroutine *MATCHPAT*. For example, the TS can either be fully enumerated by motif set $\{\mathbf{A}, \mathbf{B}\}$ or $\{\mathbf{AB}\}$. The former set splits TS into ten subsequences while the latter only produces five. Both of the enumeration cover the whole TS. It is preferable to segment TS with longer motifs, as they more truly represent events with physical meanings.

3.1.4 Elastic rescaling and alignment of motifs

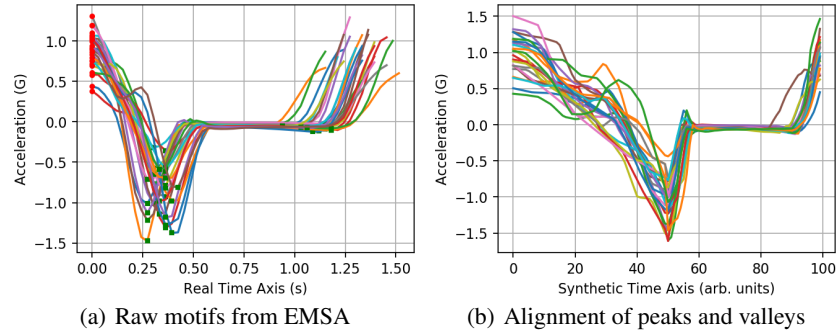
Finally, EMSA maps raw motifs, $m_{r,i}$ onto a *synthetic time-axis* t_v with *nearest-neighbor interpolation* under the condition of aligned time-anchors (fig. 4(a)), which is defined by the timestamps of peaks (p_i) and valleys (v_i) in the given raw motif. Time-anchors, namely “A” and “B” in the example, are pinned to exact timestamps on t_v to generate the thinnest *envelope* (fig. 4(b)). This technique can maintain the shape of original data after rescaling.

Figure 4: Runtime comparison on different lengths of time-series



(a) Naive SWM scales exponentially with TS length (b) EMSA scales linearly with TS length

Figure 5: Elastic rescaling of motifs on slipper accelerometer data



(a) Raw motifs from EMSA (b) Alignment of peaks and valleys

4 Experimental Results

Three real-world TS are segmented by EMSA and classified with compressed learning technique in [14]. Walking gait (a.k.a. slipper data) and cabinet door data are recorded with EcoBT Mini sensor (<http://epl.tw/ecobt-mini/>) with a sampling rate of 33 Hz (Datasets and demonstration video are available at <http://emsa.ml panda.rocks>). ECG data comes from PTB diagnostic ECG database [8], which samples electrical activity of human hearts under 1 kHz. The experiments are conducted with Python 3.6 with Scipy, on Intel i5-7640X CPU, 16G RAM in Arch Linux. Table 1 shows more than 5000X of acceleration to find and rescale unit patterns compared to SWM which is applied in [14].

To precondition slipper sensor signal, moving average (MA), elliptic, and Butterworth low-pass filters are tested. Although MA filter has a frequency response of sinc function, it smooths signal more naturally in time-domain, thus preserves critical features of peaks and valleys without the need of forward-backward filtering[9] to retain phase responses. In peak-valley search, different values of points on each side to consider do not dominate segmentation results as long as raw data is re-sampled at a frequency just high enough to represent the events. The number of bins in histogram effectively manifest into discretization of signals; we choose 0.2 (Fig. 3(b)) as bin-width in walking gait accelerometer data. Selection of larger bin-width implies that IoT end-points may adopt lower resolution ADC, which is cheaper and less power hungry, in data acquisition.

Performance of EMSA and SWM are compared in figure 5(a) and 5(b). Different lengths of time-series are both zero-meaned and low-pass filtered by five-point moving average. In figure 5(a), naive SWM is configured to search for query sequence of 50 points only, with Euclidean distance threshold of 0.062, and $\pm 10\%$ of motif length tolerance (i.e., each candidate motifs are shrunk and stretched by -10% to $+10\%$ then compared with query motif to find the best match). Naive SWM performs in exponential time with walking gait data. On the other hand, EMSA extracts peaks and valleys before segmentation and alignment. Also, all processing steps in EMSA are bounded in linear time, so the new algorithm offers significant speed-up when the input time-series is very long.

Table 1: Comparison of total CPU time required for motif segmentation

	SWM (s)	EMSA (s)	Speed Up
Slipper	422	0.0700	6029X
Door	4184	0.0362	115580X
ECG	14014	0.2340	59889X

Table 2: Accuracy of classification w.r.t. different compression ratios

Comp.	Slipper dataset				Door dataset				ECG dataset			
	0.1	0.2	0.5	1	0.1	0.2	0.5	1	0.1	0.2	0.5	1
Inst. 1	82.4	85.9	88.7	89.2	82	85	87.5	86.6	84.7	87.1	87.6	88.9
Inst. 2	85.5	89.7	90.5	92.3	97.9	98.8	99.1	99.3	86.8	86	87.3	87.7
Inst. 3	85.2	90.3	92.5	94.2	84.6	89.2	90.5	92.1	96.6	98.6	98.5	99.3
Inst. 4	83.5	87.6	89.6	90.5								

One critical aspect of EMSA is that output motifs are aligned and mapped to synthetic time-step, as shown in figure 6(b). Raw motifs in figure 6(a) are exploded from original time-series by peaks in the walking gait data. Therefore, the starting times are all marked with red circles, which denote peaks. EMSA then re-sample the interval between peak and valley by 50-point uniform time-steps. Likewise, the intervals from valley to the end of each motif are sampled by 50-point. All motifs are aligned to their peaks and valleys, thus yielding a sharp envelope of walking gait (as shown in difference between figure 6(a) and 6(b), after elastic rescaling, the first half contains more information than the latter half. Techniques of adaptively selecting time-steps will be left as future work).

Experiment results also show that envelopes generated by EMSA possess good discriminative capability. As shown in Table 2, even 90% of the encoded sequence is compressed during random projection, we can still achieve acceptable accuracy in classification.

5 Conclusions

EMSA is a novel time-domain feature extracting algorithm efficiently segment and elastically rescale motifs from TS into subsequences ready for envelope extraction. Experimental results demonstrate at least thousand times reduction in finding and rescaling motifs. EMSA enables and accelerates in-situ compressed sensing on IoT end-points, which are usually battery-powered and computation-limited. Applying multi-view ensemble learning[13, 2] to synergize with frequency-domain techniques, such as Fast Fourier Transform, will be left as future works.

6 Acknowledgments

This research is supported in part by the Ministry of Science and Technology of Taiwan (MOST 106-2633-E-002-001; MOST 106-2119-M-007-027). The authors would also like to thank National Taiwan University IoX Center, National Taiwan University of Science and Technology for the academical resources.

References

- [1] John Aach and George M. Church. “Aligning gene expression time series with time warping algorithms”. In: *BIOINFORMATICS* (Feb. 2001).
- [2] Rie Kubota Ando and Tong Zhang. “Two-view Feature Generation Model for Semi-supervised Learning”. In: *Proceedings of the 24th International Conference on Machine Learning. ICML ’07*. Corvallis, Oregon, USA: ACM, 2007, pp. 25–32. ISBN: 978-1-59593-793-3.
- [3] I. P. Androulakis, J. Vitolo, and C. Roth. “Selecting maximally informative genes to enable temporal expression profiling analysis”. In: *In Proceedings of Foundations of Systems Biology in Engineering*. 2005.

- [4] Ioannis P. Androulakis. “New approaches for representing, analyzing and visualizing complex kinetic mechanisms”. In: *European Symposium on Computer-Aided Process Engineering-15, 38th European Symposium of the Working Party on Computer Aided Process Engineering*. Ed. by Luis Puigjaner and Antonio Espuña. Vol. 20. Computer Aided Chemical Engineering Supplement C. Elsevier, 2005, pp. 235–240.
- [5] Robert Calderbank, Sina Jafarpour, and Robert Schapire. “Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain”. In: (2009).
- [6] Emmanuel J. Candes and Michael B. Wakin. “An Introduction To Compressive Sampling”. In: *IEEE Signal Processing Magazine* 25.2 (Mar. 2008), pp. 21–30. ISSN: 1053-5888.
- [7] Thomas H. Cormen et al. *Introduction to Algorithms*. 2nd. McGraw-Hill Higher Education, 2001. ISBN: 0070131511.
- [8] A. L. Goldberger et al. “PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals”. In: *Circulation* 101.23 (June 2000), e215–e220.
- [9] F. Gustafsson. “Determining the initial states in forward-backward filtering”. In: *IEEE Transactions on Signal Processing* 44.4 (Apr. 1996), pp. 988–992.
- [10] Eamonn Keogh. “Efficiently Finding Arbitrarily Scaled Patterns in Massive Time Series Databases”. In: *Knowledge Discovery in Databases: PKDD 2003: 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, Cavtat-Dubrovnik, Croatia, September 22-26, 2003. Proceedings*. Ed. by Nada Lavrač et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 253–265.
- [11] Eamonn Keogh, Jessica Lin, and Ada Fu. “HOT SAX: efficiently finding the most unusual time series subsequence”. In: *Fifth IEEE International Conference on Data Mining (ICDM’05)*. Nov. 2005.
- [12] Eamonn Keogh and Chotirat Ann Ratanamahatana. “Exact indexing of dynamic time warping”. In: *Knowledge and Information Systems* 7.3 (Mar. 2005), pp. 358–386. ISSN: 0219-3116.
- [13] Vipin Kumar and Sonajharia Minz. “Multi-view Ensemble Learning: An Optimal Feature Set Partitioning for High-dimensional Data Classification”. In: *Knowl. Inf. Syst.* 49.1 (Oct. 2016), pp. 1–59. ISSN: 0219-1377.
- [14] Y. J. Lee et al. “Compressed learning for time series classification”. In: *2016 IEEE International Conference on Big Data (Big Data)*. Dec. 2016, pp. 923–930.
- [15] Jessica Lin et al. “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. San Diego, California: ACM, 2003, pp. 2–11.
- [16] Alexander Maier, Sebastian Schriegel, and Oliver Niggemann. “Big Data and Machine Learning for the Smart Factory—Solutions for Condition Monitoring, Diagnosis and Optimization”. In: *Industrial Internet of Things: Cybermanufacturing Systems*. Ed. by Sabina Jeschke et al. Cham: Springer International Publishing, 2017, pp. 473–485.
- [17] B. Martinez et al. “The Power of Models: Modeling Power Consumption for IoT Devices”. In: *IEEE Sensors Journal* 15.10 (Oct. 2015), pp. 5777–5789. ISSN: 1530-437X.
- [18] Abdullah Mueen. “Enumeration of Time Series Motifs of All Lengths”. In: *2013 IEEE 13th International Conference on Data Mining*. Dec. 2013, pp. 547–556.
- [19] Yoshiki Tanaka, Kazuhisa Iwamoto, and Kuniaki Uehara. “Discovery of Time-Series Motif from Multi-Dimensional Data Based on MDL Principle”. In: *Machine Learning* 58.2 (Feb. 2005), pp. 269–300. ISSN: 1573-0565.
- [20] Dragomir Yankov et al. “Detecting Time Series Motifs Under Uniform Scaling”. In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’07. San Jose, California, USA: ACM, 2007, pp. 844–853.
- [21] S. Yin et al. “A Review on Basic Data-Driven Approaches for Industrial Process Monitoring”. In: *IEEE Transactions on Industrial Electronics* 61.11 (Nov. 2014), pp. 6418–6428. ISSN: 0278-0046.