

# Automated Anomaly Detection in Large Sequences

Paul Boniol, Michele Linardi, Federico Roncallo, Themis Palpanas  
EDF R&D; LIPADE, Université de Paris

boniol.paul@gmail.com; {michele.linardi, federico.roncallo}@parisdescartes.fr; themis@mi.parisdescartes.fr

**Abstract**—Subsequence anomaly (or outlier) detection in long sequences is an important problem with applications in a wide range of domains. However, current approaches have severe limitations: they either require prior domain knowledge, or become cumbersome and expensive to use in situations with recurrent anomalies of the same type. In this work, we address these problems, and propose NorM, a novel approach, suitable for domain-agnostic anomaly detection. NorM is based on a new data series primitive, which permits to detect anomalies based on their (dis)similarity to a model that represents normal behavior. The experimental results on several real datasets demonstrate that the proposed approach outperforms by a large margin the current state-of-the-art algorithms in terms of accuracy, while being orders of magnitude faster.

**Index Terms**—Data Series, Time Series, Anomaly discovery.

## I. INTRODUCTION

Massive collections of data series<sup>1</sup> are becoming a reality in virtually every scientific and social domain, and there is an increasingly pressing need for developing techniques that can efficiently analyze them [1], [2].

Anomaly, or outlier detection is an old problem [3]–[5], finding applications in a wide range of domains. In the specific context of sequences, which is the focus of this paper, we are interested in identifying anomalous subsequences, that is, unlike the outlier, not a single value, but a sequence of values.

Existing techniques either explicitly look for a set of pre-determined types of anomalies [6], [7], or identify as anomalies the subsequences with the largest distances to their nearest neighbors (termed *discords*) [5], [8]. We observe that these approaches pose limitations to the subsequence anomaly identification task, for several reasons, explained below.

First, the anomalous behavior is not always known. Therefore, techniques that use specific domain knowledge for mining anomalies (e.g., in cardiology [6], and engineering [9]) involve several finely-tuned parameters, and do not generalize to new cases and domains. Second, in the case of general, domain-agnostic techniques for subsequence anomaly detection, the state-of-the-art algorithms (e.g., [5], [8]) have been developed for the case of a *single* anomaly in the dataset, or multiple different (from one another) anomalies. The reason is that these algorithms are based on the distance of a subsequence to its Nearest-Neighbor (NN) in the dataset: the subsequence that has the farthest NN is marked as an anomaly.

Figure 1 depicts this situation. We show a snippet of the MIT-BIH Supraventricular Arrhythmia Database (MBA) ECG

<sup>1</sup>If the dimension that imposes the ordering of the sequence is time then we talk about *time series*. In the rest of this paper, we will use the terms *sequence*, *data series*, and *time series* interchangeably.

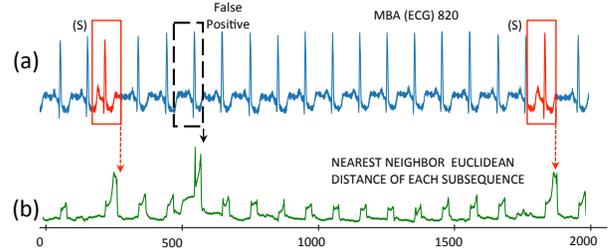


Fig. 1. (a) MBA ECG (2000 points snippet from patient 820), with two anomalous Supraventricular premature beats (S). (b) Euclidean distances of each subsequence (length 75) to its best non-trivial match in the full sequence: anomalies do *not* have the largest distance to their nearest neighbors.

recording [10], [11] of patient 820. This sequence includes *repeated* anomalous subsequences (*ventricular premature contractions*, marked by solid red rectangles). Following the state-of-the-art approaches [5], [8], we plot in Figure 1(b) the distance of each subsequence (of length 75) to its NN, and we observe that the (known) anomalies do not correspond to the most distant NN (i.e., the highest peak in Figure 1(b)). This is because our dataset includes several anomalies that are similar to one another (i.e., of the same type). At the same time, these approaches mark as outliers subsequences that are normal (dotted black rectangle), resulting in false positives.

Third, in order to remedy this situation, the  $m^{\text{th}}$  *discord* approach has been proposed [12]. This approach takes into account the multiplicity,  $m$ , of the anomalous subsequences that are similar to one another, and marks as anomalies all the subsequences in the same group, by computing the  $m^{\text{th}}$  (instead of the  $1^{\text{st}}$ ) NNs for each subsequence. Nevertheless, this approach assumes that we know the multiplicity  $m$ , which is not true in practice (otherwise, we need to re-execute the algorithms for several different  $m$  values).

In this work, we address the aforementioned problems, and propose NorM, a novel approach suitable for subsequence anomaly detection. The proposed approach allows us to detect anomalies based on their (dis)similarity to a model that represents the normal (expected) behavior. NorM starts by carefully selecting some of the subsequences of the dataset, based on a scoring mechanism. The selected set of subsequences are then used to build the normal behavior model, which is itself a sequence. This process is automatic (it uses the minimum description length principle to take some of the decisions), without the need for user intervention, and is effective even when the dataset contains multiple anomalies. At the end, NorM detects subsequence anomalies by comparing candidate

subsequences to this normal behavior model.

We experimentally demonstrate that NorM accurately identifies both single and repeated anomalies, and gracefully scales to large datasets, outperforming by a large margin the detection accuracy and time performance of the current state-of-the-art algorithms.

## II. PRELIMINARIES

A data series  $T \in \mathbb{R}^n$  is a sequence of real-valued numbers  $t_i \in \mathbb{R} [t_1, t_2, \dots, t_n]$ ;  $|T| = n$  is the length (or size) of  $T$ . We are typically interested in local regions of the data series, namely subsequences. A subsequence  $T_{i,\ell} \in \mathbb{R}^\ell$  of a data series  $T$  is a subset of contiguous values from  $T$  of length  $\ell$  (usually  $\ell \ll n$ ) starting at position  $i$ . Formally,  $T_{i,\ell} = [t_i, t_{i+1}, \dots, t_{i+\ell-1}]$ . The problem we are addressing in this work is the identification of anomalous subsequences (of a given length) within a long data sequence.

Given two sequences,  $A$  and  $B$ , of the same length,  $\ell$ , we can calculate their Z-normalized Euclidean distance,  $dist$ , as follows:  $dist(A, B) = \sqrt{\sum_i^{\ell} (\frac{A_{i,1} - \mu_A}{\sigma_A}, \frac{B_{i,1} - \mu_B}{\sigma_B})^2}$ , where  $\mu$  and  $\sigma$  represent the mean and standard deviation of  $A$  and  $B$ .

Given a subsequence  $T_{i,\ell}$ , we say that its  $m^{th}$  Nearest Neighbor ( $m^{th}$  NN) is  $T_{j,\ell}$ , if  $T_{j,\ell}$  has the  $m^{th}$  shortest distance to  $T_{i,\ell}$ , among all the subsequences of length  $\ell$  in  $T$ , excluding trivial matches [13]; a trivial match of  $T_{i,\ell}$  is a subsequence  $T_{a,\ell}$ , where  $|i - a| < \ell/2$  (i.e., the two subsequences overlap by more than half their length).

The state-of-the-art solutions for subsequence anomaly detection use the following definition for anomalies (*discords*):

*Definition 1 (discord [5], [8], [14]–[16]):* Among all subsequences of length  $\ell$  of series  $T$ , the subsequence  $T_{i,\ell}$  that has the largest distance to its NN is called a (data series) discord.

This is an intuitive definition: a subsequence is a discord if its NN is very far away. However, this definition fails when we have two neighboring discords, with a small distance to each other, and a very large distance to all the rest of the subsequences. In order to capture these situations, the  $m^{th}$ -discord has been proposed:

*Definition 2 ( $m^{th}$ -discord [12]):* Among all subsequences of length  $\ell$  of series  $T$ , the subsequence  $T_{i,\ell}$  that has the largest distance to its  $m^{th}$  NN is called an  $m^{th}$ -discord.

We now propose a definition that extends the previous two for the case of the  $k$  most significant anomalies:

*Definition 3 (Top- $k$   $m^{th}$ -discord):* A subsequence  $T_{i,\ell}$  is a Top- $k$   $m^{th}$ -discord if it has the  $k^{th}$  largest distance to its  $m^{th}$  NN, among all subsequences of length  $\ell$  of  $T$ .

Note that this definition subsumes the previous two: the simple discord is equivalent to Top-1 1<sup>st</sup>-discord, and the  $m^{th}$ -discord is equivalent to Top-1  $m^{th}$ -discord.

Even though discords have been extensively studied and used in the literature, they have several shortcomings that can severely limit their practical use.

## III. PROBLEM FORMULATION

We now formulate a new approach for subsequence anomaly detection, based on the notion of normal (expected) behavior.

Since we are interested in subsequence anomalies, we first define the set of all subsequences of length  $\ell$  in a given data series  $T$ :  $\mathbb{T}_\ell = \{T_{i,\ell} | \forall i. 0 \leq i \leq |T| - \ell + 1\}$ . In general, we assume that  $\mathbb{T}_\ell$  contains both normal and anomalous subsequences. We define normal behavior as follows:

*Definition 4 (Normal Model,  $N_M$ ):* Given a data series  $T$ ,  $N_M$  is a model that represents the normal (i.e., not anomalous) trends and patterns of  $T$ .

The above definition allows several interpretations, which can lead to different kinds of models. Subsequence anomalies can then be defined in a uniform way: anomalies are the subsequences that have the largest distances to the expected, normal behavior,  $N_M$  (or their distance is above a set threshold). In this work, we propose a formalization for  $N_M$  and experimentally demonstrate its effectiveness on all datasets in the literature used for subsequence anomaly discovery.

*Definition 5 (Subsequence Anomaly):* Given a data series  $T$ , the set  $\mathbb{T}_\ell$  of all its subsequences of length  $\ell$ , and the Normal Model  $N_M$  of  $T$ , the subsequence  $T_{j,\ell} \in \mathbb{T}_\ell$  with a distance to  $N_M$   $d = \min_{i \in [0, \ell_{N_M} - \ell]} \{dist(T_{j,\ell}, N_{M_{i,\ell}})\}$  is an anomaly if  $d$  is in the Top- $k$  largest distances among all subsequences in  $\mathbb{T}_\ell$ , or  $d > \epsilon$ , where  $\epsilon \in \mathbb{R}_{>0}$  is a threshold.

Note that the only essential input parameter<sup>2</sup> is the length  $\ell$  of the anomaly (which is also one of the inputs in all relevant algorithms in the literature [5], [8], [12], [14]–[17]).

As we elaborate later on, in this work we choose to define  $N_M$  as a sequence that summarizes normality in  $T$ , by representing the average behavior of a set of (ideally only) normal sequences. Intuitively,  $N_M$  is the data series, which tries to minimize the sum of Z-normalized Euclidean distances between itself and some of the normal subsequences in  $T$ . Last but not least, we need to compute  $N_M$  in an unsupervised way, i.e., without having normal/abnormal labels for the subsequences in  $\mathbb{T}_\ell$ .

Observe that this definition of  $N_M$  implies the following challenge: even though  $N_M$  summarizes the normal behavior only, it is computed based on  $T$ , which may include (several) anomalies. In our work, we address this challenge by taking advantage of the fact that anomalies are a minority class.

We can now define the problem we want to solve.

*Problem 1 (Subsequence Anomaly Detection):* Given a data series  $T$ , and the set  $\mathbb{T}_\ell$  of all its subsequences of length  $\ell$ , detect the subsequence anomalies in  $\mathbb{T}_\ell$ .

## IV. PROPOSED APPROACH

In this section, we describe NorM, our solution for automated subsequence anomaly detection.

NorM detects anomalies based on their distance from the *Normal Model* sequence. It takes as input a data series  $T$ , and the length  $\ell$  of the candidate anomalies. The algorithm first computes the Normal Model  $N_M$  based on  $T$ , and

<sup>2</sup>The parameter  $k$  (or  $\epsilon$ ) is not essential, as long as the algorithm can *rank* the anomalies. In practice, experts start by examining the most anomalous pattern, and then move down in the list (there is oftentimes no rigid threshold separating anomalous from non-anomalous behavior [3]). All anomaly discovery processes function this way.

subsequently detects and returns a ranked list of the anomalous subsequences in  $T$  based on  $N_M$ . We note that the length of the anomalies,  $\ell$ , is a user-defined parameter in all subsequence anomaly detection techniques, and can be set by the domain expert (e.g., in the case of electrocardiogram data, cardiologists are interested in analyzing heartbeats, which have a known length). The length of the Normal Model,  $\ell_{N_M}$ , needs to be larger than  $\ell$ . In our experiments, we use the default value  $\ell_{N_M} = 3\ell$ ; the results also show stable performance as  $\ell_{N_M}$  varies.

**[Computing the Normal Model]** Recall that  $N_M$  should capture (summarize) the normal behavior of the data. This may not be very hard to do for a sequence  $T$  that does *not* contain any anomalous subsequences. In practice however, we would like to apply the NorM approach in an unsupervised way on any sequence, which may contain several anomalies.

We compute the  $N_M$  sequence in three steps. First, we extract the subsequences that can serve as candidates for building the  $N_M$ . These candidates are either randomly selected from  $T$  (NorM-smpl), or correspond to motifs<sup>3</sup> (NorM-SJ). Then, we group these subsequences according to their similarity in a set of clusters  $\mathbb{C}$ , adopting a hierarchical clustering strategy, augmented by automated identification of the right number of clusters using the Minimum Description Length principle. The last step consists of scoring each cluster, and selecting the cluster that best represents normal behavior. Formally, for a given cluster  $c \in \mathbb{C}$ , we select the cluster that maximizes the following formula:  $Norm(c, \mathbb{C}) = \frac{Frequency(c)^2 \times Coverage(c)}{\sum_{x \in \mathbb{C}} dist(Center(c), Center(x))}$ , where  $Frequency(c)$  is the number of subsequences in  $c$ , and  $Coverage(c)$  is the time interval between the first and the last occurrence of a subsequence in  $c$ .

Based on the subsequences of the selected cluster, we build  $N_M$  by computing its centroid (mean subsequence).

**[Normal Model Based Anomaly Detection]** At this point, we have our Normal Model  $N_M$ , and we now discuss the problem of how to identify the anomalous subsequences in a series  $T$ . Intuitively, the anomalous subsequences are the ones that are far away from  $N_M$ . Our technique starts by considering the pairwise distances between each subsequence of length  $\ell$  in  $T$  to subsequences of the same length in  $N_M$ . This operation results in a meta-sequence,  $N_M \bowtie_{\ell} T$  where  $(N_M \bowtie_{\ell} T)_i, 1 = \min(dist(T_{i,\ell}, N_{M,1,\ell}), \dots, dist(T_{i,\ell}, N_{M,|N_M|-\ell+1,\ell}))$ . In other words, it contains at position  $i$  the nearest neighbor distance between subsequence  $T_{i,\ell}$  and any subsequence of the same length,  $\ell$ , in  $N_M$ .

These distances correspond to their degree of abnormality: the larger the distance is to all normal patterns, the more abnormal the subsequence is. We then extract the  $k$  subsequences of length  $\ell$ , which have the highest distances in  $N_M \bowtie_{\ell} T$  and rank them according to their distances, or extract all the subsequences with a distance larger than some threshold.

<sup>3</sup>Motifs of  $T$  are the subsequences that have the smallest distance to each other.

## V. EXPERIMENTAL EVALUATION

We implemented our algorithms in C (compiled with gcc 5.4.0) and Python 3.5, and used a server with Intel Xeon CPU E5-2650 2.20GHz and 250GB RAM.

**[Datasets]** We benchmark our system using annotated, real and synthetic datasets. Following previous work [18], we use several synthetic datasets with injected anomalies. We refer to those datasets using the label SRW-[# of anomalies]-[% of noise]-[length of anomaly]. Our real datasets are the following. Simulated engine disks data (SED) from the NASA Rotary Dynamics Laboratory [19], representing disk revolutions recorded over several runs (3K rpm speed). MIT-BIH Supraventricular Arrhythmia Database (MBA) [10], [11], which are electrocardiogram recordings from 5 patients, containing multiple instances of two different kinds of anomalies. Five additional real datasets from various domains that have been studied in earlier works [8], [20], and their anomalies are simple *discords* (usually only 1): aerospace engineering (Space Shuttle Marotta Valve [20]), gesture recognition (Ann’s Gun dataset [8]), medicine (Patient’s respiration measured by the thorax extension [20], ECG recordings qtb/se1102 [20]), and electrical consumption study (Dutch Power Consumption data [20]).

**[Measures]** We use the Top- $k$  accuracy measure to evaluate the effectiveness of the methods. Top- $k$  accuracy is defined as the number of correctly identified anomalies among the top- $k$  answers of the algorithm, divided by  $k$  (this corresponds to precision on the anomaly class  $TP_A / (TP_A + FP_A)$ , where  $TP_A$  is the number of detected true anomalies, and  $FP_A$  the number of false positives). We also measure time, in order to evaluate the efficiency and scalability of the methods.

**[Algorithms]** We compare NorM to the current state-of-the-art algorithms. We consider two techniques that enumerate Top- $k$  1<sup>st</sup> discords, GrammarViz (GV) [8] and STOMP [5]. Moreover, we compare NorM against the Disk Aware Discord Discovery algorithm (DAD) [12], which finds  $m^{th}$  discords. We also compare to Local Outlier Factor (LOF) [23] and Isolation Forest [24]. Finally, we use LSTM-AD [25], a *supervised* deep learning technique. Note that in contrast to all other methods, LSTM-AD is a *supervised* technique.

### A. Results

**[Critical Difference Diagram]** These experiments test the capability of each method to correctly retrieve the  $k$  anomalous subsequences in each dataset (for the *discord* based techniques, we consider the Top- $k$  1<sup>st</sup> *discord* and the  $m^{th}$  *discord* (with  $m = k$ )). After rejecting the null hypothesis using the Friedman test, we use the pairwise Post-Hoc Analysis to test to produce the critical difference diagram for the algorithms and datasets in our study. The critical difference diagram (see Figure 2) resulting from the Wilcoxon signed-rank test with  $\alpha = 0.05$  shows that NorM-SJ and NorM-smpl are the overall winners, and significantly better than all other methods.

Regarding LSTM-AD, we note that in general it is more accurate than the *discord* based algorithms. Nevertheless, we stress that LSTM-AD only achieves this performance, because

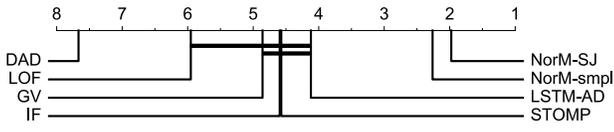


Fig. 2. Critical difference diagram.

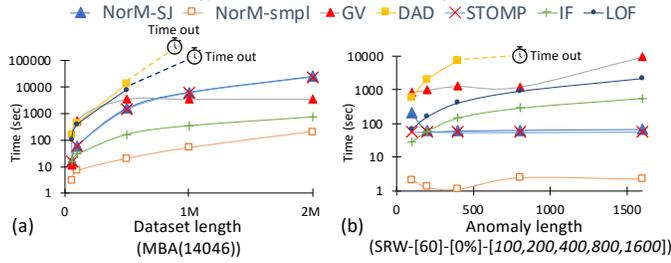


Fig. 3. Scalability: execution time vs (a) dataset size, (b) anomaly length. Timeout at 8 hours.

(contrary to the rest of the techniques) it benefits from a training phase on *labeled* data, which are often times not available.

**[Scalability Evaluation]** Figure 3(a) shows the execution time (seconds in log scale) of NorM and all the competitors, when varying the size of the dataset. We use several prefix snippets (50K to 2M points) of the real dataset MBA(14406), and we set  $k$  equal to the number of anomalies that are annotated in each snippet. Since NorM-smpl performs a limited number of distance calculations during both subsequence clustering (only few samples from the data series are selected) and anomaly scoring (limited to the subsequences in  $N_M$ ), We observe that it is 1-2 orders of magnitude faster than the competitors, and gracefully scales with the dataset size. Moreover, NorM-SJ that uses the STOMP algorithm for the Normal Model computation stage, has a small additional time overhead (when compared to STOMP). GV, DAD and LOF adopt different pruning strategies in order to reduce the number of Euclidean distance computations, which prove to be less effective. DAD and LOF, in particular, reach the time-out point (8 hours in our experiments) for datasets  $\geq 1M$  points.

Figure 3(b) depicts the time performance results as we vary the length of the anomalies between 100-1600 points (SRW-[60]-[0%]-[100-1600] datasets). The performance of STOMP is constant, because its complexity is not affected by the (anomaly) subsequence length. NorM remains relatively stable, since the Euclidean distances are computed using the STOMP algorithm. In NorM, only the clustering operations are affected by the length of the subsequences to consider, which in all experiments we ran was always a very small number ( $\sim 1-2\%$  of all subsequences). LOF and IF are computing distances using all overlapping subsequences and the computational time is therefore affected by their length. As shown in Figure 3(d), both of these two methods perform orders of magnitude worse than STOMP and NorM. GV and DAD do not scale with the anomaly length, either.

## VI. CONCLUSIONS

Even though the problem of anomaly detection in data series has attracted lots of attention, the techniques that have

been proposed so far fall short in terms of effectiveness and efficiency. In our work, we describe a novel approach that is based on the representation of normal behavior, which enables us to detect both single and recurrent anomalies, irrespective of the domain, and leads to superior accuracy and time performance.

## REFERENCES

- [1] T. Palpanas, "Data series management: The road to big sequence analytics," *SIGMOD Rec.*, vol. 44, no. 2, pp. 47–52, 2015.
- [2] T. Palpanas and V. Beckmann, "Report on the first and second interdisciplinary time series analysis workshop (itisa)," *SIGMOD Rec.*, "Accepted for publication, 2019.
- [3] V. Barnett and T. Lewis, *Outliers in Statistical Data*. John Wiley and Sons, 1994.
- [4] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos, "Online outlier detection in sensor data using non-parametric models," in *VLDB*, 2006.
- [5] C. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. Dau, D. Silva, A. Mueen, and E. Keogh, "Matrix profile I: all pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets," in *ICDM*, 2016.
- [6] M. Hadjem, F. Naït-Abdesselam, and A. A. Khokhar, "St-segment and t-wave anomalies prediction in an ECG data using rusboost," in *Healthcom*, 2016.
- [7] D. Abboud, M. Elbadaoui, W. Smith, and R. Randall, "Advanced bearing diagnostics: A comparative study of two powerful approaches," *MSSP*, vol. 114, 2019.
- [8] P. Senin, J. Lin, X. Wang, T. Oates, S. Gandhi, A. P. Boedihardjo, C. Chen, and S. Frankenstein, "Time series anomaly discovery with grammar-based compression," in *EDBT*, 2015.
- [9] J. Antoni and P. Borghesani, "A statistical methodology for the design of condition indicators," *Mechanical Systems and Signal Processing*, 2019.
- [10] G. et al., "Physiobank, physiotoolkit, and physionet," *Circulation*, vol. 101, 2000.
- [11] G. B. Moody and R. G. Mark, "The impact of the mit-bih arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, 2001.
- [12] D. Yankov, E. Keogh, and U. Rebbapragada, "Disk aware discord discovery: Finding unusual time series in terabyte sized datasets," in *ICDM*, 2007.
- [13] Y. Z. et al., "Matrix profile II: exploiting a novel algorithm and gpus to break the one hundred million barrier for time series motifs and joins," in *ICDM*, 2016.
- [14] Y. Liu, X. Chen, and F. Wang, "Efficient Detection of Discords for Time Series Stream," *Advances in Data and Web Management*, 2009.
- [15] Y. Bu, O. T. Leung, A. W. Fu, E. J. Keogh, J. Pei, and S. Meshkin, "WAT: finding top-k discords in time series database," in *SIAM*, 2007.
- [16] W. Luo and M. Gallagher, "Faster and parameter-free discord search in quasi-periodic time series," in *Advances in Knowledge Discovery and Data Mining*, 2011.
- [17] E. Keogh, S. Lonardi, C. Ratanamahatana, L. Wei, S.-H. Lee, and J. Hanley, "Compression-based data mining of sequential data," *DMKD*, 2007.
- [18] P. Senin, J. Lin, X. Wang, T. Oates, S. Gandhi, A. P. Boedihardjo, C. Chen, and S. Frankenstein, "Grammarviz 3.0: Interactive discovery of variable-length time series patterns," *TKDD*, 2018.
- [19] A. Abdul-Aziz, M. R. Woike, N. C. Oza, B. L. Matthews, and J. D. Lekki, "Rotor health monitoring combining spin tests and data-driven anomaly detection methods," *Structural Health Monitoring*, 2012.
- [20] E. J. Keogh, J. Lin, and A. W. Fu, "HOT SAX: efficiently finding the most unusual time series subsequence," in *ICDM*, 2005.
- [21] (2007) <http://data-acoustics.com/measurements/bearing-faults/bearing-4/>.
- [22] (2015) [http://www.nyc.gov/html/tlc/html/about/trip\\_record\\_data.shtml](http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml).
- [23] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," in *SIGMOD*, 2000.
- [24] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *ICDM*, ser. ICDM, 2008.
- [25] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *ESANN*, 2015.
- [26] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, 2017.