TRAINING LSTM FOR UNSUPERVISED ANOMALY DETECTION WITHOUT A PRIORI KNOWLEDGE

Yann Cherdo *Paul de Kerret* *Renaud Pawlak*

Mantu AI Lab, 950 Route des Colles, 06410 Biot, France

ABSTRACT

Unsupervised anomaly detection on time-series is widespread in the industry and an active research topic. Recently, impressive results have been obtained by leveraging the progresses of deep learning, and in particular through the use of Long Short Term Memory (LSTM) neural networks. Yet, latest state-of-the-art unsupervised LSTM-based solutions still require a priori knowledge about normality as they need to train the model on time-series without any anomaly. In contrast, we propose a novel anomaly detector, coined as *LSTM-Decomposed (LSTM-D)*, that does not require this normality knowledge. More specifically, we pre-process the time-series with a spectral based information reduction such that the LSTM-based detector receiving the time-series becomes less likely to *learn* the anomaly, and hence miss its detection. We motivate our intuitions through simple examples and verify the performance improvement with respect to state-of-the-art solutions in a reference and publicly available data set.

Index Terms— Anomaly Detection, Long Short Temporal Memory, Recurrent Neural Networks, Unsupervised Learning, Discrete Fourier Transform

1. INTRODUCTION AND RELATED WORKS

Anomaly Detection refers to the problem of finding anomalous patterns in a data. The definition of an anomalous pattern can vary from a use-case to another. When that definition is given with the data, explicitly or implicitly through labels, the problem is categorized as supervised anomaly detection [1]. When no such information is provided, the problem is called unsupervised anomaly detection and commonly refers to an anomaly being any *rare* pattern [1]. This approach also sometimes refers to an outlier detection [2].

In real use-cases, anomaly labels or any knowledge about the anomaly can be hard to obtain, which justifies the growing interest for unsupervised approaches [3]. To apply unsupervised anomaly detection on time-series, companies such as Amazon, Twitter, Etsy or Yahoo have developed their own models that are a mixture of classical statistical, decomposition and machine learning algorithms [4–7]. Some models are also bio-inspired, e.g., modeling the episodic memory of the cortex [8]. Recently, Convolutional Neural Networks (CNN) have been used to efficiently learn normal patterns and reveal anomalous ones [9]. One of the most recent state-of-the-art approach is LSTM based anomaly detection [10] that recently had a great impact in the unsupervised anomaly detection field [11]. In [12], LSTM neural networks are used to predict a time-series after training on a normal time-series without any anomaly. A threshold is then applied to the error between the prediction and the true time-series to find anomalous patterns. Some simple one-layer LSTM networks can be used as in [11] although stacked (or Deep) LSTM networks with several layers show better results, as in [12–14].

Although these approaches are denoted as *unsupervised* as they do not need any *labeled* anomaly, they still require to train on normal time-series. The goal of this paper is to tackle this limitation. More specifically, our main contributions are:

- We propose a novel anomaly detector, coined as *LSTM Decomposed (LSTM-D)* anomaly detector, which consists of a spectral and information reduction based algorithm that is applied before feeding the state-of-the-art LSTM based detector.
- We verify the improvement detection with respect to state-of-the-art algorithms on a publicly available data set. We furthermore empirically show a strong reduction of the computation time, due to sub-sampling of the input time-series.

2. SYSTEM MODEL AND PROBLEM FORMULATION

2.1. System Model

Following the conventional notations for anomaly detection [3, 12, 14], we consider a time-series $\mathbf{x} \in \mathbb{R}^{n_e}$ with $n_e \in \mathbb{N}$ being the number of samples, such that

$$\mathbf{x} = \{x_0, x_1, ..., x_{n_e-1}\}. \quad (1)$$

Each time-series sample $x_i, i \in \{0, \dots, n_e - 1\}$ is potentially corrupted by an anomaly that modifies its "normal behaviour" in the sense that the value at this point does not fit with the usual pattern of the time-series. The goal of the anomaly detector is then to associate to each sample $x_i, i \in \{0, \dots, n_e - 1\}$ an estimated binary anomaly label $\hat{a}_i \in \{0, 1\}$ to indicate

whether a sample is corrupted by an anomaly. These labels are then stacked together to form the vector

$$\hat{\boldsymbol{a}} = \{\hat{a}_0, \hat{a}_1, ..., \hat{a}_{n_e-1}\}. \quad (2)$$

To evaluate the anomaly detection with respect to the true labels $\boldsymbol{a}_h$, we follow common use and report Precision, Recall and f_1 metrics [9, 12].

2.2. Anomaly Detection using LSTM Networks [12]

We now review our baseline algorithm formed by a stacked LSTM based anomaly detector from [12, 14].

LSTM Predictor The first block of the anomaly detector is formed by an LSTM neural network, which is a recurrent neural network that has proven very efficient in capturing the temporal dependencies of a time-series using an internal memory [10, 15].

The LTSM neural network is trained on a *normal* time-series to predict x_i from past samples $\{x_{i-w-1}, ..., x_{i-1}\}$, for a given time window w. The intuition being that the LSTM has learned the properties of the time-series if it is capable of predicting the next sample from past ones. Clearly, the window width w is a parameter with a significant impact on the detection accuracy (See [13]).

Anomaly Detection By comparing the true value of the ith sample x_i with the predicted value, the prediction error for the ith sample is formed as:

$$e_i = |\hat{\mathbf{x}}_i - \mathbf{x}_i|. \quad (3)$$

Using this prediction error, a threshold function T_a is applied to decide whether the sample is an anomaly or not:

$$\mathrm{T}_a(e_i) = \{a_i = \mathbb{1}_{e_i > \theta^a}, \forall i \in \{0, n_e - 1\}\} \quad (4)$$

with the anomaly threshold $\theta^a \in [0, 1]$. This threshold is computed using Maximum Likelihood Estimation (MLE) on the prediction error so as to balance the ratio between false positive and missed detections.

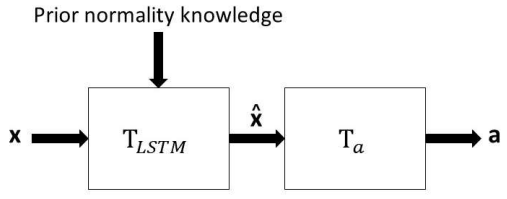


Fig. 1: Illustration of the LSTM anomaly detector from [12].

3. PROPOSED APPROACH: LSTM-DECOMPOSED

We now proposed a modified LSTM-based anomaly detector that is robust with respect to the lack of normality knowledge.

3.1. Some Insights

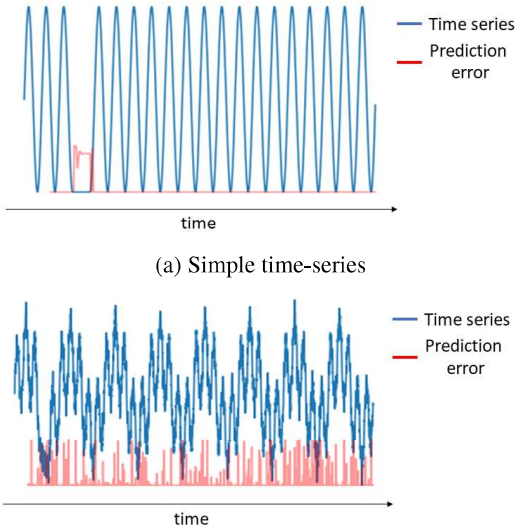


Fig. 2: LSTM prediction errors.

As a first step to gain intuition, we show in Fig. 2 the prediction error obtained when using the LTSM-based detector on a toy-example time-series. We first generate a time-series from a simple sinusoid such that $x(t) = \sin(2\pi f_0 t)$ with $f_0 = 1/500$ and then corrupt it with an anomaly making the time-series constant for some time. The exact same time-series is then enriched by adding two modes and some noise to obtain $x'(t) = x(t) + \sin(2\pi f_1 t) + \sin(2\pi f_2 t) + \eta$, with $f_1 = 1/100$, $f_2 = 1/20$, and η being a standard additive white Gaussian noise.

We can then observe that the trained LSTM discriminates the anomaly for the simple time-series but is unable to do so for the more complex one. LSTM networks work as black boxes such that it is hard to provide a definitive explanation. Yet, a tentative interpretation is that the LSTM network learns in a very sharp and precise manner a simple model, and hence discriminates well an anomaly. In contrast, it is forced to make a compromise between many elements when faced with a more complex time-series, and hence tends to learn more easily the anomaly, and consequently miss its detection.

3.2. LSTM-Decomposed (LSTM-D)

Building on the above intuition, we propose to introduce a pre-processing step before the LSTM network to simplify the

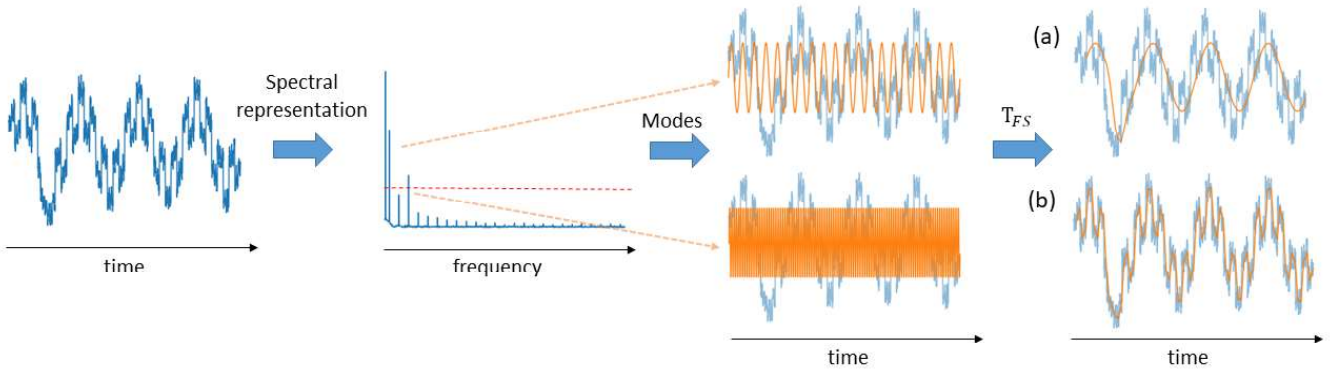


Fig. 3: Illustration of the LSTM-D anomaly detector.

time-series. This can be thought as zooming in or out to better catch an anomalous pattern with respect to others.

We propose to apply a bank of Low Pass Filters (LPF) to extract different views or features of the time-series. We will show later how the Discrete Fourier's Transform (DFT) can be used to find appropriate frequency cuts. We then apply the LSTM anomaly detection described in Section 2.2 on each of the filtered time-series.

Although this method appears at first sight to increase the complexity due to the multiplication of the detectors, it actually results for most cases in a strong complexity drop. This is due to the sampling rate reduction allowed by the LPF. Indeed, following the Nyquist-Shannon sampling theorem, sampling at a rate $2 \times f$ after LPF is sufficient to reconstruct the time-series [16]. In practice, we use a sampling rate of $f_s = w \times f$ with $w = 20$ in order to have these $w = 20$ points within every period of $\frac{1}{f}$.

Mathematically, this pre-processing is written as follows. Let us denote by $\mathcal{F}$ a set of dominant modes (see Section 3.3 for the frequency selection process). Then, $\forall f \in \mathcal{F}$, we denote by $\mathbf{x}_f$ the time-series obtained after LPF and sub-sampling, and we write

$$\mathbf{x}_f = \mathrm{T}_{\mathrm{FS}}(\mathbf{x}, f) \tag{5}$$

where T_{FS} represents the LPF followed by the sub-sampling described above.

As descriptions of resulting time-series for each frequency f are different, one value of θ_f^a is chosen for each frequency f. Without any a posteriori anomaly label, the system uses all frequencies within $\mathcal{F}$ and combines the anomalies detected for every frequency. A posteriori labels can also be used to find the best dominant mode(s). The process of the LSTM-D anomaly detector is summarized in Fig. 4.

3.3. Dominant Modes and DFT

The frequency set $\mathcal{F}$ is a key parameter that can be arbitrary chosen and we propose below one motivated design for

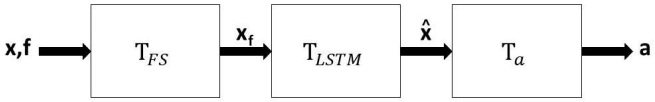


Fig. 4: The proposed LSTM-Decomposed anomaly detector.

choosing the frequencies. Due to the absence of *any a priori knowledge* on the time-series *and* on anomalies, we choose to focus on the main frequency components of the time-series. Our motivating intuition being that applying a LPF with the most significant frequencies allows to preserve the important information while simplifying the time-series.

Consequently, we compute the spectral representation of the time-series using a DFT [18]. Let us denote by $\{\tilde{X}[f_k]\}_{k=0}^{n_e-1}$ for $f_k = \frac{k}{n_e}$ the complex valued spectral representation with

$$\tilde{X}[f_k] = \sum_{m=0}^{n_e-1} x_m e^{-2\pi i m f_k}, \ \forall f_k \in \left\{0, \frac{1}{n_e}, \ldots, \frac{1}{2}\right\}. \tag{6}$$

Building on (6), we select the most significant modes, i.e., all modes whose amplitudes are above a given threshold denoted by $\theta^{\mathcal{F}}$ and computed using MLE. Mathematically, this is written as

$$\mathcal{F} = \left\{ f_k \middle| |\tilde{X}[f_k]| > \theta^{\mathcal{F}}, \forall f_k \in \left\{0, \frac{1}{n_e}, \ldots, \frac{1}{2}\right\}\right\}. \tag{7}$$

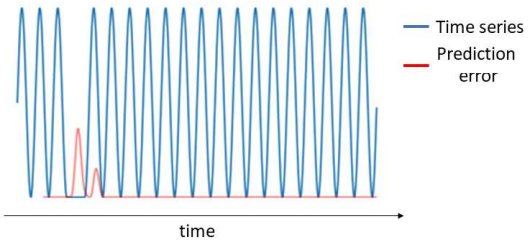
For clarity, the whole pre-processing is summarized in Fig. 3.

3.4. Qualitative Evaluation

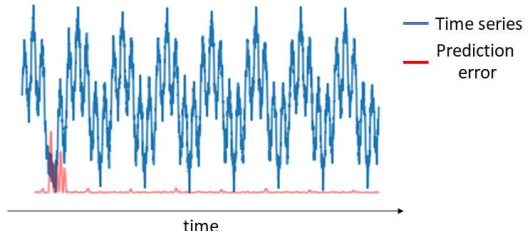
Before turning to a quantitative performance evaluation in the next section, we start with a qualitative evaluation through the toy example described in Section 3.1. The prediction error using the LSTM-D is then shown in Fig. 5 for both the simple and the more complex time-series. In contrast to the conventional LSTM-based detector, the proposed approach clearly finds the anomaly in the complex time-series.

Data	Model	Computation training time (s)	Precision	Recall	F1
ECG	LSTM-no prior knowledge	2661	0	0	0
ECG	LSTM-with prior knowledge	1911	1	1	1
ECG	**LSTM-D (proposed approach)**	**12**	**1**	**1**	**1**
PowerDemand	LSTM-no prior knowledge	2877	0.95	0.68	0.79
PowerDemand	LSTM-with prior knowledge	2355	0.95	1	0.97
PowerDemand	**LSTM-D (proposed approach)**	**3.5**	**0.99**	**1**	**0.99**
SpaceShuttle	LSTM-no prior knowledge	2057	1	0.84	0.91
SpaceShuttle	LSTM-with prior knowledge	1412	1	1	1
SpaceShuttle	**LSTM-D (proposed approach)**	**5**	**0.99**	**1**	**0.99**

Table 1: Performance evaluation on the dataset from [17]. Simulations were run on an Intel Core i7-7700 CPU 3.60GHz.



(a) Simple time-series



(b) Complex time-series

Fig. 5: LSTM-D prediction errors.

4. SIMULATION RESULTS

4.1. Approach

As a reference baseline, we use the anomaly detector from [12, 14] formed by a stacked LSTM (30 units on first layer, 20 units on second layer). To emphasize the contribution of our approach, we use the same architecture for the proposed LSTM-D model. Note that all approaches require to tune several hyper-parameters. In particular, the thresholds θ_f^a are optimized through a validation data set while LSTM from [12, 14] requires to also optimize the width w.

4.2. Datasets

For the performance evaluation, we use 3 datasets from the reference paper [12, 14]. The electrocardiograms (ECG) dataset contains a single anomaly corresponding to a preventricular contraction. The power demand dataset shows power consumption with normal behavior being weeks containing 5 peaks corresponding to weekdays followed by 2 dips corresponding to weekends. This dataset contains 3 anomalous patterns. Finally, the space shuttle Marotta valve dataset containing 3 anomalous regions.

It is important to note that anomaly labels are time windows containing an anomalous pattern. Consequently, we also aim at detecting anomalous windows and we consider that an anomalous patterns has been found when at least one point in that pattern has been detected as anomalous.

4.3. Performance Evaluation and Discussion

We can observe in Table 1 that the performance of the stacked LSTM from the literature heavily suffers from training on data with anomalies, i.e., from the lack of a priori normality knowledge. The coarse granularity in the objective values comes from the small number of anomalies in the data set. To get a more thorough performance evaluation, our next step will be to evaluate our approach on larger data sets, and in particular on the publicly available Yahoo Webscope data set[1]. Finally, we can observe a strong reduction of the computation time due to the important sub-sampling (See Section 3.2).

5. DISCUSSION AND FUTURE WORK

In this paper, we have introduced a spectral based preprocessing approach to make the state-of-the-art LSTM based anomaly detector robust to the lack of prior normality knowledge. Interestingly, the proposed approach also leads to a strong complexity reduction. This is a first approach towards the design of anomaly detectors without normality knowledge and many aspects remain to be further investigated. For example, the LPF could be replaced by a CNN as its properties are similar although sharing with the LSTM the same loss function.

[1]https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70

6. REFERENCES

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys*, vol. 41, no. 3, p. 15, 2009.

[2] Z. Ferdousi and A. Maeda, "Unsupervised outlier detection in time series data," *Proc. International Conference on Data Engineering (Workshops)*, 2006.

[3] A. Singh, "Anomaly Detection for Temporal Data using Long Short-Term Memory(LSTM)," *Master Thesis, KTH, School of Information and Communication Technology (ICT)*, vol. 7, 2017. [Online]. Available: http://www.diva-portal.org/smash/get/diva2:1149130/FULLTEXT01.pdf

[4] S. Guha, N. Mishra, G. Roy, and O. Schrijvers, "Robust random cut forest based anomaly detection on streams," *Proc. International Conference on Machine Learning (ICML)*, 2016.

[5] A. Kejariwal, "Twitter/AnomalyDetection," , 2015. [Online]. Available: https://github.com/twitter/AnomalyDetection

[6] A. Stanway, "Etsy skyline. Online Code Repos," , 2013. [Online]. Available: https://github.com/etsy/skyline

[7] N. Laptev, S. Amizadeh, and I. Flint, "Generic and scalable framework for automated time-series anomaly detection," *ACM SIGKDD International Conference*, pp. 1939–1947, August 2015.

[8] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing, Elsevier*, vol. 262, pp. 134–147, 2017.

[9] M. Munir, S. Siddiqui, A. Dengel, and S. Ahmed, "DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series," *IEEE Access*, vol. 7, pp. 1991–2005, January 2019.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *MIT Press*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] L. Bontemps, J. McDermott, and N. Le-Khac, "Collective anomaly detection based on long short-term memory recurrent neural networks," *Proc. International Conference of Future Data and Security Engineering*, 2016.

[12] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long Short Term Memory Networks for Anomaly Detection in Time Series," *Proc. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2015.

[13] P. Filonov, A. Lavrentyev, and A. Vorontsov, "Multivariate industrial time series with cyber-attack simulation: Fault detection using an lstm-based predictive data model," *arXiv*, 2016. [Online]. Available: https://arxiv.org/pdf/1612.06676.pdf

[14] S. Chauhan and L. Vig, "Anomaly detection in ECG time signals via deep long short-term memory networks," *Proc. International Conference on Data Science and Advanced Analytics (DSAA)*, 2015.

[15] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters, Elsevier*, vol. 42, pp. 11–24, 2014.

[16] W. Wu, "Discrete Sampling, Discrete Generalizations of the Nyquist-Shannon Sampling Theorem," *Stanford university*, 2010.

[17] A. F. Eamonn Keogh, Jessica Lin. (2005) Time Series Dataset. Consulted in October 2019. [Online]. Available: https://www.cs.ucr.edu/ eamonn/discords/

[18] J. Greenberg, "The Discret Fourier Transform," *MIT*, 2007.