ANOMALY DETECTION FOR TIME SERIES USING VAE-LSTM HYBRID MODEL

Shuyu Lin[1]*, Ronald Clark*[2]*, Robert Birke*[3]*, Sandro Schönborn*[3]*, Niki Trigoni*[1]*, Stephen Roberts*[1]

[1] University of Oxford, Oxford OX1 2JD, UK
[2] Imperial College London, South Kensington, London SW7 2AZ, UK
[3] ABB Future Labs, Segelhofstrasse 1K, 5404 Baden-Dättwil, Switzerland

ABSTRACT

In this work, we propose a VAE-LSTM hybrid model as an unsupervised approach for anomaly detection in time series. Our model utilizes both a VAE module for forming robust local features over short windows and a LSTM module for estimating the long term correlation in the series on top of the features inferred from the VAE module. As a result, our detection algorithm is capable of identifying anomalies that span over multiple time scales. We demonstrate the effectiveness of our detection algorithm on five real world problems and find our method outperforms three other commonly used detection methods.

Index Terms— Anomaly Detection, Time Series, Deep Learning, Unsupervised Learning

1. INTRODUCTION

Anomaly detection for time series is concerned with detecting unexpected system behaviours across time to provide informative insights. In many industrial applications, anomaly detection is used for monitoring sensor failures, alerting users of external attacks and detecting potential catastrophic events at an early stage [1]. Despite all the benefits, designing a good anomaly detection algorithm is extremely challenging. This is because the training data are often unbalanced with very few labelled anomalies. Furthermore, most anomalous behaviours are not known a priori and a good anomaly algorithm is expected to be able to detect even unseen anomalies. Due to these constraints, anomaly detection algorithms often have to be trained in an unsupervised fashion.

Broadly, we may characterize three types of anomalies that commonly occur in time series: namely point, context and collective anomalies [2]. Of the three types, *point anomalies* are the easiest to detect as data points can be treated independently during detection and no temporal relationships need to be considered. For this reason, simple threshold approaches or multi-layer perceptron (MLP) based methods [3] work relatively well for point anomalies. *Context and collective anomalies*, on the contrary, are more challenging. Context anomalies depend on the value of surrounding data points and thus local information is required for their detection and convolutional neural networks (CNNs) with larger receptive fields have been shown to work well in this case [4]. *Collective anomalies* occur when a series of data points together exhibits abnormal behaviours. As collective anomalies always occur in sequence over a reasonably long period, recurrent neural networks (RNNs), have been shown to be effective [5]. However, although a number of successful approaches have been proposed, none of the existing methods work well for all anomaly types.

In this paper, we propose a hybrid anomaly detection method that combines the representation learning power of a deep generative model - in the form of a variational autoencoder (VAE) - with the temporal modelling ability of a long short-term memory RNN (LSTM), as shown in Figure 1. Via the VAE module our model aims to capture the structural regularities of the time series over local windows, while the LSTM module attempts to model the longer term trend. Both the VAE and LSTM units do not require labelled anomalies for training. The code of our algorithm and experiments included in this paper is available at *https://github.com/lin-shuyu/VAE-LSTM-for-anomaly-detection*. In summary, our contributions are:

- We utilize a VAE model to summarize the local information of a short window into a **low-dimensional embedding**.
- We utilize a LSTM model, which acts on the low-dimensional embeddings produced by the VAE model, to manage the **sequential patterns** over longer term.
- The hierarchical structure allows us to detect **anomalies occurring over both short and long periods**.

The remainder of the paper is structured as follows. We first briefly outline VAE and LSTM models and ways in which they have been used for anomaly detection. We then present our hybrid VAE-LSTM model, followed by detection results given by our and other methods on real world time series. Finally we conclude with avenues for future research.

2. BACKGROUND AND RELATED WORK

In this section, we provide an overview of two machine learning models, namely VAEs and LSTMs, which serve as major

978-1-5090-6631-5/20/$31.00 ©2020 IEEE

building blocks for our anomaly detection algorithm. We also relate to existing anomaly detection algorithms.

VAE: VAEs [6, 7] are a type of generative probabilistic model known for learning embedding schemes that can infer the generation factors for the majority of the training data. This makes VAEs extremely suited for modelling the normal behaviours in an anomaly detection task. As a result, VAEs have been used for anomaly detection in various works with promising results [8, 9, 10, 11]. However, anomaly detection algorithms based only on VAEs often fail in detecting long-term anomalies, as VAE models cannot analyse information beyond a short local window. Our approach overcomes this limitation by using VAEs as local feature extractor and coupling it with a LSTM module to take care of long-term trends.

LSTM: LSTMs are a type of RNN that can capture long term dependencies in the input. This makes them ideally suited for our task where anomalies occur infrequently. Researchers have explored the idea of using RNN models for anomaly detection [5, 12]. Our method differs from those approaches in the way that our LSTM module is not applied to raw samples but to embeddings that represent a local window. Such setup makes our algorithm capable of ignoring redundant raw samples and tracking events over longer terms.

Hybrid: Hybrid models are a common approach for video analysis where a representation learning module is used to distil spatial information in a single image frame and a sequential module is applied to model the temporal correlation across a series of frames [13, 14]. [15] applies such a hybrid model to detect rare events in a surveillance video clip. The major difference between our approach and the hybrid models in video applications is that the representation learning module for videos is applied to an image, i.e. a data point at a single time stamp, whereas our representation learning module for time series processes a sequence of data points over a short period of time to form a local temporal basis for the latter sequential module to build upon.

3. OUR MODEL

Given a time series $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N\}$, where $\boldsymbol{x}_i \in R^m$ is a m-dimensional reading at the i-th time stamp that contains information about m different channels, we formulate the anomaly detection task as follows. At time t $(L \leq t \leq N)$, we are allowed to use a sequence of L past readings, i.e. $\boldsymbol{S}_t = [\boldsymbol{x}_{t-L+1}, \cdots, \boldsymbol{x}_t]$, to predict a binary output $y_t \in \{0, 1\}$ with 1 indicating an anomaly has occurred in $\boldsymbol{S}_t$. Such formulation allows our algorithm for *online detection*. Figure 1 gives an overview of our detection algorithm, which consists of a VAE module for extracting local features of a short window and a LSTM module for estimating the long-term trends. In this section, we will first introduce how these two modules are trained in an unsupervised manner and then explain how our algorithm is used for anomaly detection.

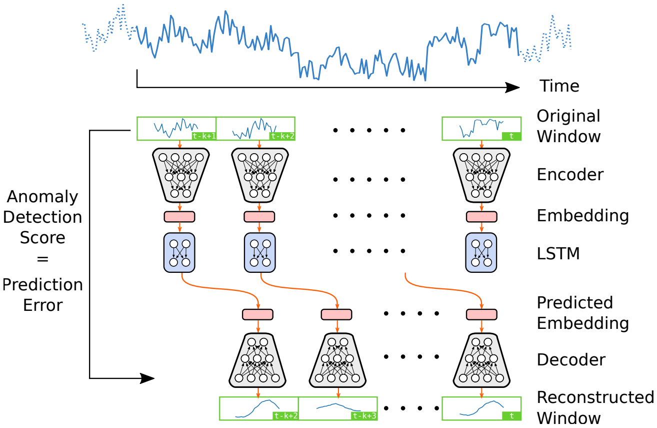


Fig. 1. Our VAE-LSTM model detects anomalies over a sequence of k consecutive windows of a given time series. i-th window $\boldsymbol{w}_i$ is encoded into a low-dimensional embedding $\boldsymbol{e}_i$, which is fed into a LSTM model to predict the next window's embedding $\hat{\boldsymbol{e}}_{i+1}$. The predicted embedding is then decoded to reconstruct the original window $\hat{\boldsymbol{w}}_{i+1}$. The reconstruction error serves as our anomaly detection score.

3.1. Training VAE-LSTM Models

To train our VAE-LSTM models in an unsupervised manner, we first need to separate a training and a test set from the given time series. An example of training-test set separation is given in Figure 2, where we take a continuous segment of the given time series which contains *no anomalies* as the training data and the rest of the time series with anomalies is kept as the test data for evaluation.

The VAE model consists of an encoder and a decoder. It takes a local window of p consecutive readings as input, estimates a low-dimensional embedding of q dimensions through the encoder and reconstructs the original window through the decoder. To train the VAE model, we generate rolling windows from the training data. For example, $\boldsymbol{w}_t = [\boldsymbol{x}_{t-p+1}, \cdots, \boldsymbol{x}_t]$ denotes the window ending at time t. The LSTM model operates on the VAE embeddings of a sequence of k non-overlapping windows. We use $\boldsymbol{W}_t = [\boldsymbol{w}_{t-(k-1)\times p}, \boldsymbol{w}_{t-(k-2)\times p}, \cdots, \boldsymbol{w}_t]$ to denote the sequence of windows ending at time t and $\boldsymbol{E}_t = [\boldsymbol{e}_t^1, \cdots, \boldsymbol{e}_t^k]$ to denote the corresponding embeddings in $\boldsymbol{W}_t$ with $\boldsymbol{e}_t^i$ indicating the embedding of the i-th window in $\boldsymbol{W}_t$. For a training data of N_{train} readings, we can generate $N_{\text{train}} - p$ rolling windows for training the VAE model and $N_{\text{train}} - pk$ rolling sequences for training the LSTM model. We reserve a randomly drawn 10% of the generated sequences from the training data as a validation set and all the windows and sequences in the validation set are excluded from training.

With the remaining windows in the training set, we optimise the VAE model parameters to maximise the ELBO loss defined in [16]. After the VAE model has been optimised, we use the encoder from the trained VAE model to estimate all

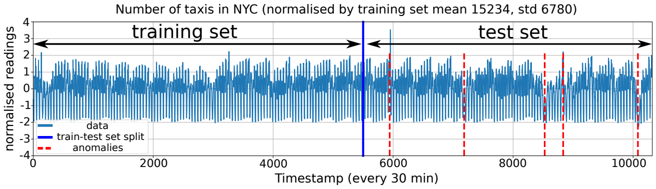


Fig. 2. An example of training and test set separation on the NYC taxi request time series.

the embedding sequences $\boldsymbol{E}_t$ in the training set. To train the LSTM model, we have the LSTM model take the first $k-1$ embeddings in a sequence $\boldsymbol{E}_t$ and predict the next $k-1$ embeddings, i.e.

$$[\hat{\boldsymbol{e}}_t^2, \cdots, \hat{\boldsymbol{e}}_t^k] = \text{LSTM}\big([\boldsymbol{e}_t^1, \cdots, \boldsymbol{e}_t^{k-1}]\big) \tag{1}$$

We optimise the LSTM model parameters by minimizing the prediction error of the final embedding, i.e. $\min \|\hat{\boldsymbol{e}}_t^k - \boldsymbol{e}_t^k\|$. Notice that all the model parameters for both VAE and LSTM units are optimised without anomaly labels.

3.2. Anomaly Detection using the VAE-LSTM Model

After training, our VAE-LSTM model can be used for anomaly detection in real time. At time t, the VAE-LSTM model analyses a test sequence $\boldsymbol{W}_t$ that contains $k \times p$ past readings tracing back from t. Our model first uses the encoder from the VAE to estimate the sequence of embeddings $\boldsymbol{E}_t$ in $\boldsymbol{W}_t$. Then it feeds the first $k-1$ embeddings to the LSTM model to predict the next $k-1$ embeddings $[\hat{\boldsymbol{e}}_t^2, \cdots, \hat{\boldsymbol{e}}_t^k]$, as given in Equation (1). Finally, our model reconstructs the last $k-1$ windows using the predicted embeddings and the VAE decoder, i.e.

$$\hat{\boldsymbol{w}}_{t-(k-i)\times p} = \text{Decoder}(\hat{\boldsymbol{e}}_t^i), \;\text{for}\; i = 2, \cdots, k. \tag{2}$$

With the reconstructed windows, we can define a score function d_t to flag anomalous behaviours by summing up the prediction errors for $\boldsymbol{W}_t$, i.e.

$$d_t = \sum_{i=2}^{k} \|\hat{\boldsymbol{w}}_{t-(k-i)\times p} - \boldsymbol{w}_{t-(k-i)\times p}\|_2 \,. \tag{3}$$

To detect an anomaly, we also need to define a threshold θ on the score function d_t, above which we will flag an anomaly alert $y_t = 1$ at the current t. The corresponding sequence $\boldsymbol{W}_t$ will mark a suspicious window where the anomaly event might have occurred. When there is sufficient data, we should determine θ using a validation set that contains both normal and anomalous examples. The threshold that gives the best performance, such as F1 score or other metrics, on this validation set is the detection threshold for the given time series. When there is limited data, we can use a validation that only contains normal samples to evaluate the distribution of the score function and define the threshold as a given percentile of this distribution, for example.

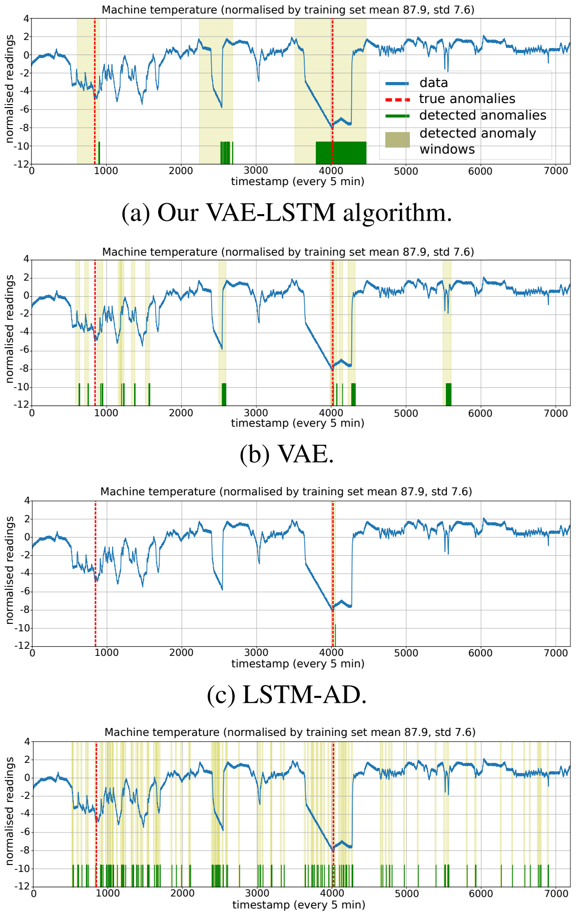
Table 1. Precision, recall and F1 score at the threshold that gives the best F1 score (L: detection window length).

Dataset	Method	L	Prec	Recall	F1
Ambient temperature	Ours	168	0.806	**1.0**	**0.892**
	VAE [8]	24	0.686	0.5	0.573
	LSTM-AD [5]	24	**1.0**	0.5	0.666
	ARMA [17]	24	0.184	**1.0**	0.311
CPU utilization AWS	Ours	144	**0.694**	**1.0**	**0.819**
	VAE [8]	24	0.348	0.5	0.410
	LSTM-AD [5]	24	0.274	**1.0**	0.430
	ARMA [17]	24	0.234	**1.0**	0.380
CPU utilization EC2	Ours	192	0.993	**1.0**	**0.996**
	VAE [8]	24	0.949	**1.0**	0.974
	LSTM-AD [5]	24	**1.0**	0.436	0.608
	ARMA [17]	24	0.938	**1.0**	0.968
Machine temperature	Ours	288	0.559	**1.0**	**0.717**
	VAE [8]	48	0.211	**1.0**	0.207
	LSTM-AD [5]	48	**1.0**	0.5	0.667
	ARMA [17]	48	0.142	**1.0**	0.248
NYC taxi	Ours	168	0.961	**1.0**	**0.980**
	VAE [8]	24	0.662	0.8	0.725
	LSTM-AD [5]	24	**1.0**	0.2	0.333
	ARMA [17]	24	0.769	0.4	0.526

4. EXPERIMENTS AND RESULTS

We evaluate our VAE-LSTM algorithm on five real world time series with actual anomaly events: ambient temperature in an office, CPU usage from Amazon Web Services (AWS) and from a server in Amazon's East Coast data center, internal temperature of an industrial machine and number of taxi passengers in New York City [18]. We compare our algorithm with three other commonly used anomaly detection algorithms for time series: VAE [8], LSTM-AD [5] and ARMA [17]. Table 1 lists the numerical results as well as the detection window length. We evaluate three metrics - precision, recall and F1 score (all metrics are evaluated at the threshold that gives the best F1 score). The detection window length for each dataset is chosen to be equal across all methods. Ours appears longer because of the hierarchical structure of our model which allows us to detect events over longer periods. Counting the true and false positives/negatives among the detection results can be difficult, as anomalous events occur only at a single time stamp whereas all detection algorithms reason over a window. We adopt a simple strategy proposed by [11] to alleviate this issue.

A visualisation of the anomaly detection results given by all four methods on the machine temperature series is shown in Figure 3. LSTM-AD achieves a high precision in most datasets but suffers from low recall, indicating accurate detected anomalies but high chance of missing out true anomalies (Figure 3.c). VAE has a good recall but low precision, indicating a significant amount of false positive detections (Fig-



(a) Our VAE-LSTM algorithm.

(b) VAE.

(c) LSTM-AD.

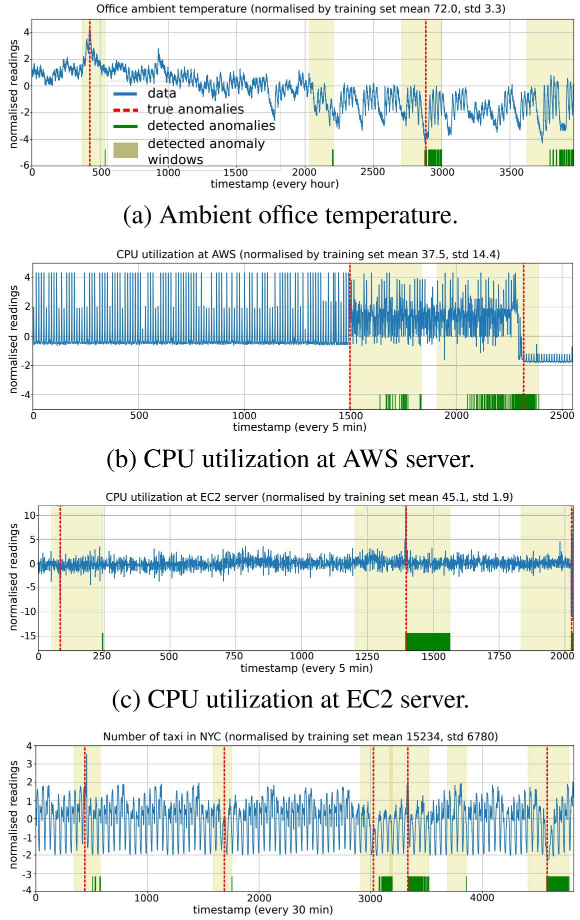
(d) ARMA.

Fig. 3. Anomalies detected using our VAE-LSTM, VAE, LSTM-AD and ARMA methods on the industrial machine temperature series. Blue line: time series, red dashed line: ground truth anomalies, short green lines: detected anomalies, light yellow windows: detected anomaly windows.

ure 3.b). ARMA does not perform well in either precision and recall and its detection is clearly the worst as (Figure 3.d).

In contrast, our VAE-LSTM algorithm achieves 100% recall for all datasets, meaning no missed anomaly and the ability to detecting all types of anomalies. At the same time, our method also achieves a decent precision, indicating that the number of false positives is low. For example, a false positive is reported in the ambient temperature series (Figure 4.a) around time $t = 2000$. From visual inspection there exists an unusual spike in the highlighted window and, hence, it might be sensible to raise up human supervisors' attention. We would argue such precaution is indeed beneficial in failure critical scenarios. Good performance in both precision and recall results in a high F1 score achieved by our method, leading all other methods by a good margin. Examples of our method's detection result are given in Figures 3a and 4.

A potential downside of our method is the delay in the anomaly detection for some cases. For example, the first anomaly in the EC2 CPU utilization series is only detected after about 150 time steps. This could be alleviated using multiple scale windows and we leave this for future research.



(a) Ambient office temperature.

(b) CPU utilization at AWS server.

(c) CPU utilization at EC2 server.

(d) The number of NYC taxi passengers.

Fig. 4. Anomalies detected by our VAE-LSTM hybrid model.

5. CONCLUSION

In this work, we propose a VAE-LSTM hybrid model as an unsupervised learning approach for anomaly detection in time series. Our model utilizes both a VAE module for forming robust local features over a short window and a LSTM module for estimating the long term correlations in the sequence. As a result, our detection algorithm is capable of identifying all types of anomalies that might span over multiple time scales. We demonstrate the effectiveness of our detection algorithm on five real world sequences and show that our method outperforms other commonly used detection methods.

Acknowledgement

This work was supported by the EPSRC Centre for Doctoral Training, EP/L015897/1, and the China Scholarship Council. We thank the reviewers for their useful comments.

6. REFERENCES

[1] Markus Goldstein and Seiichi Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data.," *PloS one*, vol. 11 4, pp. e0152173, 2016.

[2] Varun Chandola, Arindam Banerjee, and Vipin Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 15, 2009.

[3] Anup K Ghosh and Aaron Schwartzbard, "A study in using neural networks for anomaly and misuse detection.," in *USENIX security symposium*, 1999, vol. 99, p. 12.

[4] Tailai Wen and Roy Keyes, "Time series anomaly detection using convolutional neural networks and transfer learning," *IJCAI'19 Workshops*, 2019.

[5] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proceedings*. Presses universitaires de Louvain, 2015, p. 89.

[6] Diederik P. Kingma and Max Welling, "Auto-encoding variational bayes," *CoRR*, vol. abs/1312.6114, 2013.

[7] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, 2014, pp. 1278–1286.

[8] Jinwon An and Sungzoon Cho, "Variational autoencoder based anomaly detection using reconstruction probability," 2015.

[9] Suwon Suh, Daniel Chae, Hyon-Goo Kang, and Seungjin Choi, "Echo-state conditional variational autoencoder for anomaly detection," 07 2016, pp. 1015–1022.

[10] Yuta Kawachi, Yuma Koizumi, and Noboru Harada, "Complementary set variational autoencoder for supervised anomaly detection," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2366–2370, 2018.

[11] Haowen Xu, Yang Feng, Jie Chen, Zhaogang Wang, Honglin Qiao, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, and et al., "Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications," *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*, 2018.

[12] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '18*, 2018.

[13] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni, "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2043–2050.

[14] Ronald Clark, Sen Wang, Hongkai Wen, Andrew Markham, and Niki Trigoni, "Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[15] Yong Shean Chong and Yong Haur Tay, "Abnormal event detection in videos using spatiotemporal autoencoder," in *International Symposium on Neural Networks*. Springer, 2017, pp. 189–196.

[16] Shuyu Lin, Stephen Roberts, Niki Trigoni, and Ronald Clark, "Balancing reconstruction quality and regularisation in elbo for vaes," 2019.

[17] Brandon Pincombea, "Anomaly detection in time series of graphs using arma processes," 2007.

[18] Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, 2017.