STAD: State-Transition-Aware Anomaly Detection Under Concept Drifts

Bin Li[1], Emmanuel Müller[2]

[1]TU Dortmund, Otto-Hahn-Straße 14, 44227 Dortmund, Germany
[2]TU Dortmund, Otto-Hahn-Straße 14, 44227 Dortmund, Germany

Abstract

The detection of temporal abnormal patterns over streaming data is challenging due to volatile data properties and lacking real-time labels. The abnormal patterns are usually hidden in the temporal context, which can not be detected by evaluating single points. Furthermore, the normal state evolves over time due to concept drift. A single model does not fit all data over time. Autoencoders are recently applied for unsupervised anomaly detection. However, they usually get expired and invalid after distributional drifts in the data stream. In this paper, we propose an autoencoder-based approach (STAD) for anomaly detection under concept drift. In particular, we use a state-transition-based model to map different data distributions in each period of the data stream into states, thereby addressing the model adaptation problem in an interpretable way. We empirically demonstrate the state transition process and evaluate the anomaly detection performance on the Covid-19 dataset of Germany.

Keywords

State transition, Anomaly detection, Concept drift, Autoencoder

1. Introduction

Anomaly detection in streaming data is gaining traction in the current big data research. Despite the high demand in a variety of real-world applications [1] (e.g., health care, device monitoring and predictive maintenance), rare existing models show convincing performance in real-time deployment. The detection of abnormal patterns in streaming data is challenging. On the one hand, labels are unavailable or expensive to acquire in real-time, such that supervised approaches usually fail. On the other hand, the conventional batch models easily get expire, while a single stationary model does not fit the ever-changing data stream.

Recently, autoencoders have been employed for anomaly detection in an unsupervised manner [2]. Autoencoders are trained to reconstruct the normal data[1], such that for any unknown data instance, a high reconstruction error indicates an anomaly. Specifically, for time series data, the temporal dependencies between data points can be captured by constructing autoencoders using Recurrent Neural Networks (RNNs) and their variants [3, 4]. Although such methods show

OLUD 2022: First Workshop on Online Learning from Uncertain Data Streams, July 18, 2022, Padua, Italy
bin.li@tu-dortmund.de (B. Li); emmanuel.mueller@tu-dortmund.de (E. Müller)
http://ls9-www.cs.tu-dortmund.de/ (E. Müller)
0000-0002-9707-4596 (B. Li)

[1]Unless specifically stated, instead of normally distributed data, normal data refers to the opposite of abnormal data in the anomaly detection context.

impressive performance on time series data, they usually ignore that such data are commonly collected in a streaming way and do not allow full access during the training phase. Therefore, an adaptive autoencoder is desired, which can be initialized with a few normal data and be updated according to the real-time data distribution changes.

Another major challenge of anomaly detection in streaming data is distinguishing between abnormal patterns and concept drifts. Once the data stream drifts to a novel distribution, a stationary model trained only on outdated data may detect most of the upcoming data undesirably as anomalies.

Given the severe problems, our goal is to consider the concept drift detection and anomaly detection as a whole, adapt the model to the latest data distribution, and detect anomalies only concerning the temporal context where they are located. Previous concept drift researches focus on detecting changes of the joint probability $P(X, y)$ under supervised setting, namely, the decision boundary changes along with the distributional changes in the input data [5]. However, for anomaly detection, the class distribution between normal and abnormal is extremely unbalanced, and labels are usually missing, so it is impractical to use traditional supervised approaches [6, 7], e.g., detect drifts based on the changes of real-time prediction error rate. Instead, the adaptation based on changes of the prior probability $P(X)$ will ensure the autoencoder reconstructs the normal data in from the current concept. Statistical tests are commonly used for unsupervised drift detection [8]. For instance, the two-sample tests examine whether samples from two collections are generated from the same data distribution. However, existing methods conduct tests mostly in the original input space, which only works for linearly detectable drifts. Ceci et al. [9] introduce both PCA and autoencoder to embed features into a latent space for the change detection. However, their change detector is distance-based and highly depends on a user-defined threshold.

In this paper, we propose STAD (State-Transition-aware Anomaly Detection). In STAD, data distribution in a time period is defined as a *states*. We use *state transitions* to model the concept drifts between periods. As autoencoders are well studied for non-linear time series anomaly detection, we are motivated to extend the state transition paradigm to autoencoders. We follow the standard usage of autoencoders for anomaly detection and novelly couple the detection of concept drifts and anomalies with the informative latent representation of autoencoders. An existing autoencoder can be reused when a data concept reappears in the stream. A state transition is triggered by the detection of concept drift, and this will further guide the reuse or adaptation of autoencoders for the next period. The states quantify the uncertainty caused by concept drifts and raise interpretability in understanding the decision of autoencoders and changes in the data stream.

2. Problem definition

2.1. Terminology

2.1.1. Data Stream

Let $\mathcal{X} = \{X_t\}_{t\in\mathbb{N}}^{D}$ be a D-dimensional data stream, where X_t denotes the observation at timestamp t. The data stream contains unlabeled anomalies as well as distributional changes

caused by concept drifts. Instead of explicitly categorizing different concept drift types [5], we uniformly consider that a concept drift occurs in the data stream between timestamps t and $t+c$ if the prior probability $P_{<t}(X) \neq P_{>t+c}(X)$, where $P_{<t}$ and $P_{>t+c}$ are respectively the data distribution from the last concept drift to t and from $t+c$ to the next concept drift. The period $[t, t+c]$ is the *drift period*, which is defined as the minimum period that covers the whole distributional change. The data distribution other than drift periods is assumed to be stable. Due to the lack of labels under the unsupervised setting, we only consider the prior (virtual) shifts [5] in the data stream.

2.1.2. State transition

Imitating the automata theory, we formulate concept drifts in streaming data with a state transition model $\mathcal{M} = \langle \mathcal{X}, \mathcal{S}, \delta \rangle$ where $\mathcal{X}$ is a multivariate data stream, $\mathcal{S} = \{S_1, S_2, ..., S_N\}$ is a set of states (N is the user-defined maximum number of states that can be maintained), δ is a set of transition functions $\delta : \{S_i \Rightarrow S_j\}(S_i, S_j \in \mathcal{S}, i \neq j)$. For each state $S_i = \langle P_i, AE_i \rangle (i = 1, ..., N)$, P_i is the empirically estimated distribution in latent space, AE_i is the autoencoder trained on the new concept data. In this work, we assume that sufficient data after the concept drift is available to learn P_i and AE_i.

Considering that no information about the upcoming new concept is accessible, despite a potential high error rate, we still keep using the previous model for anomaly detection until the model adaptation is finished. Or in other words, the previous model is used during the upcoming drift period. For distributional stationary data streams where no concept drift occurs, there will be only a single state without transition, and the model reduces to a single conventional autoencoder.

2.1.3. Anomaly

An observed data snippet $X_t^{t+a} = \{x_t, ..., x_{t+a}\}(t, a \in \mathbb{N})$ is abnormal if it is significantly deviated from its temporal neighbors (data snippets in the same *state*). The significance of the deviation can be determined by thresholding or statistical techniques. Both concept drifts and anomaly snippets are distributionally deviate from their temporal neighbors. In our study, we distinguish them in terms of length. After the concept drifts, we assume that the data distribution stays stationary in the new concept for a significantly longer period. In contrast, the data stream returns to the previous distribution after a short anomaly snippet.

2.2. Problem statement

Given a D-dimensional data stream $\mathcal{X} = \{X_t\}_{t \in \mathbb{N}}^D$, we aim to identify any period $[t, t+a]$ where the corresponding data snippet X_t^{t+a} is abnormal. The detection process should be unsupervised and in real-time. We also detect concept drifts in the data stream and switch to an existing autoencoder or train a new one on the newly arrived data.

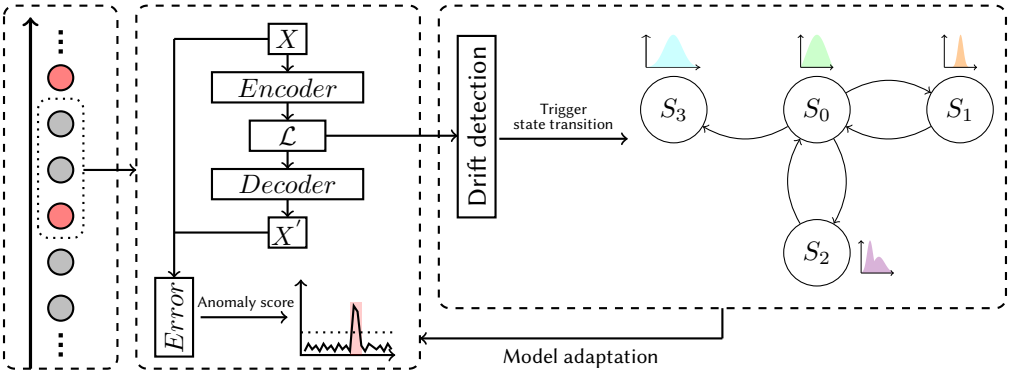

Figure 1: STAD overview: The left block is a multivariate data stream, where red dots denote abnormal data points and the dashed box is a data snippet. The middle block is an conventional autoencoder-based anomaly detection module, which detects abnormal snippets from the data stream. The right block takes latent representations from the autoencoder and conducts concept drift detection, which consequently triggers state transition and model adaptation.

3. State-transition-aware anomaly detection

In this section, we propose STAD, a state-transition-aware anomaly detection model, which employs autoencoders as the base model. The latent representations of autoencoders are used to detect concept drifts, which consequently trigger state transitions. An overview of STAD is shown in Figure 1.

3.1. Reconstruction and latent representation learning

Let $f_{Enc}\colon \mathbb{R}^D \to \mathbb{R}^H$ and $f_{Dec}\colon \mathbb{R}^H \to \mathbb{R}^D$ be the encoder and decoder of an autoencoder. The encoder maps a snippet X_t^{t+w} of the multivariate streaming data into a H-dimensional latent representation $L \in \mathbb{R}^H$, while the decoder reconstructs the same format snippet $X_t'^{t+w}$ from L, where w is the snippet length and $t, w \in \mathbb{N}$. A common assumption for anomaly detection using autoencoders is that pure normal data are available for the initial model training. The reconstruction error $e_t^{t+w} = |X_t^{t+w} - X_t'^{t+w}|$ indicates the goodness of fit to the normal data. In the test phase, abnormal snippets will cause larger reconstruction errors than normal data such that they are separable. The encoder and decoder can be implemented with a variety of deep models [10, 11]. Considering the temporal dependencies in streaming data, Recurrent Neural Networks (RNNs) and their variants [3, 4] are naturally suitable for the target. In the following illustration, as an example, we take the LSTM-Autoencoder [4], which takes data snippets as input and produces a single latent representation for each snippet. To map the multivariate reconstruction error to the likelihood of anomalies, a commonly used approach is to estimate a multivariate Gaussian distribution from the reconstruction error of normal data and measure the Mahalanobis distance between the reconstruction error of an unknown data point to the estimated distribution [4]. Moreover, the Gaussian Mixture Model (GMM) [10] and energy-based model [11] can also be used for likelihood estimation. The thresholding over the estimated anomaly likelihood in an unsupervised manner is challenging, especially in the

Algorithm 1 Concept Drift Detection

Input: Stack $\mathcal{L}_{hist}$ with minimum size m, queue $\mathcal{L}_{new}$ with size n, current state $S = \langle P, AE \rangle$, state transition model $\mathcal{M} = \langle \mathcal{X}, \mathcal{S}, \delta \rangle$

while stream does not end **do**
  $L_t \leftarrow$ AnomalyDetection(AE,X_t^{t+w}) ▷ Get latent representation
  $\mathcal{L}_{new} \leftarrow \mathcal{L}_{new} \cup L_t$
  if $\mathcal{L}_{new}.size > n$ **then** ▷ Move the oldest element of $\mathcal{L}_{new}$ to $\mathcal{L}_{hist}$
    $L_{t-n+1} = \mathcal{L}_{new}.pop()$
    $\mathcal{L}_{hist} \leftarrow \mathcal{L}_{hist} \cup L_{t-n+1}$
  end if
  if $\mathcal{L}_{hist}.size \geq m$ **then**
    for $h = 0, .., H-1$ **do** ▷ Dimension-wise test
      if KSTest($\mathcal{L}_{hist}^h$,$\mathcal{L}_{new}^h$) is True **then** ▷ Equation 1
        $S \leftarrow$ StateTransition(S, $\mathcal{L}_{new}$, $\mathcal{S}$, δ)
        Report concept drift, clear $\mathcal{L}_{hist}$ and $\mathcal{L}_{new}$
        break
      end if
    end for
  end if
end while

real-time prediction scenario. A possible non-parametric dynamic thresholding technique is proposed in [12]. The unsupervised approach for the adaptive threshold in different periods is not the main focus of this paper and will be addressed in our future work. In the following sections, we focus on adapting autoencoders based on the state transitions.

3.2. Drift detection in the latent space

In real-time, the latent representations of the autoencoder are accumulated for concept drift detection. Existing concept drift detection approaches mostly work in the original space, targeting linear separable concept drifts. Considering the complex concept drifts in multivariate streaming data, even non-linear distributional changes can be observed in the autoencoder latent space. We perform dimension-wise two-sample Kolmogorov–Smirnov test (KS-test) [13, 14] as a non-parametric and distribution-free statistical test to check whether two latent representations are drawn from the same continuous distribution. Algorithm 1 shows the online concept drift detection process. Formally, let $\mathcal{L}_{hist} = \{L_{t-m-n+1}, L_{t-m-n+2}, ..., L_{t-n}\}$ be the accumulated latent representation since the last concept drift and $\mathcal{L}_{new} = \{L_{t-n+1}, L_{t-n+2}, ..., L_t\}$ be the latest latent representations. F_{hist} and F_{new} are the empirical estimated cumulative distribution functions from the two latent representation sets. The null hypothesis (i.e., the observations in $\mathcal{L}_{hist}$ and $\mathcal{L}_{new}$ are from the same distribution) will be rejected if

$$\sup_L |F_{hist}(L) - F_{new}(L)| > c(\alpha)\sqrt{\frac{m+n}{m \cdot n}} \tag{1}$$

where sup is the supremum function, α is the significance level, m, n are the size of $\mathcal{L}_{hist}$ and $\mathcal{L}_{new}$, $c(\alpha) = \sqrt{-\ln(\frac{\alpha}{2}) \cdot \frac{1}{2}}$. Since the KS-test is designed for univariate data, we conduct parallel tests in each latent dimension and report concept drift if the null hypothesis is rejected in at least one of the dimensions. Once a concept drift is detected, the historical and latest sample sets are emptied and we further collect samples from the new data distribution.

3.3. State transition model

Modeling reoccurring data distributions (e.g., seasonal changes), coupling autoencoders with drift detection, and reusing models based on the distributional features can increase the efficiency of updating a deep model in real-time. In STAD, for each period between two concept drifts in the data stream, the data distribution, as well as the corresponding autoencoder are represented in a fixed-length queue $\mathcal{S}$. The first state $S_0 \in \mathcal{S}$ represents the beginning period of the data stream before the first concept drift. After every new concept drift, a new autoencoder will be trained from scratch if no existing element in the queue fits the current data distribution; otherwise, the state will transit to the existing one and reuse the corresponding autoencoder. In our study, we assume that sufficient data after the concept drifts can be accumulated to initialize a new autoencoder. In future work, we plan to discover state transitions with limited data (e.g., tolerantly reusing existing autoencoders).

To compare the distributional similarity between the newly arrived latent representation Q and the distributions of existing states $\{P_i | i = 1, ..., N\}$, we employ the symmetrized Kullback–Leibler Divergence. The similarity between Q and an existing state distribution P_i is defined as

$$\begin{aligned} D_{KL}(P_i, Q) &= D_{KL}(P_i||Q) + D_{KL}(Q||P_i) \\ &= \sum_{L \in \mathcal{L}} P_i(L) log \frac{P_i(L)}{Q(L)} + Q(L) log \frac{Q(L)}{P_i(L)} \end{aligned} \tag{2}$$

The next step is to estimate the corresponding probability distributions from the sequence of latent representations. In [14, 13], the probability distribution of categorical data is estimated by the number of object appearances in each category. In our case, the target is to estimate the probability distribution of fixed length real-valued latent representations. In previous research, one possibility for density estimation of streaming data is to maintain histograms of the raw data stream [15]. In STAD, we take advantage of the fix-sized latent representation of autoencoders and maintain histograms of each period in the latent space for the density estimation. Let $\mathcal{L} = \{L_1, L_2, ..., L_t\}$ be a sequence of observed latent representations, where $L_i = \langle h_1^i, h_2^i, ..., h_H^i \rangle$ and H is the latent space size, the histogram of $\mathcal{L}$ is

$$g(k) = \frac{1}{t} \sum_{L_i \in \mathcal{L}} \frac{e^{h_k^i}}{\sum_{j=1}^{H} e^{h_j^i}} \quad (k = 1...H) \tag{3}$$

and the density of a given period is estimated by $P(k) = g(k)$. Hence, Equation 2 can be converted to

$$D_{KL}(P_i, Q) = \sum_{k=1...H} P_i(k) log \frac{P_i(k)}{Q(k)} + Q(k) log \frac{Q(k)}{P_i(k)} \tag{4}$$

Algorithm 2 State Transition Procedure

function StateTransition(S_{hist}, $\mathcal{L}_{new}$, $\mathcal{S}$, δ)
  $P_{new} = \text{DensityEstimation}(\mathcal{L}_{new})$
  if $\min_{S_i=\langle P_i, AE_i\rangle \in \mathcal{S}} \{D_{KL}(P_{new}, P_i)\} \leq \epsilon$ **then** ▷ Equation 4
    $\delta \leftarrow \delta \cup (S_{hist} \Rightarrow S_{min})$
    return S_{min}
  end if
  $S_{new} \leftarrow \langle P_{new}, AE_{new}\rangle$ ▷ AE_{new}: Trained on new concept data
  $\mathcal{S} \leftarrow \mathcal{S} \cup S_{new}$
  $\delta \leftarrow \delta \cup (S_{hist} \Rightarrow S_{new})$
  if $\mathcal{S}.size > N$ **then**
    Remove the oldest state and relevant transitions
  end if
  return S_{new}
end function

For a newly detected concept with distribution Q, if there exists a state $S_i (i \in [1, N])$ with corresponding probability distribution P_i satisfies $D_{KL}(P_i, Q) \leq \epsilon$, where ϵ is a tolerant factor, and S_i is not the direct last state, the concept drift can be treated as a reoccurrance of the existing concept, therefore the corresponding autoencoder can be reused, and the state transfers to the existing state. If no autoencoder is reusable, a new one will be trained on the latest arrived data after concept drift. To prevent an explosion in the number of states, the state transition model $\mathcal{M} = \langle \mathcal{X}, \mathcal{S}, \delta \rangle$ only maintains the N latest states. The state transition procedure is described in Algorithm 2.

4. Case study

We carry out a case study using the Covid-19 daily infection case dataset of Germany[2], where the waves of the epidemic can be considered as human-interpretable concept drifts and the public holidays with abnormal statistic numbers are the anomalies. The Covid dataset (Figure 2) contains daily new infection cases and death cases in Germany from March 2020 to April 2021. The data stream follows a 7-day period and fluctuates with the trend depending on the development of the epidemic, seasons, and local prevention policies. The LSTM-autoencoder and scoring function in [4] is applied as the base model. Both the encoder and decoder consist of a single LSTM unit, and the latent representation is three dimensional. We use the data between March and May 2020 to initialize the autoencoder and let the rest data arrive in a streaming fashion. The model takes sliding windows (snippets) of 7 timestamps length (a week) without overlap as inputs for the autoencoder. For both initial and real-time training, the autoencoders are trained with 50 epochs with 0.4 dropout rate. For the KS-test, $m = 3$, $n = 2$, and the significance level is set to $\alpha = 0.05$. The real-time processing starts from June 2020. The dashed lines in Figure 2 are the positions where concept drifts are detected in the latent space. All four

[2] https://www.arcgis.com/home/item.html?id=f10774f1c63e40168479a1feb6c7ca74

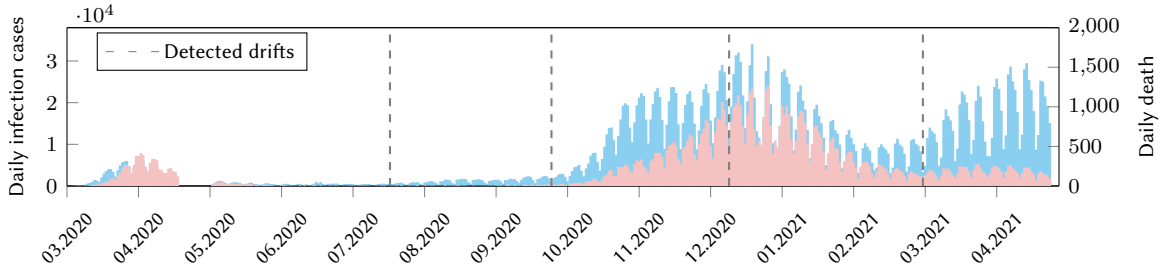

Figure 2: Covid-19 daily infection (blue) and death (pink) cases in Germany

detected drifts are near significant changes in the evolution of the epidemic. The threshold ϵ of KL-divergence is set to 0.0025. The size of the new buffer $\mathcal{L}_{new}$ is 14. As shown in Figure 3, no reusable autoencoder is found for the first three concept drifts such that three new states with corresponding new autoencoders are created. After the concept drift near March 2021, the upcoming data in $\mathcal{L}_{new}$ has KL-divergence below ϵ with state S_2 (end September to early December), therefore it triggers a backward state transition to S_2.

In the test phase, we manually labeled 11 weeks containing public holidays in Germany as abnormal snapshots and ranked the anomaly scores in the periods corresponding to each state. In the evaluation of recall in the ranking list, we got 18% for $R@1$, 54% for $R@5$ and 90% for $R@10$. A major reason is that some data points from the beginning of concept drifts are mistakenly alarmed as anomalies before the model update. In the follow-up work, we aim to reduce the false positive detection of anomalies by distinguishing concept drifts and abnormal snapshots by their length.

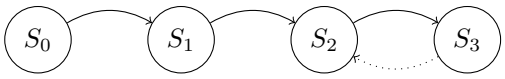

Figure 3: State transition of Covid-19 data stream (Dotted arrow: reusing state)

5. Conclusion

We have proposed an autoencoder-based streaming data anomaly detection approach STAD, which uses the latent representation to detect concept drift and model state transitions between different data distributions in the data stream. With a demo experiment, we showed the state-transition-aware anomaly detection process during the stream evolution. However, there are still open challenges. In the current work, we assume that sufficient data are available for online training. However, the states of some periods in the real data stream are too short, such that the data for training a new model in real-time is not sufficient. One future work is to discover efficient strategies for reusing autoencoders for such cases. Another further research direction is to discover semantic explanations for each state, which helps the human better understand the model as well as the changes of data.

References

[1] J. Sipple, Interpretable, multidimensional, multimodal anomaly detection with negative sampling for detection of device failure, in: International Conference on Machine Learning, PMLR, 2020, pp. 9016–9025.

[2] M. Sakurada, T. Yairi, Anomaly detection using autoencoders with nonlinear dimensionality reduction, in: Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis, 2014, pp. 4–11.

[3] E. Marchi, F. Vesperini, F. Weninger, F. Eyben, S. Squartini, B. Schuller, Non-linear prediction with lstm recurrent neural networks for acoustic novelty detection, in: 2015 International Joint Conference on Neural Networks (IJCNN), IEEE, 2015, pp. 1–7.

[4] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, G. Shroff, Lstm-based encoder-decoder for multi-sensor anomaly detection, arXiv preprint arXiv:1607.00148 (2016).

[5] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, G. Zhang, Learning under concept drift: A review, IEEE Transactions on Knowledge and Data Engineering 31 (2018) 2346–2363.

[6] J. Gama, P. Medas, G. Castillo, P. Rodrigues, Learning with drift detection, in: Brazilian symposium on artificial intelligence, Springer, 2004, pp. 286–295.

[7] M. Baena-Garcıa, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavalda, R. Morales-Bueno, Early drift detection method, in: Fourth international workshop on knowledge discovery from data streams, volume 6, 2006, pp. 77–86.

[8] M. U. Togbe, Y. Chabchoub, A. Boly, M. Barry, R. Chiky, M. Bahri, Anomalies detection using isolation in concept-drifting data streams, Computers 10 (2021) 13.

[9] M. Ceci, R. Corizzo, N. Japkowicz, P. Mignone, G. Pio, Echad: embedding-based change detection from multivariate time series in smart grids, IEEE Access 8 (2020) 156053–156066.

[10] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, H. Chen, Deep autoencoding gaussian mixture model for unsupervised anomaly detection, in: International Conference on Learning Representations, 2018.

[11] S. Zhai, Y. Cheng, W. Lu, Z. Zhang, Deep structured energy based models for anomaly detection, in: International Conference on Machine Learning, PMLR, 2016, pp. 1100–1109.

[12] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, T. Soderstrom, Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding, in: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, 2018, pp. 387–395.

[13] C. Chen, Y. Wang, J. Zhang, Y. Xiang, W. Zhou, G. Min, Statistical features-based real-time detection of drifted twitter spam, IEEE Transactions on Information Forensics and Security 12 (2016) 914–925.

[14] T. Dasu, S. Krishnan, S. Venkatasubramanian, K. Yi, An information-theoretic approach to detecting changes in multi-dimensional data streams, in: In Proc. Symp. on the Interface of Statistics, Computing Science, and Applications, Citeseer, 2006.

[15] R. Sebastiao, J. Gama, Change detection in learning histograms from data streams, in: Portuguese Conference on Artificial Intelligence, Springer, 2007, pp. 112–123.