IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 6 July 2022, accepted 28 September 2022, date of publication 10 October 2022, date of current version 17 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3213038

RESEARCH ARTICLE

Improving Deep Learning Based Anomaly Detection on Multivariate Time Series Through Separated Anomaly Scoring

ADAM LUNDSTRÖM[1,2], MATTIAS O'NILS[1], FAISAL Z. QURESHI[1,3], (Senior Member, IEEE), AND AXEL JANTSCH[1,4], (Senior Member, IEEE)

[1]Department of Electronics Design, Mid Sweden University, 85170 Sundsvall, Sweden
[2]SCA, 85188, Sundsvall, Sweden
[3]Faculty of Science, University of Ontario Institute of Technology, Oshawa, ON L1G 0C5, Canada
[4]Institute of Computer Technology, TU Wien, 1040 Vienna, Austria

Corresponding author: Adam Lundström (adam.lundstrom@miun.se)

This work was supported in part by The Knowledge Foundation (kks.se) within the Industrial Graduate School Smart Industry Sweden.

ABSTRACT The importance of anomaly detection in multivariate time series has led to the development of several prominent deep learning solutions. As a part of the anomaly detection process, the scoring method has shown to be of significant importance when separating non-anomalous points from anomalous ones. At this time, most of the solutions utilize an aggregated score which means that relevant information created by the anomaly detection model might be lost. Therefore, this study has set out to examine to what extent anomaly detection in multivariate time series based on deep learning can be improved if all the residuals from each individual channel is considered in the anomaly score. To achieve this, an aggregated and separated scoring method has been applied with a simple denoising convolutional autoencoder. In addition, the performance has been compared with other state-of-the-art methods. The result showed that the separated approach has the potential to generate a significantly higher performance than the aggregated one. At the same time, there were some indications suggesting that an aggregated scoring is better at generalizing when no labels are available to select the anomaly thresholds. Therefore, the result should serve as an encouragement to use a separated scoring approach together with a small sample of labeled anomalies to optimize the thresholds. Lastly, due to the impact of the anomaly score, the result suggests that future research within this field should consider applying the same anomaly scoring method when comparing the performance of deep learning algorithms.

INDEX TERMS Anomaly detection, anomaly scoring, deep learning, multivariate time series (MVTS).

I. INTRODUCTION

Finding anomalies in time series can be of great value for a number of different applications within the smart industry including, manufacturing, maintenance, security and server machines [1]. This can be especially true for multivariate data that has both temporal and interrelations dependencies [1]. Because of the lack of real labeled data as well as class imbalances and data heterogeneity, unsupervised and semi-supervised learning for anomaly detection has been researched extensively to perform accurate anomaly detection using few or no labels [1], [2]. Several different types of methods have been developed for multivariate anomaly detection in time series data including statistical based methods such as [3] and [4] but also traditional machine learning methods such as [5]. In recent times however, the use of deep learning has become a prominent way of finding anomalies in multivariate times series data that has shown great success on benchmark datasets. The different types of architectures used include autoencoders, Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), Generative Adversarial Networks (GAN) as well as graph

The associate editor coordinating the review of this manuscript and approving it for publication was Frederico Guimarães.

networks [1]. In addition, several different scoring methods have been developed. A scoring method defines, based on the output of the developed model, what should and should not be regarded as an anomaly. In a recent study it was shown that the scoring method has a significant impact on the overall performance that can surpass the importance of algorithm selection [6]. Currently, most state-of-the art methods employ an aggregation score [6], meaning that the residuals for each of the individual channels are aggregated, when determining the anomaly probability. Yet, this means that relevant information about the anomalous behaviour disappears and can, as was shown in [7], mean that anomalies caused by a single anomalous signal in a multivariate space, are missed. Because of the loss of valuable information when reducing the model output, it is arguably of interest to examine the behaviour of individual signals separately. The main research question is therefore:

- To what extent can the performance of multivariate anomaly detection on time series data based on deep learning be improved by constructing an anomaly score considering the residuals of each individual channel separately?

To answer this question, a convolutional autoencoder is used together with two different scoring methods: one with the suggested separated scoring and one with aggregated scoring based on previous work. These are evaluated on benchmark datasets and compared with other state-of-the-art methods. The overall framework for the suggested separated approach is illustrated in Figure 1, which consists of three different layers with different multivariate data inputs, $\mathcal{D}$: two offline layers with training and optimization and one online layer that handles continues data streams. The first layer trains a model in an unsupervised fashion and the second layer constructs separated thresholds based on a channel-wise residuals and labels. Then, these thresholds are applied to the channel-wise residuals to produce the channel-wise score. The contributions of this study are that it:

- Constructs a separated scoring method for multivariate anomaly detection on time series data based on deep learning algorithms.
- Evaluates to what extent a separate scoring method for multivariate anomaly detection on time series data based on deep learning algorithms can outperform an aggregated scoring method.

II. RELATED WORK

In previous work, several different types of methods for anomaly detection in multivariate time series data have been used, both with regard to the deep learning algorithm, and the scoring method. Recent work by J. Audibert et al. [8] has successfully shown the advantages of using adversarial training. In their method, called USAD, two different autoencoders with a shared encoder are trained with a separated optimization goal. The optimization goal of the first autoencoder is to create a representation of the input that is as accurate as possible and the goal of the second one is to create a representation that is as inaccurate as possible. Their anomaly score is based on the weighted sum of the output of the respective decoders and using a single anomaly threshold to separate anomalous from non-anomalous points. Furthermore, in a study by D. Li et al. [9], MAD-GAN was introduced which is a GAN that uses a generator and a discriminator consisting of two LSTM networks. The anomaly score in their method is based on an aggregation considering both the reconstruction of the input data as well as the result from the discriminator.

Another used is Variational Autoencoder (VAE), which in contrast to standard autoencoders encodes the Gaussian distributions. Both LSTM-VAE described by D. Park, Y. Hoshi, and C. Kem in [10] and OmniAnomaly by Y. Su et al. [11] employs recurrent-based VAE. In OmniAnomaly, Gated Recurrent Unit (GRU) cells and dense layers are used together with a planar normalization flow that defines the latent space of the input, which is then decoded. They calculate the anomaly score at point t as an aggregation based on the contribution of each channel by considering the reconstruction probability and then use a single threshold to separate anomalous and non-anomalous points. Similarly, LSTM-VAE is based on RNN, but employs long-short-term-memory (LSTM) cells instead of GRU. They also use probability-based score based on the negative log of the encoder-decoder structure together with a single threshold.

Similarly to the VAE, B. Zong et al. [12] have proposed a method called Deep Autoencoder Gaussian Mixture Model (DAGMM) that bases the anomaly detection on the estimated likelihood of a particular point in time belonging to a learned Gaussian distribution. The method uses two different networks consisting of a standard dense autoecoder and an estimation network that uses the compressed representation from the autoencoder and learns the Gaussian mixture. The anomaly score is based on the likelihood of membership to the trained Gaussian mixture and an anomaly is defined if the score surpasses a predefined threshold.

Furthermore, convolutional neural networks has extensively been applied. In a study by [13], a convolutional autoencoder together with a GRU neural network was applied. They use an aggregated scoring method based on the eigenvalues of each channel. In addition, a method called Multi-Scale Convolutional Recurrent Encoder-Decoder (MSCRED) was introduced by C. Zhang et al. [14] where a combination of recurrent and convolutional autoencoder is utilised. The network uses convolutions to compress the data and skip-connection with convolutional LSTM cells. In addition, they convert the multivariate time series to correlation matrices based on different lengths as input, which is unique compared to the other methods presented in this paper. The network is then used to recreate the input matrices and a particular point in time is defined as anomalous if the count of anomalous cells, which are cells where the difference

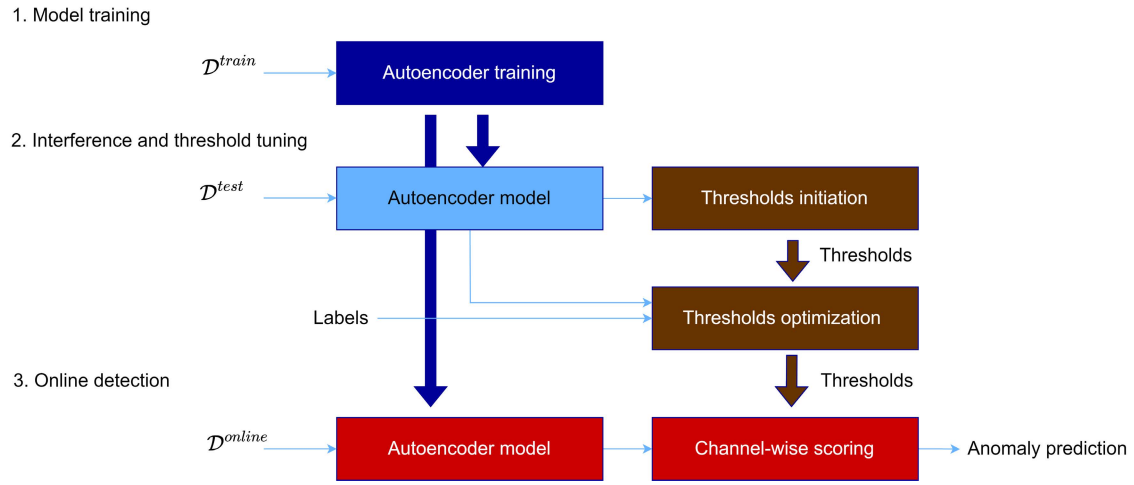

FIGURE 1. Framework for the anomaly detection method.

between the input matrices and reconstructed output matrices is higher than a thresholds, is higher than a given aggregation threshold.

Graph neural networks have also shown great results in recent works. Firstly, A. Deng and B. Hooi present GDN [15], which uses an attention-based prediction approach, meaning that the future relation is estimated and compared to the actual future data. Their anomaly score is defined as the maximum residual from the robustly standardized (based on the inter-quartile range) individual residuals for each sensor. This means that they use a single threshold to separate anomalous from not-anomalous points. Furthermore, GTA was introduced by Z. Chen et al. [16], which uses multi-scale dilated convolutions and incorporates multi-head attention in the learning process. They apply a simple aggregation of the residuals for each channel and define a point in time anomalous if the score surpasses a certain threshold.

L. Shen, Z. Li and J. T. Kwok present a recurrent based temporal hierarchy one-class network (THOC) [17]. The method uses dilated RNN with multi-scale skip connection where the data is scaled into a Multiscale Support Vector Data Description (MSVDD) that is used to define the anomaly score at a certain point in time. A point is anomalous if the score is above a predefined threshold.

Regarding anomalies scoring methods [6] explored different types of aggregated based methods that showed great success when deployed together with state-of-the-art algorithms such as OmniAnomaly and USAD. They recommend using the method called Gauss-D where the scoring is aggregated as the cumulative negative log of the Gaussian probability distribution of each of the channels. Furthermore, they showed that the scoring method has a significant impact on the overall performance of the anomaly detection method.

While the result from these different studies have shown great promise, none of them apply a separate anomaly scoring approach, and considering the loss of information that an aggregated score inevitably leads to, it seems to be of great value to examine to what extent it affects the anomaly detection performance.

In addition to deep learning methods, statistical and traditional machine learning methods have, in a recent study showed promising result as alternatives to deep learning methods [18]. The authors argue with their compelling result that benchmarks for anomaly detection on multivariate time series, to a greater extent, should consider statistical and traditional machine learning approaches. Some of the statistical methods that showed promising results were Principal Component Analyses (PCA) [19] and Independent Component Analyses (ICA) [20] which can be applied as reconstruction based methods for anomaly detection were the input data is encoded and reconstructed to compare with the original input to find anomalies [21]. Regarding, traditional machine learning solutions, One-Class Support Vector Machine (OCSVM) [22], Isolation Forest (IF) [23] as well as Local Outlier Factor (LOF) [24], which uses different types of separation techniques for finding anomalies, showed performance that could be comparable with those by the considered deep learning methods.

III. METHOD

A. PROBLEM FORMULATION

We are provided two sets of time series data $\mathcal{D}^{\text{train}}$ and $\mathcal{D}^{\text{test}}$. $\mathcal{D}^{\text{train}} = \{\mathbf{x}_0, \cdots, \mathbf{x}_n\}$, where $\mathbf{x}_t \in \mathbb{R}^C$ is the C-channel reading at time $t \in [0, n]$. We assume that $\mathcal{D}^{\text{train}}$ does not include any anomalous data. $\mathcal{D}^{\text{test}} = \{(\mathbf{x}_0, y_0), \cdots, (\mathbf{x}_m, y_m)\}$, where $\mathbf{x}_t \in \mathbb{R}^C$ is the C-channel reading at time $t \in [0, m]$ and $y_t \in [0, 1]$ labels the corresponding $\mathbf{x}_t$ as either not anomalous or anomalous. The goal is to train a model $\mathcal{M}$ on $\mathcal{D}^{\text{train}}$ and use this model to predict if a particular $\mathbf{x_t} \in \mathcal{D}^{\text{test}}$ is anomalous.

We assume that model has access to the l prior readings when classifying the reading at the current time step t as anomalous or not with the exception of the first l readings

where all l reading are assumed to be accessible. Let $\mathbf{X}_t \in \mathbb{R}^{l\times C}$ collects the last l readings row-wise. We will use labels train and test to indicate which of the two sets $\mathbf{X}_t$ refers too.

B. METHOD OVERVIEW

In Figure 2, the method overview is described. It includes two basic steps which are 1) training and 2) inference and threshold tuning. In training, an autoencoder is trained which is then used to generate the channel-wise residual for each time step in the test data. Then, these residuals together with a threshold optimization procedure are used to produce the anomaly prediction based on separated anomaly scoring. The optimization is based on labels provided for the test dataset and the scoring is constructed by normalizing the channel-wise residuals using their respective threshold and setting the score to the normalized channel-wise residual with the highest value.

C. PART 1: TRAINING

1) MODEL TRAINING

We are interested in capturing the variations within each channel. This is the first step towards identifying which values fall outside the ''usual'' or acceptable ranges. We employ a Denoising Convolutional Autoencoder (DCAE), Figure 3, to capture data characteristics. Convolutional autoencoders have in different forms successfully been applied in the time series data domains and shown great performance as shown in for example [14] and [13]. There are limitations to the described DCAE such as the usages of kernels with a fixed size meaning that it can be challenging for it to capture long-term anomalies. However, we found that the performance of the DCAE is sufficient in order to answer our research question, which is to evaluate to what extent the performance of multivariate anomaly detection on time series data based on deep learning can be improved by constructing an anomaly score considering the residuals of each individual channel separately. The DCAE learns to reconstruct $\mathbf{X}_t^{\text{train}}$ by minimizing the residual

$$|\mathbf{X}_t^{\text{train}} - \hat{\mathbf{X}}_t^{\text{train}}|_2, \tag{1}$$

where $\mathbf{X}_t^{\text{train}}$ and $\hat{\mathbf{X}}_t^{\text{train}}$ refers the input and its reconstruction, respectively. The denoising aspect of the network is to prevent it from over-fitting by inducing Gaussian noise with the purpose of forcing it to learn the most significant features. We selected a fixed kernel size of seven and the time series is compressed using strides of two. When the network has been trained, the model can be tested on new data where a residual above a certain threshold deems the point X_t to be anomalous.

2) CHANNEL-WISE RESIDUALS COMPUTATION

Given the input $\mathbf{X}_t^{\text{train}}$ and its reconstruction, we construct an residual measure as follows:

$$\mathbf{E}_t = |\mathbf{X}_t^{\text{train}} - \hat{\mathbf{X}}_t^{\text{train}}|_1. \tag{2}$$

Note that E_t is defined over time window l, whereas we are interested in computing an residual measure $\mathbf{e}_t$ for the reading $\mathbf{x}_t$ at time t. We compute $\mathbf{e}_t$ by averaging $\mathbf{E}_t$ over a time window l_e as follows:

$$\mathbf{e}_t = \mathbf{k}^T \mathbf{E}_t. \tag{3}$$

$\mathbf{k}$ is the smoothing kernel defined as follows:

$$\mathbf{k}^T = \frac{1}{l_e}\left[\mathbf{0}^T \mid \mathbf{1}^T\right]. \tag{4}$$

Here $\mathbf{1} \in \mathbb{R}^{l_e}$ is a vector of ones and $\mathbf{0} \in \mathbb{R}^{l-l_e}$ is a vector of zeros. l_e represents kernel-width, and we have set it equal to 10.

3) SCORING FUNCTIONS

We will use Gauss-D approach proposed in [6] as the baseline and show that our method that computes channel-wise threshold improves upon the results achieved by the baseline. For the remainder of this discussion, we will refer to the Gauss-D approach as DCAE_a and our method as DCAE_s.

4) GAUSS-D (DCAE_a)

We fit a Gaussian distribution on channel-wise residual computed over a time window up to the current time t:

$$\mu^i = \frac{1}{w}\sum_{j=0}^{w-1} e_{t-j}^i; \quad (\sigma^i)^2 = \frac{1}{w-1}\sum_{j=0}^{w-1}\left(e_{t-j}^i - \mu^i\right)^2, \tag{5}$$

$$A_t^i = -\log\left(1 - \Phi\left(\frac{\mathbf{e}_t^i - \mu^i}{\sigma^i}\right)\right), \tag{6}$$

where Φ is cdf of $N(0, 1)$. The aggregate score is then $a_t = \sum_i A_t^i$. An anomaly is defined if the score a_t surpass a single threshold.

5) SEPARATED APPROACH

The separated approach suggested provides a score based on the evaluation of the residuals from each of the channels. Considering the thresholds $\boldsymbol{\tau} = (\tau^1, \tau^2, \ldots, \tau^C)$ the point x_t is determined to be anomalous if any e_t^i is above τ^i for $i = 1, \ldots, C$. Yet, as all $\tau \in \boldsymbol{\tau}$ are set separately, they differ in magnitude, which implies that it is not possible to present a simple value to the user that describes the anomaly score for a point x_t. Therefore, e^i and τ^i are scaled with the same scalar so that all $\tau \in \boldsymbol{\tau}$ are normalized to 0.5 meaning that each threshold is set to a shared value but the relation between e_t^i and τ^i is kept:

$$e_t = e_t \circ (0.5 \oslash \boldsymbol{\tau}). \tag{7}$$

Here $\mathbf{0.5} \in \mathbb{R}^C$ is a vector of 0.5 and $\circ$ and $\oslash$ denotes the Hadamard product and division, respectively. When this has been completed, it is possible to define the anomaly score a_t for x_t as the max scaled e_t:

$$a_t = \max(e_t). \tag{8}$$

Then, the label y_t, is defined as follows:

$$y_t = \begin{cases} 0 & \text{if } a_t <= 0.5 \\ 1 & \text{else.} \end{cases} \tag{9}$$

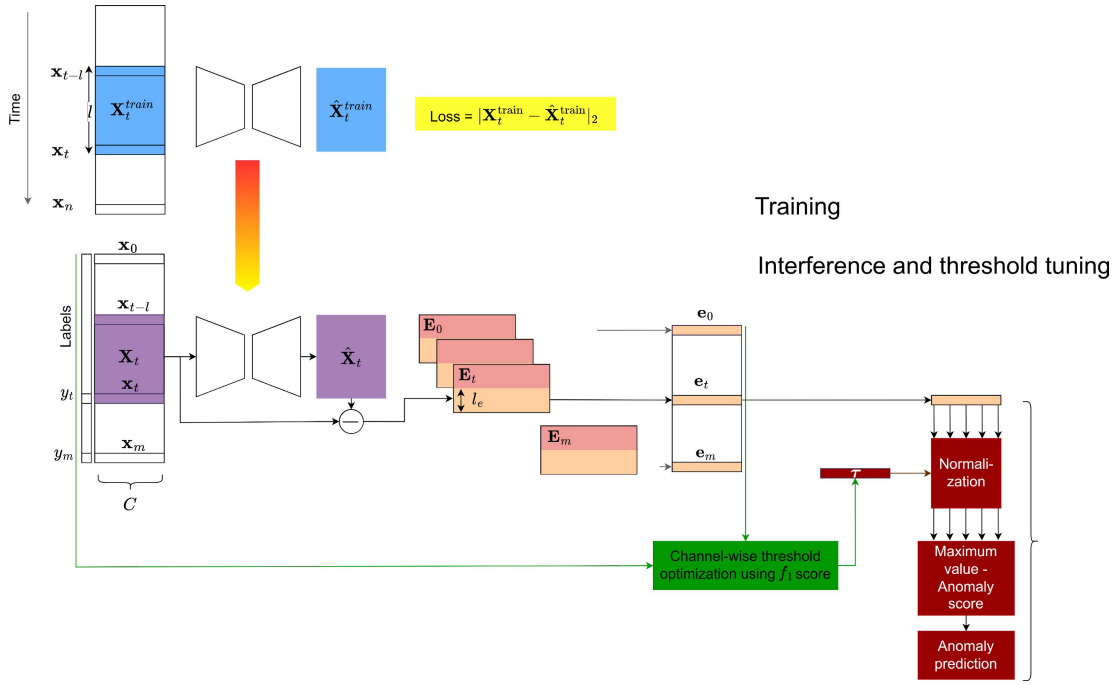

FIGURE 2. Method overview. A model is trained using non-anomalous training data and is then used to generate channel-wise residuals. Together with labels and a optimization procedure a combined score can be produced considering each individual channel.

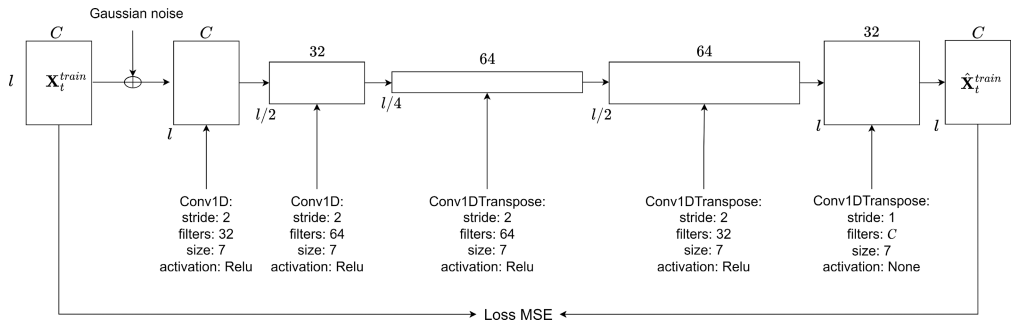

FIGURE 3. The architecture of the autoencoder. It uses strides of size two and kernels of size seven to encode the data and transpose with the same kernel size to decode the data.

D. PART 2: INFERENCE AND FINE-TUNING

Inference involves deciding whether or not a reading at time t is anomalous or not given the last $w-1$ readings.[1] We use the following procedure to decide whether or not a reading $\mathbf{x}_t \in \mathcal{D}^{\text{test}}$ is anomalous. First, construct $\mathbf{X}_t \subset \mathcal{D}^{\text{test}}$ as discussed previously. Recall that $\mathbf{X}_t \in \mathbb{R}^{l \times C}$. Use the autoencoder to reconstruct $\mathbf{X}_t$ and follow the steps discussed above to compute $\mathbf{e}_t$, the residual measure for the reading in question. Next, given $\{\mathbf{e}_{t-(w-1)}, \mathbf{e}_{t-(w-2)}, \cdots, \mathbf{e}_t\}$, compute μ^i and σ^i needed to compute the scores using Gauss-D approach as described above.

1) UPDATING THRESHOLDS

In this paper the thresholds which gives the highest f_1 score have been used to evaluate the different methods. This is a procedure that has been used by previous research to benchmark a method against other approaches, including [8], [16] and [13]. For Gauss-D, the threshold was selected by simple testing the score for different values in a reasonable range and choosing the best one. For the separated approach, an optimization procedure was applied described in the next section.

2) OPTIMIZATION OF SEPARATE THRESHOLDS

Before the optimization begins, the initial thresholds $\boldsymbol{\tau}_{init}$ are set to the maximum residual on the training data for each respective channel. Then, the optimization procedure is applied, which consists of two steps. In both of these, the optimization has been carried out in a greedy fashion where changes to each individual threshold has been made iteratively until a condition has been met. In the first step,

[1] Use training data when last w test readings are not available.

the goal is to maximise the f_1 score with the optimized thresholds $\boldsymbol{\tau}_{optimized}$. This is achieved by optimizing $\boldsymbol{\tau}_{init}$ based on the true labels, L from the test dataset, and the residual for each of the separate channels, e, according to Equation (10). In the experiment, this was achieved by first decrementing the thresholds until the f_1 score stopped improving or the threshold became less than zero. Then, the thresholds were incremented until the f_1 score had not improved for a certain number of iterations or an overall iteration limit was reached. The changes to each threshold was calculated based on the highest loss for each channel.

$$\begin{aligned} \max_{\boldsymbol{\tau}} \quad & f_1(L, prediction(e, \boldsymbol{\tau})) \\ \text{s.t.} \quad & \tau^i \geq 0, \quad \tau^i \in \boldsymbol{\tau}, \quad i = 1, .., C \end{aligned} \tag{10}$$

When the problem defined by Equation (10) has been solved, some thresholds might have been significantly incremented in the search for a global maximum without affecting the overall f_1 score. This means that new anomalies might be missed because of too high anomaly thresholds. Therefore, we need to minimize the difference between the initial thresholds $\boldsymbol{\tau}_{init}$ and the new thresholds $\boldsymbol{\tau}$ without lowering the f_1 score based on $\boldsymbol{\tau}_{optimized}$, which is described in Equation (11).

$$\begin{aligned} \min_{\boldsymbol{\tau}} \quad & g(\boldsymbol{\tau}) = abs(\boldsymbol{\tau}_{init} - \boldsymbol{\tau}) \\ \text{s.t.} \quad & f_1(L, prediction(e, \boldsymbol{\tau}_{optimized})) \\ & = f_1(L, prediction(e, \boldsymbol{\tau})) \end{aligned} \tag{11}$$

IV. EXPERIMENTS

A. DATASETS

In this paper, as can be seen in Table 1, three different benchmark datasets have been used for evaluation. These are Water Distribution (WADI) [25], Secure Water Treatment (SWaT) [26] and Server Machine dataset (SMD) [11].

1) WADI

WADI consists of 123 different dimensions that describe a water distributing process [25]. The data was collected during 16 days where 14 were measured under normal conditions and 2 under attack scenarios. As there are different versions of the dataset, it is worth noting that the first version was used, as was done by other state-of-the-art methods such as [11] and [8]. The dataset can be retrieved via iTrust, Centre for Research in Cyber Security, Singapore University of Technology and Design [27]

2) SWaT

SWaT is a dataset that contains 51 dimensions describing a water treatment process [26], measured during 11 days, 7 days with normal operation and 4 days with different types of attacks [27]. As with WADI, there are different versions of the dataset available and similar to other methods such as [11] and [8], the first edition was used. The dataset can be retrieved via iTrust, Centre for Research in Cyber Security, Singapore University of Technology and Design [27]

3) SMD

SMD was presented by [11] and consists of 28 different entities divided between 3 different machines that all are described using 38 dimensions of data. It was collected from an internet company during 5 weeks and each of the entities have a training set as well as a test set which contains different types of labeled faults.

4) MSL

Mars Science Laboratory (MSL), provided by [28], is a dataset from a spacecraft on a mission launched by NASA. It consist in total of 27 different entities with 55 dimensions. The first dimension represents a telemetry channel and the rest are command signals that are one hot encoded.

5) SMAP

Soil Moisture Active Passive (SMAP) is also provided by [28] and is similar to MSL but describes another spacecraft and mission launched by NASA. It consists of 55 entities with 25 dimensions. Just like MSL, the first signal represents the telemetry value and the rest are one hot encoded commands.

TABLE 1. Description of datasets.

Dataset	Train set	Test set	Entities	Dimensions	Anomalies
WADI	1048571	172801	1	123	5.99%
SWaT	496800	449919	1	51	11.98%
SMD	708405	708420	28	38	4.16%
MSL	58317	73729	27	55	10.72%
SMAP	135183	427617	55	25	13.13%

B. ACCURACY METRICS

For the datasets mentioned above, two types of accuracy metrics are regularly used to compare the performance of different methods. These are point-wise score and point adjust score [8]. Point-wise score, in this paper denoted as f_{p1}, is simply based on the precision and recall for all of the predicted and true points. As this can give a low accuracy even though all different segments have been identified, point adjust, f_{pa1}, was suggested by [11]. The point adjust method determines all points within an anomalous segment as anomalous and treats all other points the same as for the point-wise metric. The main problem with this approach is that it will give a higher reward for identifying a long anomalous segment than a short one. Because of this issue [6] has developed a new metric that to a greater extent resolves the issue with these approaches. They define it as the composite f_1 score, f_{c1}, which is displayed in Equation (12). For this metric, recall is the overall segment recall R_e and the precision is the point-wise precision P_t.

$$f_{c_1} = 2 \cdot \frac{R_e \cdot P_t}{R_e \cdot P_t} \tag{12}$$

While the most appropriate metric could be different from case to case, in this paper we argue that f_{c1} is generally the most appropriate metric for anomaly detection in time series. All three metrics have been applied in this study. The point-wise and point adjust methods were used to compare our method to other methods described in previous work. In experiments comparing the aggregated scoring method to the separated scoring method, f_{c1} score have been applied.

C. SETUP

In the experiments, a similar setup used in previous work was used for a fair comparison, particularly [1], [8] and [6]. Firstly, both SWaT and WADI were down-sampled with a factor 3 to reduce the training time. Then, the data was normalized using a min-max scalar for each individual channel, shown in Equation (13). This was however not done to the datasets SMD, MSL and SMAP who already are provided normalized.

$$x_t^i = \frac{x_t^i - min(x^i)}{max(x^i) - min(x^i)} \tag{13}$$

Furthermore, to avoid extreme values and thus unnecessarily high residuals on the test data, similarly to [6], thresholds were set on the input data to cut it according to Equation (14). The purpose of doing this was to speed up the optimization procedure and values beyond these limits were shown to have insignificant effects on the overall performance in initial experiments.

$$f(x_t^i) = \begin{cases} -3 & \text{if } x_t^i < -3 \\ 4 & \text{if } x_t^i > 4 \\ x_t^i & \text{otherwise} \end{cases} \tag{14}$$

Also, for a better visualization of the result, a max limit for the separated anomaly score was set to 2.5, which does not affect the selection of anomalous points. In terms of hyperparameters for the DCAE, no major optimization efforts were conducted since the focus in this study has been the separated scoring method. This means that it is likely that higher accuracy can be achieved by optimizing hyperparameters such as kernel size to each dataset. The window size l was set to 128 for the SWaT data, as it seemed to give the highest accuracy and 32 for the other datasets to reduce training time. No major reduction in accuracy could be observed by reducing l for the other datasets. Furthermore, as was done in paper [6], the window size w in Gauss-D from Equation (5), was set to the same length as the training dataset. The batch size was set to 64, which generally seemed to give the best performance. For the training process the maximum number of epochs were set to 70 for WADI and SWAT and 100 for the rest. The deviation for the Gaussian noise was set considering the dataset size and performance to 0.2 for SWaT and WADI and 0.1 for SMD, MSL and SMAP. In addition, 20% was held out from the training data and used for early stopping for the training procedure. The deep learning methods that have been compared in the experiment are USAD [8], OmniAnomaly [11], LSTM-VAE [10], DAGMM [12], MAD-GAN [9], THOC [17], GTA [16] and GDN [15]. For each of the these, when possible, the result provided by the original paper was used. When this was not possible due to for example a different threshold setting method or lack of metrics presented, the result was retrieved from the experiment by [8] and [1].

In addition to the deep learning methods, the best performing statistical and traditional machine learning methods in the study by Audibert et al. [18] have also been used for comparison. These were implemented using the default setup provided by scikit-learn [29]. An exception was made for FastICA and PCA where we tested different number of components and selected the one related to the highest f_1 score. This was due to, in contrast to the other methods, high variations in accuracy depending on amount of components used. The methods selected were FastICA [21], [30] which is a version of ICA [20], PCA [19], [21], LOF [24], OCSVM [22], and IF [23]. These methods can be implemented so that a decision threshold applied to an anomaly score defines the anomaly prediction. To get the highest possible f_1 score we used a similar approach as in [18], which is to normalize the anomaly score between 0 and 1 and select the threshold with the highest f_1 score.

TABLE 2. Evaluation of datasets without point adjust.

Method	SWaT			WADI		
	P_t	R_t	f_{p1}	P_t	R_t	f_{p1}
USAD [8]	0.9851	0.6618	0.7917	0.9947	0.1318	0.2328
OmniAnomaly [11]	0.9825	0.6497	0.7822	0.9947	0.1298	0.2296
LSTM-VAE [10]	0.9897	0.6377	0.7756	0.9947	0.1282	0.2271
DAGMM [12]	0.4695	0.6659	0.5507	0.0651	0.9131	0.1216
MAD-GAN [9]	0.9897	0.6374	0.7700	0.4144	0.3392	0.3700
OCSVM [22]	0.9995	0.5853	0.7383	0.5536	0.2382	0.3331
LOF [24]	0.4529	0.3905	0.4193	0.1574	0.2207	0.1838
IF [23]	0.9985	0.5857	0.7383	0.2585	0.3423	0.2946
PCA [21]	0.9620	0.6027	0.7411	0.9299	0.2880	0.4398
ICA [21]	0.9620	0.6027	0.7411	0.9938	0.1441	0.2518
$DCAE_a$	0.8111	0.7478	0.7782	0.3441	0.5133	0.4120
$DCAE_s$	0.9594	0.8272	**0.8884**	0.9069	0.5697	**0.6998**

TABLE 3. Evaluation of datasets with point adjust.

Method	SWaT			WADI		
	P_{pa}	R_{pa}	f_{pa1}	P_{pa}	R_{pa}	f_{pa1}
USAD [8]	0.9870	0.7402	0.8460	0.6451	0.3320	0.4296
OmniAnomaly [11]	0.7223	0.9832	0.8328	0.2652	0.9799	0.4174
LSTM-VAE [10]	0.7123	0.9258	0.8051	0.4632	0.3220	0.3799
DAGMM [12]	0.8292	0.7674	0.7971	0.2228	0.1976	0.2094
THOC [17]	0.9808	0.7994	0.8809	0.4212	0.6334	0.5059
GTA [16]	0.9483	0.8810	0.9134	0.8391	0.8361	0.8376
GDN [15]	0.9585	0.9142	0.9359	0.8562	0.8541	0.8552
OCSVM [22]	0.9197	0.822	0.8681	0.5536	0.2382	0.3331
LOF [24]	0.8703	0.7531	0.8075	0.2484	0.3896	0.3034
IF [23]	0.9377	0.8261	0.8784	0.7736	0.7736	0.6822
PCA [21]	0.9564	0.8855	0.9196	0.9489	0.4032	0.5659
ICA [21]	0.9564	0.8855	0.9196	0.9943	0.1580	0.2727
$DCAE_a$	0.9614	0.8300	0.8909	0.4688	0.9324	0.6239
$DCAE_s$	0.9434	0.9757	**0.9593**	0.9365	1	**0.9672**

TABLE 4. Evaluation of MSL, SMAP and SMD datasets with point adjust.

Method	MSL			SMAP			SMD		
	P_{pa}	P_{pa}	f_{pa1}	P_{pa}	R_e	f_{pa1}	P_{pa}	R_{pa}	f_{pa1}
USAD [8]	0.8810	0.9786	**0.9109**	0.7697	0.9831	0.8186	0.9314	0.9617	0.9382
OmniAnomaly [11]	0.9140	0.8891	0.8952	0.7585	0.9756	0.8054	0.9809	0.9438	0.9441
LSTM-VAE [10]	0.8599	0.9756	0.8537	0.7164	0.9875	0.7555	0.8698	0.7879	0.8083
DAGMM [12]	0.7562	0.9803	0.8112	0.6334	0.9984	0.7124	0.6730	0.7231	0.7231
OCSVM [22]	0.6528	0.9668	0.7793	0.6420	0.9987	0.7816	0.8843	0.9302	0.9067
LOF [24]	0.3960	0.9954	0.5666	0.4520	1	0.6226	0.6911	0.8416	0.7589
IF [23]	0.6828	1	0.8115	0.5109	0.9863	0.6732	0.8927	0.906	0.8993
PCA [21]	0.7281	0.9687	0.8314	0.5242	0.9108	0.6655	0.9548	0.8879	0.9201
ICA [21]	0.7373	0.9668	0.8366	0.3427	0.9856	0.5086	0.9548	0.8879	0.9201
$DCAE_a$	0.7373	0.9811	0.8419	0.6198	0.9974	0.7645	0.8329	0.9086	0.8691
$DCAE_s$	0.7899	0.9973	0.8816	0.7105	0.9993	**0.8305**	0.9482	0.9561	**0.9521**

TABLE 5. Average accuracy of datasets with point adjust.

Method	Average f_{pa1}
	f_{pa1}
USAD [8]	0.7890
OmniAnomaly [11]	0.7790
LSTM-VAE [10]	0.7205
DAGMM [12]	0.6506
OCSVM [22]	0.5789
LOF [24]	0.6010
IF [23]	0.7890
PCA [21]	0.7805
ICA [21]	0.6915
$DCAE_a$	0.7981
$DCAE_s$	**0.9181**

TABLE 6. Comparison of the performance of Gauss-D and the seperated anomaly score.

Method	SWaT			WADI		
	P_t	R_e	f_{c1}	P_t	R_e	f_{c1}
$DCAE_a$	0.7575	0.7429	0.7501	0.3445	0.8571	0.4914
$DCAE_s$	0.8897	0.9714	**0.9288**	0.892	1	**0.9429**

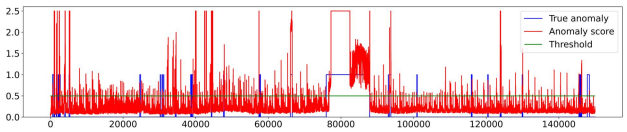

FIGURE 4. Result on SWaT using optimized aggregated dynamic Gauss-D.

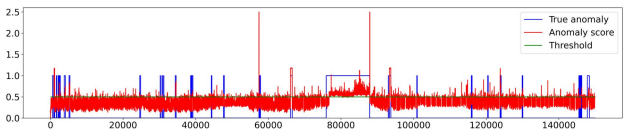

FIGURE 5. Result on SWaT using optimized separated thresholds.

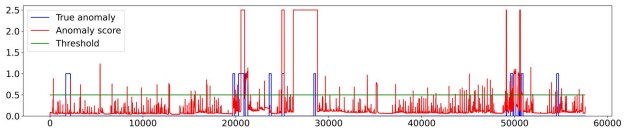

FIGURE 6. Result on WADI using optimized aggregated dynamic Gauss-D.

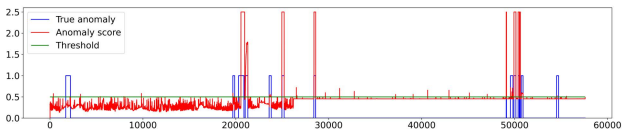

FIGURE 7. Result on WADI using optimized separated thresholds.

V. RESULTS

A. RESULT CONSIDERING BEST THRESHOLDS

Tables 2 and 3 present the results considering the threshold which gave the highest score for each method for SWaT and WADI with f_{p1} and point adjust f_{pa1}, respectively. Table 4 shows the result for point adjust for MSL, SMAP and SMD. As can be seen, the method suggested in this study has the highest score both with and without point adjust on SWaT and WADI. It also has the highest score on SMAP, third highest on MSL and the highest score on SMD. Furthermore, the average accuracy with point adjust, displayed in Table 5, is significantly higher than the other methods. Interestingly, DCAE with the aggregated method performed surprisingly well in relation to more complex methods and received the second highest average score.

B. COMPARISON TO AGGREGATION

If we compare the result of the aggregated scoring method and the separated method, it is clear that if optimized thresholds are used, the separated method is significantly superior. This is also true for the f_{c1} score displayed in Table 6. In figures 4 and 5, the result from the evaluation of the SWaT dataset for the respective methods are displayed and in figures 6 and 7, the same result is presented for WADI. As can be seen, it is much more difficult to separate the anomalies from non-anomalous points using the aggregated score than using the separated score. This result in anomalies found by the separated method becomes more distinct, while the score from the aggregated method appears to be more random.

TABLE 7. Different test data size for optimization with f_{c1} on SWaT dataset.

Method	0.25			0.5			1		
	P_t	R_e	f_{c1}	P_t	R_e	f_{c1}	P_t	R_e	f_{c1}
$DCAE_a$	0.9285	0.4091	**0.5679**	0.9901	0.3571	0.5249	0.7575	0.7429	0.7501
$DCAE_s$	0.3894	0.8636	0.5368	0.9374	0.9286	**0.9330**	0.8897	0.9714	**0.9288**

TABLE 8. Different test data size for optimization with f_{c1} on WADI dataset.

Method	0.25			0.5			1		
	P_t	R_e	f_{c1}	P_t	R_e	f_{c1}	P_t	R_e	f_{c1}
$DCAE_a$	0.0651	1	0.1222	0.5179	0.8571	0.6456	0.3445	0.8571	0.4914
$DCAE_s$	0.0661	1	**0.124**	0.8324	0.8571	**0.8446**	0.892	1	**0.9429**

C. GENERALIZABILITY

To verify that we are not simply over-fitting the thresholds to the particular types of anomalies seen in the test set, the aggregation and separated scoring method were optimized considering different sizes of the test set simulating a more online-like scenario. The results presented in tables 7 and 8, are based on the performance of the part of the test set that was not part of the optimization, meaning for example that for optimization of 25% of the test data, the evaluation is based on the remaining 75%. As can be seen, the aggregation method is slightly better at generalizing with no or few examples for the SWaT dataset, yet once the optimization size increases, the separated approach is superior. It is also worth mentioning that the aspect that reduces the f_1 score for low optimization size is the precision, meaning that the seperated method can still capture all the relevant anomalies but with a higher false positive rate.

VI. DISCUSSION

Considering the main research question of this study, the results strongly suggest that using separate thresholds can significantly increase the accuracy of an anomaly detection method when some labels have been defined. This also means that a separated scoring method seems to be superior to an aggregated method when a dataset of labeled anomalies is available. Yet, to generalize this conclusion, the separated scoring method should be applied to more deep learning methods. In addition, some of the results indicate that the aggregated method could have a greater sense of generalizability than using separated thresholds. This makes sense because using separated thresholds will create a more sensitive anomaly detection method. This is arguably both an advantage and potential setback of the suggested method. The advantage is that the method can be adjusted to identify anomalies only visible from one or a few signals, which can be difficult to achieve with aggregated methods. This means that a relatively high level of recall can be achieved regardless of whether or not the thresholds have been optimized. However, as the result has shown, the precision is expected to be fairly low without using an appropriate method to adjust the thresholds based on test data. This indicates that the most appropriate scoring method might be dependent on the problem definition. For example, if no labels are available, an aggregation based method might be most appropriate. Therefore, from a practical perspective the result should act as a strong incentive for users to provide a small sample of labels which can be used to optimize a separated scoring method, such as the one presented in this work, that is applied together with a deep learning model. From a theoretical perspective, it is recommended that future work focuses on improving scoring methods based on the separation of channels residuals but also to develop effective methods that utilize expert knowledge in labeling that can be used together with a separated scoring method to make the anomaly detection capability improve over time in an online-based scenario.

Furthermore, the results support the conclusions in [6] about the significant impact of the scoring method. This implies that bench-marking a method against other state-of-the-art methods considering both the algorithm and the scoring method as a system can give inconclusive results regarding the potential of the scoring method or the algorithm individually. Therefore, if the task is to evaluate the potential of a new algorithm for anomaly detection, the appropriate way of evaluating it would be to use a standardized scoring method that is used by the approaches the algorithm is compared to.

VII. CONCLUSION

This study set out to examine to what extent an anomaly score based on residuals from individual channels could outperform an aggregated anomaly score for anomaly detection on multivariate time series. To evaluate this, two different anomaly detection methods have been developed both based on the same underlying algorithm, the DCAE. The aggregated score was based on previous state-of-the-art work, and the separated method has been developed in the study. In addition to comparing the different approaches, other state-of-the art methods have been considered to get an understanding of how well the separated approach could perform. The result of the evaluation of three different datasets has shown that the performance of the separated approach significantly outperforms the aggregated approach. In addition, despite a relatively simple algorithm, the score from the separated scoring outperformed all of the other considered state-of-the-art methods. There were also some indications that the aggregated method could be slightly better at generalizing

when only few labels are available in the test set, which implies that an aggregated scoring method could be superior when no labels are available for threshold optimization. The result is an incentive for users to provide a small set of labeled points as this has the potential of significantly increasing the accuracy using a separated scoring approach. Lastly, the result in this study does, supported by [6], suggests that future studies concerning anomaly detection should consider separating the evaluation of the scoring method and the underling algorithm, because the scoring method can have a substantial impact on the performance of the overall method, which for example can mean that a method that shows great performance can be based on a suboptimal machine learning algorithm.

REFERENCES

[1] K. Choi, J. Yi, C. Park, and S. Yoon, "Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines," *IEEE Access*, vol. 9, pp. 120043–120065, 2021, doi: 10.1109/ACCESS.2021.3107975.

[2] G. Pang, C. Shen, L. Cao, and A. Van Den Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, Mar. 2021, doi: 10.1145/3439950.

[3] M. Hu, X. Feng, Z. Ji, K. Yan, and S. Zhou, "A novel computational approach for discord search with local recurrence rates in multivariate time series," *Inf. Sci.*, vol. 477, pp. 220–233, Mar. 2019, doi: 10.1016/j.ins.2018.10.047.

[4] S. Guha, N. Mishra, G. Roy, and O. Schrijvers, "Robust random cut forest based anomaly detection on streams," in *Proc. 33rd Int. Conf. Mach. Learn.*, New York, NY, USA, Jun. 2016, pp. 2712–2721.

[5] M. Hu, Z. Ji, K. Yan, Y. Guo, X. Feng, J. Gong, X. Zhao, and L. Dong, "Detecting anomalies in time series data via a meta-feature based approach," *IEEE Access*, vol. 6, pp. 27760–27776, 2018, doi: 10.1109/ACCESS.2018.2840086.

[6] A. Garg, W. Zhang, J. Samaran, R. Savitha, and C.-S. Foo, "An evaluation of anomaly detection and diagnosis in multivariate time series," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 1–10, Jun. 2021, doi: 10.1109/TNNLS.2021.3105827.

[7] N. Renström, P. Bangalore, and E. Highcock, "System-wide anomaly detection in wind turbines using deep autoencoders," *Renew. Energy*, vol. 157, pp. 647–659, Sep. 2020, doi: 10.1016/j.renene.2020.04.148.

[8] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "USAD: Unsupervised anomaly detection on multivariate time series," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 2020, pp. 3395–3404, doi: 10.1145/3394486.3403392.

[9] D. Li, D. Chen, L. Shi, B. Jin, J. Goh, and S.-K. Ng, "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks," 2019, *arXiv:1901.04997*.

[10] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 1544–1551, Jul. 2018, doi: 10.1109/LRA.2018.2801475.

[11] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Jul. 2019, pp. 2828–2837, doi: 10.1145/3292500.3330672.

[12] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen. (Feb. 2018). *Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection*. [Online]. Available: https://openreview.net/forum?id=BJJLHbb0-

[13] X. Xie, B. Wang, T. Wan, and W. Tang, "Multivariate abnormal detection for industrial control systems using 1D CNN and GRU," *IEEE Access*, vol. 8, pp. 88348–88359, 2020, doi: 10.1109/ACCESS.2020.2993335.

[14] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, and N. V. Chawla, "A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 1409–1416, doi: 10.1609/aaai.v33i01.33011409.

[15] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2021, pp. 1–9. [Online]. Available: https://www.aaai.org/AAAI21Papers/AAAI-5076.DengA.pdf

[16] Z. Chen, D. Chen, X. Zhang, Z. Yuan, and X. Cheng, "Learning graph structures with transformer for multivariate time-series anomaly detection in IoT," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9179–9189, Jun. 2021, doi: 10.1109/JIOT.2021.3100509.

[17] L. Shen, Z. Li, and J. Kwok, "Timeseries anomaly detection using temporal hierarchical one-class network," in *Advances in Neural Information Processing Systems*, vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds. Red Hook, NY, USA: Curran, 2020, pp. 13016–13026. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/97e401a02082021fd24957f852e0e475-Paper.pdf

[18] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "Do deep neural networks contribute to multivariate time series anomaly detection?" 2022, *arXiv:2204.01637*.

[19] F. R. S. K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *London, Edinburgh, Dublin Philosoph. Mag. J. Sci.*, vol. 2, no. 11, pp. 559–572, Nov. 1901, doi: 10.1080/14786440109462720.

[20] P. Comon, "Independent component analysis, a new concept?" *Signal Process.*, vol. 36, no. 3, pp. 287–314, Apr. 1994, doi: 10.1016/0165-1684(94)90029-9.

[21] A. A. Patel, *Hands-On Unsupervised Learning Using Python*. Sebastopol, CA, USA: O'Reilly Media, 2019. [Online]. Available: https://www.oreilly.com/library/view/hands-on-unsupervisedlearning/9781492035633/

[22] B. Schölkopf, J. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001, doi: 10.1162/089976601750264965.

[23] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 413–422, doi: 10.1109/ICDM.2008.17.

[24] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 93–104, 2000, doi: 10.1145/335191.335388.

[25] C. M. Ahmed, V. R. Palleti, and A. P. Mathur, "WADI: A water distribution testbed for research in the design of secure cyber physical systems," in *Proc. 3rd Int. Workshop Cyber-Phys. Syst. Smart Water Netw.*, Apr. 2017, pp. 25–28, doi: 10.1145/3055366.3055375.

[26] A. P. Mathur and N. O. Tippenhauer, "SWaT: A water treatment testbed for research and training on ICS security," in *Proc. Int. Workshop Cyber-physical Syst. Smart Water Netw. (CySWater)*, Apr. 2016, pp. 31–36, doi: 10.1109/CySWater.2016.7469060.

[27] *iTrust, Centre for Research in Cyber Security, Singapore University of Technology and Design*. Accessed: Jan. 21, 2022. [Online]. Available: https://itrust.sutd.edu.sg/itrust-labs_datasets/

[28] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 2018, pp. 387–395, doi: 10.1145/3219819.3219845.

[29] *Scikit-Learn: Machine Learning in Python—Scikit-Learn 1.1.1 Documentation*. [Online]. Available: https://scikit-learn.org/stable/

[30] A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," *Neural Comput.*, vol. 9, no. 7, pp. 1483–1492, Jul. 1997, doi: 10.1162/neco.1997.9.7.1483.

ADAM LUNDSTRÖM received the M.S. degree in industrial engineering and management from Mid Sweden University and the M.S. degree in computer and systems sciences from Stockholm University, in 2020. He is currently pursuing the Ph.D. degree with the Industrial Graduate School Smart Industry Sweden and studying the potential for machine learning solutions in predictive maintenance. He is employed with SCA and a part of the Department of Electronics Design, Mid Sweden University.

MATTIAS O'NILS received the B.S. degree in electrical engineering from Mid Sweden University, Sundsvall, Sweden, in 1993, and the Licentiate and Ph.D. degrees in electronic systems design from the Royal Institute of Technology, Stockholm, Sweden, in 1996 and 1999, respectively. He is currently a Professor with the Department of Electronics Design and leads the Research Group in Embedded IoT Systems, Mid Sweden University. His current research interests include design methods and implementation of embedded DNN-based systems, especially in the implementation of real-time video processing and time series processing systems.

FAISAL Z. QURESHI (Senior Member, IEEE) received the B.Sc. degree in mathematics and physics from Punjab University, Lahore, Pakistan, in 1993, the M.Sc. degree in electronics from Quaid-e-Azam University, Islamabad, Pakistan, in 1995, and the M.Sc. and Ph.D. degrees in computer science from the University of Toronto, Toronto, ON, Canada, in 2000 and 2007, respectively. He is currently a Professor of computer science with the Faculty of Science, Ontario Tech University, where he leads the Visual Computing Laboratory. His research interest includes computer vision. His scientific and engineering interests center on the study of computational models of visual perception to support autonomous, purposeful behavior in the context of ad hoc networks of smart cameras. He is also active in journal special issues and conference organizations. He is a member of ACM and a Secretary and a member of CIPPRS. He served as the General Co-Chair for the Workshop on Camera Networks and Wide-Area Scene Analysis (co-located with CVPR) during 2011–2013. He also served as the Co-Chair for the Computer and Robot Vision (CRV) Conference 2015/2016 meetings.

AXEL JANTSCH (Senior Member, IEEE) received the Dipl.Ing. and Ph.D. degrees in computer science from TU Wien, Vienna, Austria, in 1987 and 1992, respectively. From 1997 to 2002, he was an Associate Professor with the KTH Royal Institute of Technology, Stockholm. From 2002 to 2014, he was a Full Professor of electronic systems design at the KTH. Since 2014, he has been a Professor of systems on chips with the Institute of Computer Technology, TU Wien. He has published five books as an editor and one as the author and over 300 peer-reviewed contributions in journals, books, and conference proceedings. He has given over 100 invited presentations at conferences, universities, and companies. His current research interests include systems on chips, self-aware cyber-physical systems, and embedded machine learning.