Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Real-World Anomaly Detection with Siamese Neural Networks and Digital Twin Systems

Andrea Castellani, Sebastian Schmitt, Stefano Squartini

2020

Preprint:

This is an accepted article published in IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS. The final authenticated version is available online at: https://doi.org/10.1109/TII.2020.3019788 Copyright 2020 IEEE

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2020.3019788, IEEE Transactions on Industrial Informatics

Real-World Anomaly Detection by using Digital Twin Systems and Weakly-Supervised Learning

Castellani Andrea, Schmitt Sebastian, and Stefano Squartini, *Senior Member, IEEE*

***Abstract*—The continuously growing amount of monitored data in the Industry 4.0 context requires strong and reliable anomaly detection techniques. The advancement of Digital Twin technologies allows for realistic simulations of complex machinery, therefore, it is ideally suited to generate synthetic datasets for the use in anomaly detection approaches when compared to actual measurement data. In this paper, we present novel weakly-supervised approaches to anomaly detection for industrial settings. The approaches make use of a Digital Twin to generate a training dataset which simulates the normal operation of the machinery, along with a small set of labeled anomalous measurement from the real machinery. In particular, we introduce a clustering-based approach, called Cluster Centers (CC), and a neural architecture based on the Siamese Autoencoders (SAE), which are tailored for weakly-supervised settings with very few labeled data samples. The performance of the proposed methods is compared against various state-of-the-art anomaly detection algorithms on an application to a real-world dataset from a facility monitoring system, by using a multitude of performance measures. Also, the influence of hyper-parameters related to feature extraction and network architecture is investigated. We find that the proposed SAE based solutions outperform state-of-the-art anomaly detection approaches very robustly for many different hyper-parameter settings on all performance measures.**

***Index Terms*—Anomaly Detection, Digital Twin Systems, Siamese Neural Networks, Weakly-Supervised Learning, Industry 4.0.**

I. Introduction

In recent years, there has been a strong tendency to equip technical machinery, ranging from single machines to complete buildings and manufacturing plants with sensors to constantly monitor their operation, especially in the context of Industry 4.0 strategies [1]. Generally, component failures or complete system failures need to be avoided as this severely impacts the functioning of the machines and leads to significant increase in maintenance, overhaul and repair (MRO) costs. In many situations, precursors of component failures can be observed in the time-series of measured sensor data and predictive maintenance approaches try to use this to reduce MRO downtime and cost [2]. An essential part of these approaches are robust and reliable anomaly detection methods which work in real-world settings.

A. Castellani and S. Schmitt are with the Honda Research Institute Europe GmbH, 63071 Offenbach, Germany (e-mail: a.castellani93@gmail.com, sebastian.schmitt@honda-ri.de)

A. Castellani is with the Machine Learning group, CITEC Excellence Cluster of Bielefeld University, 33619 Bielefeld, Germany. (e-mail: acastellani@techfak.uni-bielefeld.de)

S. Squartini is with the Department of Information Engineering, Università Politecnica delle Marche, 60131 Ancona, Italy (e-mail: s.squartini@univpm.it)

Manuscript received 21 July 2020; accepted 13 August 2020.

Anomaly detection [3] refers to the problem of finding patterns in data that do not conform to expected or normal behavior. It is an active area of research with a wide range of application areas, such as energy [4], manufacturing [5], network sensors [6], health care and video surveillance [7]. Anomaly detection techniques based on machine learning can be separated into different types of approaches [8]: supervised approaches, where a sufficiently large set of training samples with labelled data is available; unsupervised approaches, where only the unlabelled measurement data is available; and weakly-supervised approaches, where a large amount of unlabelled data with a very small set of labeled data is available. The distinction between these cases is not clear-cut, as there could be a supervised situation where a large amount of labelled data is available, but this data is only from the normal operation. In that case, the training set does not reflect the true distribution of data in the real-world, and even more importantly, the most important information carrying data, i.e. labelled anomaly samples, are missing.

For a practical application in realistic circumstances, even if there are many data available, only unlabelled data is usually available, since it requires a tremendous effort by human experts to manually create a fully labeled dataset with a large portion of the possible anomalous scenarios. This requires unsupervised anomaly detection methods to be used. In those methods, the bulk statistics of the data is learned and the prevailing features of the data are considered normal. The decision whether a data sample is normal or not is then based on the comparison to the learned statistics so that infrequent data samples are more likely to be considered anomalous. This implies a very high class imbalance for the normal and the abnormal classes. Tuning the sensitivity of the anomaly detection algorithm is a rather difficult problem in such a situation. Even a very small false positive rate leads to a very large absolute amount of wrongly detected anomalies which renders many approaches useless for practical applications. Additionally, for most complex machinery, the definition of an anomaly is not so clear. There might exist normal operation modes, which are very rare and therefore, from a statistical viewpoint, anomalies. Also real-world machinery exhibits drift of data due to de-calibration of sensors, wear and tear and degradation of the machinery. So the performance of an anomaly detection which was trained at a specific instance in time might degrade over time due to the drift in the data and regular re-training might be necessary.

In recent years, simulation technology has advanced substantially and even complex machinery can nowadays be simulated quite accurately. It is even possible to incorporate input

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2020.3019788, IEEE Transactions on Industrial Informatics

data from actual measurements and to mirror the operational history of individual machines, which is known as Digital Twin systems first introduced in 2003 [9] (see also [10]). Depending on the effort spent to create the Digital Twin, it can be realistic simulation which captures the qualitatively correct behavior of the machinery, or an almost perfect digital copy whose output can be directly compared to the real-world machinery. The former can be at least used to create a large dataset containing data samples of normal operation conditions which can be utilized in machine learning approaches. The latter can play a key role in the anomaly detection problem [11]–[13], when it runs in parallel to the physical system with the same input values and environmental conditions.

In this paper, we present novel weakly-supervised approaches to anomaly detection which we apply to the data from a company facility monitoring system. We generate a large dataset of normal operation data covering a complete year of operation using a Digital Twin simulation of the system. This enables the accurate unsupervised learning of the statistics of the normal operation states statistics including the very rare but normal states. In order to increase the sensitivity for anomalies, we additionally use a very small dataset of labeled anomalous samples which is obtained from the real-world measurements. This weakly-supervised approach has the advantage to produce a substantially lower false positive rate making it suitable for usage in the real-world. Additionally, the Siamese Networks can be retrained very efficiently as soon as new labelled data is available, making it ideal for refinement during usage as well as for adjusting to drift and re-configurations of the monitored machinery during operation.

The main contributions of this work are: (i) novel approaches to anomaly detection using data from a Digital Twin simulation for the normal operational state of a machinery, (ii) a clustering-based algorithm capable to solve the anomaly detection task in both unsupervised and weakly-supervised settings, (iii) a Siamese Autoencoder architecture for weakly-supervised anomaly detection where very few labelled training samples are needed to improve the performance over unsupervised anomaly detection methods, (iv) a thorough comparison of experimental results of the proposed methods and their performance to state-of-the-art algorithms for anomaly detection.

The rest of the paper is organized as follows. Section II presents an overview of the recent literature on anomaly detection on time-series and the usage of Digital Twin. Section III shows the proposed approach and presents the developed algorithms. Section IV presents the used Digital Twin framework. Section V shows the comparative anomaly detection algorithm investigated in this work and Section VI presents the dataset and describes the experimental setup. Section VII discusses the results obtained in our experimentation. Finally, Section VIII concludes the paper and presents future developments.

II. Related Works

Anomaly detection can be approached in different ways depending on the kind of data available and the requirements of the particular use case. Chandola et al. [3] presented a survey of different anomaly detection techniques applied to different domains and recently the problem has been addressed from the perspective of Deep Learning by Chalapathy et al. [8]. One category of deep anomaly detection methods is based on reducing the dimension of the input by mapping it to a low dimensional manifold followed by reconstruction. These networks are usually trained in unsupervised fashion with a dataset containing only data that reflects the normal state of operation. The reconstruction error is then used to distinguish anomalies with several different neural architectures, for example Autoencoders (AE) [14], Long Short Memory Network (LSTM) [6], Adversarial Autoencoder (AAE) [15] or Generative Adversarial Networks (GAN) [16]. When dealing with multivariate time-series data, Canizo et al. [17] use a CNN-RNN capable to reach an AP score of 0.994 on a real-world dataset, but the proposed architecture is trained with a fully supervised approach. Other approaches to anomaly detection make use of data-driven statistical methods [4], [18], unsupervised clustering [19] and density-based clustering [5].

Tao et al. [10] presents a detailed survey of the actual state-of-the-art of the Digital Twin in industry. Several applications are presented in different areas of design, production, prognostic and health management. To the best of our knowledge, there are just few references in literature to the usage of Digital Twin in the context of anomaly detection [11]–[13] and all of those are more focused to present the Digital Twin system rather than a anomaly detection algorithm.

While there are several supervised and unsupervised approaches to anomaly detection described in literature, only few weakly-supervised approaches have been proposed. In [20] the authors use domain adversarial training to transfer the knowledge learned from a normal dataset to another with few labeled samples. The Siamese Networks approach was introduced for signature verification in [21] in a fully supervised setting. A first weakly-supervised approach using Siamese Networks was made by Koch et al. [22]. Regarding anomaly detection, there are some applications of Siamese Autoencoders including human fall [23] and medical application [24]. Both architectures make use of Siamese Autoencoder to extract features from the data, but the anomaly detection process is made by an additional classifier, a KNN in the former and OCSVM in the latter. Utkin et al. [25] presents a variation of the Siamese Autoencoder with application to robotics. To the best of our knowledge, no previous attempts to approach the anomaly detection problem with weakly-supervised Siamese Networks and the combination of Digital Twin and real-world data have been proposed.

III. Proposed Algorithms

The overall pipeline of the proposed anomaly detection scheme is sketched in Fig. 1. It consists of a training phase of an anomaly detection algorithm (dotted rectangle) with the combination of two datasets: normal operation dataset $\mathcal{N}$, generated with the Digital Twin simulation, and a small set $\mathcal{A}$ of labeled anomalous samples from the real-world measurement system. During the regular operational phase (blue boxes), the physical values from the machinery provide a real-world dataset $\mathcal{R}$, which is fed to the trained anomaly

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2020.3019788, IEEE Transactions on Industrial Informatics

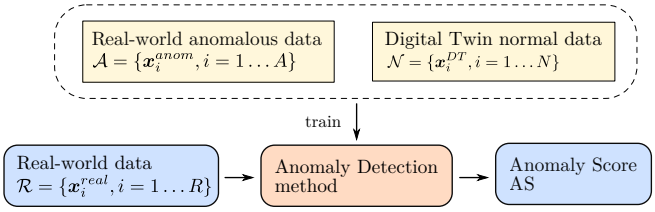


Fig. 1. Proposed pipeline for Anomaly Detection.

detection algorithm. An Anomaly Score (AS) is calculated for each data sample and it is considered an anomaly if its score exceeds a certain threshold.

A. Cluster Centers

We propose the Cluster Centers (CC) algorithm as a weakly supervised classification approach to the anomaly detection problem. In a first step, the Digital Twin dataset of normal samples $\mathcal{N}$ is processed with an unsupervised clustering algorithm, which identifies distinct normal operational modes of the industrial machinery. Here, we use a k-means approach, but in general any clustering algorithm is suitable, as long as it is possible to calculate the set of cluster centers, $\mathcal{C} = \{\boldsymbol{c}_j \text{ for } j = 1, ..., N_C\}$.

The number of clusters N_C and the clusters themselves do not need to represent the real semantically correct operational modes of the machinery, but rather organize the data into different clusters which cover the complete statistical variations of data found during normal operation. For this reason, the number of clusters should be considerably larger than the (guessed or known) true number of operational states so the clusters can also cover larger variations, transitions and switching behavior properly.

The cluster centers $\boldsymbol{c}_j$ from the clustering algorithms of only normal data samples from the Digital Twin can already be used to predict the operational clusters on the real-world data set $\mathcal{R}$. The distribution of cluster centers reflects only normal operation, and only from the Digital Twin. The expectation is that the normal operations data from the real-world aligns with this clustering, while anomalous data samples are quite distinct from normal operation data and cannot be assigned reasonably to these cluster centers. Therefore, the cluster centers can provide a fully unsupervised estimate for the AS, which is given by the distance (d) from each data point ($\boldsymbol{x}_r$) to its nearest cluster center ($\boldsymbol{c}$). For the k-means approach we use the Euclidean distance, but any other distance measure could also be used.

The set of labeled anomalies, $\mathcal{A}$, is used to refine the AS of each real-world sample $\boldsymbol{x}_r$ by adding a penalty term,

$$AS(\boldsymbol{x}_r) = \underbrace{\min_j(d(\boldsymbol{c}_j, \boldsymbol{x}_r))}_{\text{Unsupervised AS}} + \underbrace{\eta \frac{1}{\min_a(d(\boldsymbol{x}_r, \boldsymbol{x}_a)) + \zeta}}_{\text{Penalty term}}, \quad (1)$$

with the cluster centers $\boldsymbol{c}_j \in \mathcal{C}$, the anomalous data samples $\boldsymbol{x}_a \in \mathcal{A}$, a hyper-parameter $\eta > 0$ determining the influence of the penalty and a regularization factor $\zeta > 0$. This penalty term increases the AS whenever a known anomaly is close to the data sample. With this term, the information about the known anomalies is used to refine the knowledge provided by the cluster centers of normal operation, and allows for a better discrimination between normal and anomalous data samples.

The proposed CC and a regular k-Nearest Neighbors (KNN) clustering approach differ in the way the AS is calculated. We use the distance to the nearest cluster center, whereas in a KNN approach the nearest data samples are used. Also, the penalty term is not used in regular KNN approaches.

In the above algorithm more hyper-parameters could have been introduced, for example, an exponent different from 1 for the penalty term in (1) which would allow for asymmetric weighting for small and large distances. This was purposely not done to keep the approach as simple as possible. Also, instead of including the labeled anomaly samples via a penalty term after the clustering, they could have been included from the beginning by using a constrained clustering algorithm. This was not done in order to deal more easily with a constantly growing and changing set of labeled anomalies, which can be used in the AS without redoing the clustering.

B. Siamese Autoencoder

The Siamese Networks consist of two identical networks with shared weights which can be efficiently used to decide if a pair of input data samples comes from the same distribution or not.

The structure of the proposed architecture is shown in Fig. 2. It comprises an encoder E which maps the input data sample $\boldsymbol{x} \in \mathbb{R}^N$ to a latent representation $\boldsymbol{h}(\boldsymbol{x}) \in \mathbb{R}^M$, with $M < N$. The decoder network D takes the latent representation and reconstructs a data sample in the original space, $\tilde{\boldsymbol{x}} \in \mathbb{R}^N$. We refer the reconstruction of a given input sample, $\boldsymbol{x}$, as $\tilde{\boldsymbol{x}} = D(E(\boldsymbol{x})) = D(\boldsymbol{h}(\boldsymbol{x}))$, and its latent representation as $\boldsymbol{h}(\boldsymbol{x}) = E(\boldsymbol{x})$. We use the Euclidean distance d.

The main goal of the proposed network is to reconstruct normal data samples with low error and to create a clear separation of the normal and the anomalous data distribution in the latent space. This can be achieved by the following behavior of the network:

1) Normal data samples should be reconstructed as good as possible: $\boldsymbol{x} \simeq \tilde{\boldsymbol{x}}$ for $\boldsymbol{x} \in \mathcal{N}$.
2) The distance between two latent representations of any pair of normal data samples should be as small as possible: $d(\boldsymbol{h}(\boldsymbol{x}), \boldsymbol{h}(\boldsymbol{x}'))$ small for $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{N}$.
3) Anomalous data samples should not be reconstructed well, i.e. their reconstruction error should be much larger than the reconstruction errors for normal data: $d(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ large for $\boldsymbol{x} \in \mathcal{A}$.
4) The distance between the latent representations of a normal and an anomalous data sample should be large: $d(\boldsymbol{h}(\boldsymbol{x}), \boldsymbol{h}(\boldsymbol{x}'))$ large for $\boldsymbol{x} \in \mathcal{N}$ and $\boldsymbol{x}' \in \mathcal{A}$.

The desired behavior can be achieved by implementing an appropriate training procedure for the neural network. The network always evaluates a pair of data samples, $(\boldsymbol{x}_{DT}, \boldsymbol{x}_S)$, where the first sample is always taken from the normal data set, $\boldsymbol{x}_{DT} \in \mathcal{N}$. In our approach normal data samples for training are always taken from the Digital Twin dataset, which

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2020.3019788, IEEE Transactions on Industrial Informatics

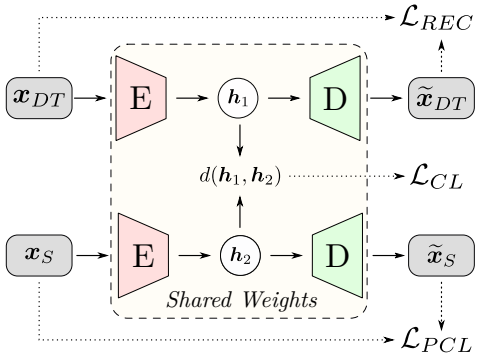


Fig. 2. Siamese Autoencoder structure with the three loss functions.

is why we include the subscript DT for clarity. The second sample can be either from the normal or the anomalous dataset, $\boldsymbol{x}_S \in \mathcal{N} \cup \mathcal{A}$. The loss function for training then has the following three contributions

$$\mathcal{L} = \mathcal{L}_{REC} + \mathcal{L}_{CL} + \mathcal{L}_{PCL} \tag{2}$$

where each contribution is calculated as sums over all pairs of data $(\boldsymbol{x}_{DT_i}, \boldsymbol{x}_{S_i})$ and are given by the following expressions:

- **Reconstruction Loss:** the Mean Square Error (MSE) between the input and its reconstruction for normal operation data, which is just the cost function of a typical Autoencoder:

$$\mathcal{L}_{REC} = \frac{1}{N} \sum_{\text{pairs } i} \left(\tilde{\boldsymbol{x}}_{DT_i} - \boldsymbol{x}_{DT_i}\right)^2. \tag{3}$$

- **Contrastive Loss:** the Euclidean distance between the latent vectors of the input pair with a local modification:

$$\mathcal{L}_{CL} = \frac{1}{2N} \sum_{\text{pairs } i} \Bigg((1 - Y_i)\, d\big(E(\boldsymbol{x}_{DT_i}), E(\boldsymbol{x}_{S_i})\big)^2 + Y_i \left\{ \max\Big(0, m - d\big(E(\boldsymbol{x}_{DT_i}), E(\boldsymbol{x}_{S_i})\big)\Big) \right\}^2 \Bigg), \tag{4}$$

  where $Y_i = 0$ if $\boldsymbol{x}_{S_i} \in \mathcal{N}$ and $Y_i = 1$ if $\boldsymbol{x}_{S_i} \in \mathcal{A}$. This contribution minimizes the difference for input from the same class ($Y = 0$) while maximize the differences for input of different class ($Y = 1$). The parameter $m > 0$ allows only samples whose distance is less than the radius defined by m to contribute to the loss function in order to avoid the domination of individual samples.
- **Partial Contrastive Loss:** enforces a large reconstruction error for anomalous data sample, which can also be viewed as a gradient inversion in the Autoencoder training process:

$$\mathcal{L}_{PCL} = \frac{1}{2N} \sum_{\text{pairs } i} Y_i \, \max\big(0, m - d\left(\tilde{\boldsymbol{x}}_{S_i}, \boldsymbol{x}_{S_i}\right)\big) \tag{5}$$

Due to the possibility to construct a very large training dataset of distinct pairs from the large normal dataset and very small anomalous dataset, this approach is able to deal with extremely unbalanced datasets, $|\mathcal{N}| \gg |\mathcal{A}|$.

After training, the AS for a new real-world data sample $\boldsymbol{x}_r \in \mathcal{R}$ is calculated as:

$$AS(\boldsymbol{x}_r) = \underbrace{\left(\tilde{\boldsymbol{x}}_r - \boldsymbol{x}_r\right)^2}_{\text{Reconstruction error}} + \underbrace{\frac{1}{N'} \sum_{i=1}^{N'} \left\| E(\boldsymbol{x}_{DT_i}) - E(\boldsymbol{x}_r) \right\|_2}_{\text{Embedding distance}}, \tag{6}$$

where $\boldsymbol{x}_{DT_i}$ are from a subset of the normal operation dataset from the training phase with $N' \leq N$ elements.

In this work we employ the Siamese architecture for two types of input data. In the so-called Siamese Autoencoder (SAE) we extract feature vectors from the measurement time series data and use standard feed forward autoencoder architectures for encoder and decoder which are chosen to be symmetric. The second variant called Convolutional Siamese Autoencoder (CNN-SAE) directly operates on raw time series data and employs 1-D Convolutions, where encoder and decoder are also chosen to be symmetric.

IV. Digital Twin

The Digital Twin model used in this work simulates the electrical power system, the Heating, Ventilation and Air Conditioning (HVAC), Combined Heat and Power (CHP) systems for a medium sized company facility. The simulation is realized with the *Green City library*[1] and the *SimulationX* software based on Modelica programming language. The detailed process of the calibration of the Digital Twin with the measurement data from the machinery is presented in a separate publication [26]. This work focuses on the CHP model which takes the actual measurements form weather data and total power demand time-series as input.

Due the high complexity of modeling realistic machines and the heating and cooling of a company facility, the Digital Twin is only able to simulate with very high fidelity power and energy consumption of the machines in a reasonably large window of time, i.e. not less than few hours. The exact transient behavior of the machines cannot be simulated exactly so comparing raw time series between Digital Twin and the measurement data is not suitable on the sampling rate of one data point per minute. However, we have validated the Digital Twin model with several periods of correct operation of the machine in 2018 and there is a discrepancy with the real-world electrical and thermal energy production of 2.18%.

V. Comparative Methods

The simplest method for deriving an AS is given by directly comparing the raw time-series of the measurement with the simulation and taking the Mean Absolute Error (MAE). As state-of-the-art algorithms for anomaly detection we employ an Isolation Forest (IF), which builds an ensemble of isolation trees for a given dataset, where anomalies are those data samples with short average path lengths in the trees. In the k-Nearest Neighbors (KNN) approach the measure of outlierness of an observation is obtained based on the distance to its neighbors. We also compare the results of supervised Support Vector

[1] http://ea-energie.de/en/products/green-city-simulationsbibliothek-2-2/

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2020.3019788, IEEE Transactions on Industrial Informatics

Machine (SVM), its unsupervised variant One-Class SVM (OCSVM), as well as to the Local Outlier Factor (LOF), which measures the outlierness of data points by its local deviation of densities with respect to its neighbors. The unsupervised dimensional reduction method Principal Component Analysis (PCA) is used as an anomaly detection technique by taking the reconstruction error of each sample as AS. Finally we also compare to a Deep Learning-based Feed Forward Autoencoder (FF-AE) and a supervised Multi Layer Perceptron (MLP).

All unsupervised algorithms are trained on the Digital Twin dataset $\mathcal{N}$ and then evaluated against the real-world data $\mathcal{R}$. In addition, one OCSVM is also trained and tested on real-world data only. The weakly-supervised and supervised algorithms are trained with the Digital Twin dataset and few randomly sampled anomalies from $\mathcal{A}$.

VI. Experimental Setup

A. Dataset Description

In this study, we use a dataset from a medium sized company, where an infrastructure monitoring system recording various energy-related modules is installed. The recorded data consists of a large number of sensors for heat, cold and electricity consumption and production, as well as a local weather station to monitor the ambient environment. This work focuses on data recorded from the CHP module, where natural gas is burned to produce heat and electrical power. The recorded time-series are: consumed and produced energy, produced heat, heating fluid volume flow, and flow and return temperatures for the heating fluid, as shown in Table I.

The dataset consists of data from *November 2017* to *February 2019*. The sampling rate is one data point per minute, leading to a total of 658081 sample points.

The synthetic Digital Twin dataset consists of the same time-series and the same processing as for the real-world dataset, but only the year 2018 was simulated.

The anomalous real-world dataset has been labeled manually. A total of 100 failure instances have been identified which translates to 24.2% of the total number of samples being anomalous. There are two kinds of anomalies. One is produced by sensor failures which result in a flat time series, and which are rather easy to detect. More interesting and harder to detect are those situations in which each time-series appears to provide a valid measurement, but the combination multiple sensor reading does not reflect normal operation. For the CHP, this is the case when, for example, the ambient temperature is rather low but still no electricity and heat are produced.

B. Data Pre-Processing and Evaluation Metrics

For all approaches except the CNN-SAE, we use features derived from the raw data as input to the models. The implemented statistical features are listed in Table I and are extracted with a sliding window approach with parameters also shown in the aforementioned table. The contextual feature refers to the number of machine shutdowns (which can be easily detected from the data) and the total working time within the time window. The time features include the information related to working days and season of the year.

TABLE I
Data Hyperparameters explored in the partial grid search. Values in bold have been used for the final configuration.

Hyperparameter	Value
Variables	**Ambient Temperature** (Ta) **Produced Thermal Power** (P_{Th}) Produced Thermal Energy (E_{Th}) **Produced Electrical Power** (P_{El}) Produced Electrical Energy (E_{El}) Water flux ($Flux$) Water flow temperature (T_{Flow}) Water return temperature (T_{Return}) Flow/return temperature difference (T_{Diff})
Statistical Features	**Mean** (μ) **Standard deviation** (σ) Skewness Kurtosis Sum Root Mean Square
Time Features	True, **False**
Contextual Feature	True, **False**
Window Length	240, 360, 480, 720, **1440**
Window Stride	30, **60**, 120

The feature vectors are composed of data from different domains and scales, and therefore need to be standardized with the *z-score* : $Z = (X - \bar{x})/\sigma$, where $\bar{x}$ is the mean of X and σ its standard deviation. Regarding time and contextual values, One-Hot Encoding is used to represent them as linearly uncorrelated vectors.

We evaluate the performance of the discussed approaches by using multiple metrics. We use the F_2 Score, the Area Under Receiver Operating Characteristic curve (AUC ROC) and the Average Precision (AP), which is a measure of the Area Under the Precision-Recall Curve (PRC). In the situation with highly imbalanced class sizes, it was shown that the PRC is more informative than ROC [27], since it better reflects the correct prediction of the minority class.

C. Implementation and Training Details

The state-of-the-art anomaly detection algorithms used for the comparison are taken from the open source library *PyOD* [28]. The neural networks are implemented with the Keras library and *TensorFlow 1.13* back-end. The data processing and training of the algorithms is done on two workstations, one with a quad-core CPU (Intel Core i5-7300HQ) and 16 GB of DDR4 RAM and another one with an Intel Xeon X5680 (6 cores and 12 threads), 128 GB of DDR4 RAM and a NVIDIA TITAN X GPU.

We use 10-fold Cross Validation, where the Digital Twin dataset and the subset of labeled anomalies for training are divided in 10 folds to train and validate the algorithm's performance. The real-world dataset is divided in two parts, 20% for validation and 80% for the final test evaluation.

In order to determine the best performing combination of hyper-parameters for the feature extraction and the architectures of the Neural Networks, a Partial Grid Search [29]

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2020.3019788, IEEE Transactions on Industrial Informatics

TABLE II
AUTOENCODERS NETWORKS HYPERPARAMETERS EXPLORED IN THE GRID SEARCH.

Hyperparameter	**Range**	**FF-AE**	**SAE**	**CNN-SAE**
Batch size	$\{16, 32, 64, 128\}$	64	128	32
Learning rate	$[10^{-2}, 10^{-5}]$	10^{-4}	10^{-3}	10^{-4}
Dropout	$[0, 0.5]$	0	0	0.1
Epoch	$[50, 200]$	150	150	100
Activation function	{tanh, ReLu, sigmoid}	tanh	ReLu	ReLu
Nr. of Encoder/Decoder layers	$[1, 5]$	3	4	5
Encoder/Decoder neurons / kernel*	$[8, 512]$	64 32 16	128 64 32 16	32 64 128 128 256
Encoder/Decoder kernel length	$[1, 10]$	/	/	10 7 5 4 3
Encoder/Decoder kernel stride	$[1, 3]$	/	/	2 2 2 3 1
Nr. of fully-connected layers	$[1, 3]$	/	/	1
Fully-connected neurons	$[10, 100]$	/	/	60
Embedding space dimension	$[1, 3]$	3	2	2

* Only for Convolutional Networks

has been used. Instead of searching in the complete hyper-parameter space by computing all the possible combination of the parameters, the search is organized as an iterated line-search. All parameters except one are fixed and the optimization is performed only along the dimension of the free parameter. Then, the best performing value for this parameter is selected and the search is done along another parameter dimension. The resulting hyper-parameters related to data and feature extraction are shown in Table I. The values in bold constitute the best configuration of parameters, determined by the search approach with Cross Validation. An example of this approach is reported in Fig. 3, which illustrates the performance variations for some different settings related to the feature selection. The left panel shows the performance as function of the statistical quantities included in the feature vector. The performance for different input time-series is instead shown in the right panel. The qualitative changes and trends in the performance are the same for all algorithms, only the SAE sometimes has a slightly different trend. The variation of the statistical features (left panel) shows that the best performance is achieved when only mean (μ) and standard deviation (σ) are used. The inclusion of higher statistical moments like skewness and kurtosis leads to a degradation of the performance most likely due to the fact that the short time-scales, switching and state transitions, are not captured well by the Digital Twin simulation and therefore cannot contribute to a successful classification on the real-world data. By inspecting the curve in the right panel we can state that the ambient temperature T_a is a rather important measurement, since not including it in the feature vector leads to a performance drop from around 0.8 to around 0.3. This was expected since the CHP is controlled on that variable. More sensors do not improve the performance, also the inclusion of the flow and return temperatures do not seem to be effective in this sense, for most algorithms. Only the SAE shows high performance when the flow temperature is used.

The resulting final dataset of feature vectors includes 8737 samples from the Digital Twin and 10945 samples from the real-world, of which 2162 samples are labeled anomalies. Each sample consists of a feature vector of dimension 6.

The hyper-parameters of the Neural Network architectures

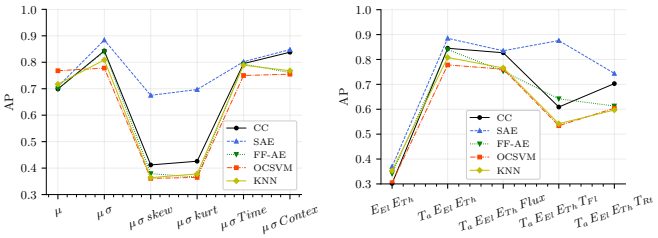


Fig. 3. Comparison of the AP performance for different settings for the selected statistical features (left) and input time series (right). The symbols and names shown on the x-axis are explained in Table I. Other features are those shown in bold in Table I.

are shown in Table II, with ranges used in the search approach reported in the first column. We use a symmetrical structure for each Autoencoder. In particular, in the CNN case, the encoder is composed of five convolutional layers followed by a global max-pooling layer and then a fully-connected layer. The decoder has five de-convolutional layers (up-sampling layer followed by a convolutional layer). In order to revert the loss of information in the global max-pooling layer, a skip-connection with linear activation is used between the last convolutional layer of the encoder and the first de-convolutional layer of the decoder. This is crucial to reconstruct the input data.

For training all neural architectures, we use the Adam optimizer with *"LeCun Uniform"* weight initialization [30] and early stopping. We also apply gradient clipping with maximum L_2 norm of 4 to improve training stability. For weakly-supervised methods, the set of labeled anomalies for training is composed of only 10 samples, randomly selected from the available labeled data. For the CC algorithm, we set $\zeta = 0.001$ and $\eta = 0.15$, since those are the best performing combination. For the OCSVM and SVC, we use a RBF kernel with $C = 1$. For the MLP we use a network with 3 hidden layer of 50 neurons each. Each experiment has been repeated 10 times and, as results, we present the mean and standard deviation of the performance measures calculated on the test set.

In order to assign a concrete label to a given data sample, we need to apply a threshold (θ_{AS}) to the calculated anomaly score (AS). We set the threshold such that 25% of the test

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2020.3019788, IEEE Transactions on Industrial Informatics

data is above the threshold and is considered anomalous. This seemingly arbitrary choice reflects our a-priori assumption of the expected anomaly rate of the machinery under investigation.

For the sake of clarity, the explicit choice of θ_{AS} is needed to calculate the F_2 score and the confusion matrices. For the others performance measures used in this work, AP and ROC, all possible values of this threshold are used.

VII. Experimental Results

In this section, we report first the performance of the proposed unsupervised and weakly-supervised approaches, along with other state-of-the-art algorithm for anomaly detection presented in section V. We apply all algorithms to the real-world use-case data introduced in section VI, where we use the Digital Twin to generate the simulation training data. We also analyze how the labeled anomalies used in the training affects the performance for the weakly-supervised approaches. Then, for completeness, we show the usefulness of the Digital Twin simulation data by comparing the performance of the anomaly detection algorithms to a case where training is done using only real-world data.

A. Results obtained with Digital Twin data simulation

The results obtained in our experiments, using the Digital Twin to generate the training set, are reported in Table III. For the weakly-supervised methods, 10 randomly selected anomalous data samples were included in the training.

The naive MAE approach of directly comparing Digital Twin data to real measurement data gives a medium performance of $AP = 0.69$. It is worth noting that simple unsupervised clustering-based approaches like KNN and the proposed CC already give rather good scores of about $AP = 0.81$ and $AP = 0.83$, respectively. The best performing unsupervised method is the deep learning based FF-AE with a slightly better $AP = 0.84$ while the training time increases substantially.

All weakly-supervised algorithms show better performance in all metrics than the unsupervised methods, which is to be expected since additional valuable information on actual anomalies is used during training. The best anomaly detection algorithm in our studies is the proposed SAE which reaches an AP score of 0.872, AUC ROC of 0.935 and F_2 of 0.823, but with a fairly high variance among our experiments. The larger variance is a direct result of the generation of the training data, since randomly selected samples of normal data are paired with only 10 randomly selected samples from anomalous dataset to produce the input pairs. This produces a large variance in the selected pairs from run to run, which explains the variance in the performance of Siamese approaches. The SAE outperforms the weakly-supervised CC by 2.8%, SVC by 19.1% and MLP by 9.2%, in terms of AP score, when trained with the same number of labeled samples. The CNN-SAE also reaches performance levels above the state-of-the-art methods and comparable to the SAE but requires a much longer training time.

In Fig. 4 we show the confusion matrices of the best performing unsupervised algorithm FF-AE (4a) and the proposed methods: unsupervised CC (4b) and weakly-supervised CC (4c), SAE (4d) and CNN-SAE (4e). With the change of the training approach, from unsupervised to weakly-supervised, the False Positive Rate (FPR) goes from 12% to less than 10%, and the False Negative Rate (FNR) is approximately halved, from 24% to 12%. The SAE as the best performing method has about 9% of FPR and 12% of FNR, which is still rather large but already at the edge of being feasible for a real-world application. Can be underlined that we only used 10 labelled anomalous samples for the weakly-supervised training. Can be reasonably expected that the FPR and FNR could be further reduced by increasing the number of labeled samples.

In Fig. 5 is depicted an example data from normal and anomaly class, as well as their respective reconstruction made with the CNN-SAE algorithm. The upper row shows a normal and anomalous data sequence which is given as input to the model. The bottom row shows the reconstruction of the convolutional network. As it can be seen, the initial targets are achieved, since the normal time series is accurately reconstructed, while the anomalous sample produces an uncorrelated output.

For the previous results, we used only 10 anomalous data samples in the weakly-supervised setting. In order to get some insights into the dependency on the number of labeled anomalous data samples, we trained the algorithms with differently sized sets of anomalies, $|\mathcal{A}| = \{5, 10, 50, 100, 500, 1000\}$, which corresponds to 0.05%, 0.11%, 0.57%, 1.14%, 5.7% and 11.44% of the normal training samples. As it is expected, there is a general trend that the performance increases with more labeled anomaly samples as observable in Fig. 6. Remarkably, the proposed Siamese approaches already give a rather high AP score above 0.85 with just 5 labeled anomaly samples. For the largest number of labeled anomalies, the proposed CC algorithm show the best performance. This is understandable, as in the clustering approach with the penalty term, accordingly to Eq. 1, the information of each anomaly is directly used. However, this is only beneficial as long as there is no noise in the set of labeled anomalies. In a real-world setting with noisy labels, the performance might not improve as much with more labeled data samples. The proposed Siamese approaches are always clearly above the state-of-the-art methods. The increase with additional labeled anomaly samples is rather slow which we take as a hint that the Siamese approaches can be operated rather robustly in a real-world setting with noisy anomaly data.

B. Results obtained with only Real-World data

For the sake of completeness, we also report results obtained by using only the unlabeled real-world data to train the algorithms. The same networks structure and hyperparameter settings as for the previous results were used in this analysis.

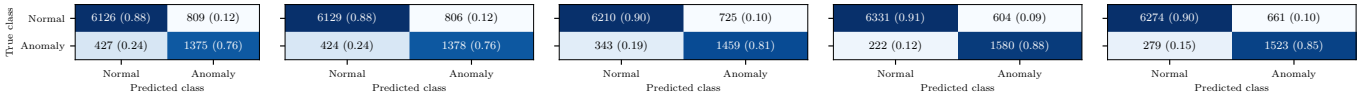
The results using multiple algorithms trained only with real-world data are reported in Table IV. Along with the performance measures of AP and ROC, we also include the percentage difference to the results using Digital Twin data ($\Delta_{AP,\%}$ and $\Delta_{ROC,\%}$). The performance of the algorithms shows the same trend as observed for the case with the Digital Twin simulation data. The best performing unsupervised

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2020.3019788, IEEE Transactions on Industrial Informatics

TABLE III
PERFORMANCE VALUES OF ALL EVALUATED ALGORITHMS, USING THE DIGITAL TWIN TO GENERATE THE TRAINING-SET. THE GLOBALLY BEST SCORES ARE HIGHLIGHTED IN BOLD, WHILE THE BEST UNSUPERVISED SCORES ARE IN ITALIC FONT. FOR THE WEAKLY SUPERVISED METHODS ONLY 10 ADDITIONAL LABELED ANOMALIES WERE USED DURING TRAINING.

Training Approach	Algorithm	Train Time	AP	AUC ROC	F_2
Unsupervised	*Mean Absolute Error (MAE)*	0.41 s	0.692 ± 0.002	0.792 ± 0.001	0.512 ± 0.001
	Isolation Forest (IF)	0.54 s	0.603 ± 0.024	0.888 ± 0.011	0.697 ± 0.026
	One-Class Support Vector Machine (OCSVM)	3.54 s	0.778 ± 0.001	0.843 ± 0.001	0.738 ± 0.001
	k-Nearest Neighbors (KNN)	0.56 s	0.807 ± 0.008	0.893 ± 0.002	0.720 ± 0.002
	Principal Component Analysis (PCA)	0.09 s	0.519 ± 0.001	0.649 ± 0.001	0.478 ± 0.001
	Local Outlier Factor (LOF)	0.18 s	0.811 ± 0.002	0.869 ± 0.001	0.711 ± 0.002
	Feed Forward Autoencoder (FF-AE)	32.4 s	$\mathit{0.842 \pm 0.007}$	$\mathit{0.914 \pm 0.006}$	$\mathit{0.746 \pm 0.013}$
	Cluster Centers (CC)	0.56 s	0.831 ± 0.001	0.909 ± 0.001	0.734 ± 0.001
Weakly-supervised	*Support Vector Classifier (SVC)*	7.8 s	0.798 ± 0.028	0.864 ± 0.024	0.490 ± 0.040
	Multi Layer Perceptron (MLP)	12.7 s	0.732 ± 0.051	0.813 ± 0.054	0.017 ± 0.051
	Cluster Centers (CC)	0.83 s	0.848 ± 0.005	0.921 ± 0.003	0.762 ± 0.005
	Siamese Autoencoder (SAE)	15.8 s	$\mathbf{0.872 \pm 0.015}$	$\mathbf{0.935 \pm 0.013}$	$\mathbf{0.823 \pm 0.017}$
	Convolutional Siamese Autoencoder (CNN-SAE)	132.6 s	0.866 ± 0.016	0.931 ± 0.014	0.798 ± 0.022



(a) FF-AE. (b) Unsupervised CC. (c) Weakly-supervised CC. (d) SAE. (e) CNN-SAE.

Fig. 4. Confusion matrices of the proposed algorithms trained with Digital Twin data: FF-AE (a), unsupervised CC (b), weakly-supervised CC (c), SAE (d), CNN-SAE (e). The numbers are the absolute detections while the numbers given in parenthesis are the in-class percentages.

TABLE IV
PERFORMANCE COMPARISON OF THE INVESTIGATED ALGORITHM TRAINED ONLY WITH REAL-WORLD DATA. THE GLOBALLY BEST SCORES ARE IN BOLD, WHILE THE BEST UNSUPERVISED SCORES ARE IN ITALIC FONT. THE $\Delta_{\%}$ IS WITH RESPECT TO THE USE OF DIGITAL TWIN DATA.

Training Approach	Algorithm	AP	$\Delta_{AP,\%}$	AUC ROC	$\Delta_{ROC,\%}$
Unsupervised	*Isolation Forest (IF)*	0.479 ± 0.008	-20.6	0.800 ± 0.009	-9.9
	One-Class Support Vector Machine (OCSVM)	0.526 ± 0.004	-32.4	0.775 ± 0.005	-8.1
	k-Nearest Neighbors (KNN)	0.260 ± 0.002	-67.8	0.491 ± 0.002	-45.0
	Principal Component Analysis (PCA)	0.507 ± 0.002	-2.3	0.614 ± 0.001	-5.4
	Local Outlier Factor (LOF)	0.295 ± 0.004	-63.6	0.496 ± 0.006	-42.9
	Feed Forward Autoencoder (FF-AE)	$\mathit{0.612 \pm 0.012}$	-27.3	$\mathit{0.822 \pm 0.012}$	-10.1
	Cluster Centers (CC)	0.468 ± 0.017	-43.7	0.686 ± 0.008	-24.5
Weakly-supervised	*Support Vector Classifier (SVC)*	0.554 ± 0.085	-30.6	0.737 ± 0.036	-14.7
	Multi Layer Perceptron (MLP)	0.711 ± 0.031	-2.9	0.767 ± 0.026	-5.7
	Cluster Centers (CC)	0.584 ± 0.033	-31.1	0.774 ± 0.023	-16.0
	Siamese Autoencoder (SAE)	$\mathbf{0.756 \pm 0.019}$	-13.3	$\mathbf{0.876 \pm 0.012}$	-6.3
	Convolutional Siamese Autoencoder (CNN-SAE)	0.589 ± 0.023	-32.0	0.789 ± 0.020	-15.3

algorithm is the FF-AE, and the best overall algorithm is the weakly-supervised SAE. However, the overall performance values are lower for all the investigated algorithms for both the metrics reported, AP and ROC. This is due to the fact that the algorithms are not trained with a dataset that reflects only the normal operation modes of the machinery, but the training data is noisy and with some anomalous samples.

VIII. FINAL REMARKS AND CONCLUSION

In this work, we present novel approaches to multi-variate time-series anomaly detection. We demonstrate the approaches at an application to real-world data from a facility monitoring system of a medium-sized company. We focus on the Combined Heat and Power (CHP) module and use a Digital Twin simulation of the facility to generate normal operation data for training. A small set of labeled anomalies from the actual monitoring system is also available to train the proposed weakly-supervised algorithms. We propose a simple clustering-based approach where the anomaly score includes a penalty term which accounts for the labeled data samples. We also present two approaches realizing Siamese Neural Networks architectures, one taking features derived from the time-series as input and the other directly operating with raw time-series data. These architectures implement autoencoder neural networks, the loss function targets the perfect reconstruction of normal operation data, while anomalous data samples should not be reconstructed well. In order to enhance the discriminatory power of the networks, the latent representations of normal and anomalous samples is forced to have large distances.

We evaluate many statistical feature types and parameters for the time-series features and observe that simple mean and

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2020.3019788, IEEE Transactions on Industrial Informatics

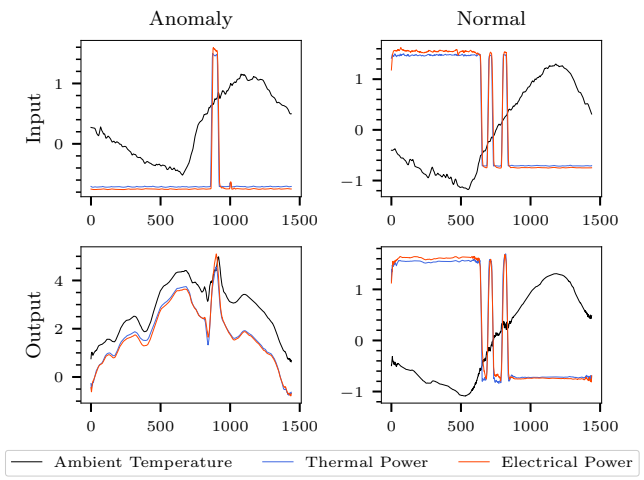


Fig. 5. Input, top row, and reconstructed sample, bottom row, of CNN-SAE. The data have been standardized to have zero mean and unitary standard deviation.

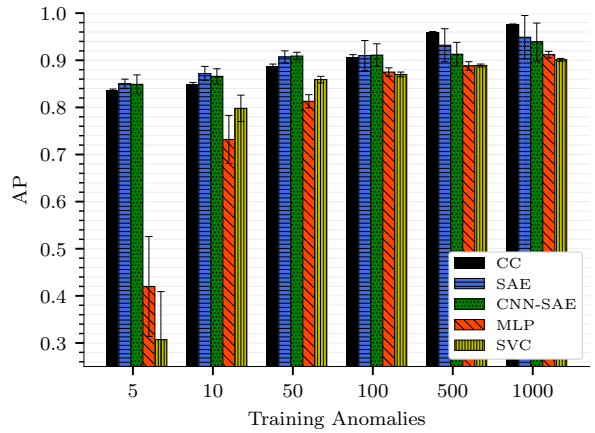


Fig. 6. Overall performances of our experiments based on varying the number of labeled anomalies.

standard deviation of a time interval of one day, with a stride of one hour, give the best results. We compare the proposed approaches to many state-of-the-art approaches and evaluate a multitude of performance measures for comparison.

We explicitly elucidate the usefulness of using a Digital Twin simulation in the context of anomaly detection. The performance is better for all tested algorithms when they are trained on normal operation data from the Digital Twin, in comparison to the cases where the algorithms are trained on real-world data only.

All the proposed weakly-supervised algorithms showed better performance that the state-of-the-art approaches according to all performance measures. The overall best performing algorithm is the Siamese Autoencoder (SAE) operating on time-series features, while the Feed Forward Autoencoder exhibits best performance of all unsupervised algorithms. When varying the number of labeled anomalous samples in the training set, the performance values of the proposed weakly-supervised approaches change mildly and still have very good performance for only five anomalous samples. In contrast, the performance strongly degrades for smaller number of samples in the training set for the Multi Layer Perceptron and the Support Vector Classifier.

The False Positive and False Negative detection rates of the best performing method, SAE, are around 9% and 12% in each class, which is acceptable but still rather large for a real-world application. However, the current approach constitutes a generic proof of concept for using Siamese Networks for anomaly detection tasks in real-world settings. Multiple options are available to improve the performance for a specific application. Including real-world normal states into the labeled dataset should enhance the performance due to improving the alignment and transfer from purely simulated Digital Twin data to real measurement data. Instead of using a fixed predefined threshold for the Anomaly Score, as done in our simulations, the threshold could be learned from the available data or adjusted by using an adaptive expert feedback.

In the present study, the set of labeled anomalies is assumed to contain no noise, i.e. no mislabeled samples. This assumption is not valid in a real-world environment, as labelling errors are to be expected, but also the definition of what constitutes an anomaly is not clear and might change over time. As a consequence, a human-in-the-loop approach with expert feedback is highly desirable where the set of labeled anomalies is re-inspected and possibly extended over time. The proposed algorithms are designed to easily incorporate the expert-feedback therein. However, their evaluation is intentionally left for future work.

REFERENCES

[1] D. Preuveneers and E. Ilie-Zudor, "The intelligent industry of the future: A survey on emerging trends, research challenges and opportunities in industry 4.0," *Journal of Ambient Intelligence and Smart Environments*, vol. 9, no. 3, pp. 287–298, 2017.

[2] T. P. Carvalho, F. A. A. M. N. Soares, R. Vita, R. da P. Francisco, J. P. Basto, and S. G. S. Alcal, "A systematic literature review of machine learning methods applied to predictive maintenance," *Computers & Industrial Engineering*, vol. 137, p. 106024, 2019.

[3] V. Chandola, A. Banerjee, and V. Kumar, "Survey of Anomaly Detection," *ACM Computing Survey (CSUR)*, vol. 41, no. 3, pp. 1–72, 2009.

[4] Y. Zhang, Z. Y. Dong, W. Kong, and K. Meng, "A composite anomaly detection system for data-driven power plant condition monitoring," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4390–4402, 2020.

[5] M. H. Chang, C. Chen, D. Das, and M. Pecht, "Anomaly detection of light-emitting diodes using the similarity-based metric test," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 3, pp. 1852–1863, 2014.

[6] D. Wu, Z. Jiang, X. Xie, X. Wei, W. Yu, and R. Li, "LSTM Learning With Bayesian and Gaussian Processing for Anomaly Detection in Industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5244–5253, 2020.

[7] R. Nawaratne, D. Alahakoon, D. De Silva, and X. Yu, "Spatiotemporal anomaly detection using deep learning for real-time video surveillance," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 393–402, 2020.

[8] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *CoRR*, vol. abs/1901.03407, 2019. [Online]. Available: http://arxiv.org/abs/1901.03407

[9] M. Grieves, "Digital twin: Manufacturing excellence through virtual factory replication," White paper, 2014.

[10] F. Tao, H. Zhang, A. Liu, and A. Y. Nee, "Digital Twin in Industry: State-of-the-Art," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2020.3019788, IEEE Transactions on Industrial Informatics

[11] G. Sugumar and A. Mathur, "Assessment of a Method for Detecting Process Anomalies Using Digital-Twinning," *Proceedings - 2019 15th European Dependable Computing Conference, EDCC 2019*, pp. 119–126, 2019.

[12] E. C. Balta, D. M. Tilbury, and K. Barton, "A digital twin framework for performance monitoring and anomaly detection in fused deposition modeling," *IEEE International Conference on Automation Science and Engineering*, vol. 2019-Augus, no. 70, pp. 823–829, 2019.

[13] P. Pileggi, J. Verriet, J. Broekhuijsen, C. Van Leeuwen, W. Wijbrandi, and M. Konsman, "A digital twin for cyber-physical energy systems," *7th Workshop on Modeling and Simulation of Cyber-Physical Energy Systems, MSCPES 2019 - Held as part of CPS Week, Proceedings*, pp. 1–6, 2019.

[14] E. Principi, D. Rossetti, S. Squartini, and F. Piazza, "Unsupervised electric motor fault detection by using deep autoencoders," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 2, pp. 441–451, 2019.

[15] J. Wu, Z. Zhao, C. Sun, R. Yan, and X. Chen, "Fault-attention generative probabilistic adversarial autoencoder for machine anomaly detection," *IEEE Transactions on Industrial Informatics*, pp. 1–10, 2020.

[16] W. Jiang, Y. Hong, B. Zhou, X. He, and C. Cheng, "A GAN-Based Anomaly Detection Approach for Imbalanced Industrial Time Series," *IEEE Access*, vol. 7, pp. 143 608–143 619, 2019.

[17] M. Canizo, I. Triguero, A. Conde, and E. Onieva, "Multi-head CNN-RNN for multi-time series anomaly detection: An industrial case study," *Neurocomputing*, vol. 363, pp. 246–260, 2019.

[18] B. Wang, Z. Li, Z. Dai, N. Lawrence, and X. Yan, "Data-driven Mode Identification and Unsupervised Fault Detection for Nonlinear Multimode Processes," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 3651–3661, 2019.

[19] P. Li, O. Niggemann, and B. Hammer, "A Geometric Approach to Clustering Based Anomaly Detection for Industrial Applications," *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, vol. 1, pp. 5345–5352, 2019.

[20] X. Li, W. Zhang, Q. Ding, and X. Li, "Diagnosing rotating machines with weakly supervised data using deep transfer learning," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1688–1697, 2020.

[21] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," in *Advances in neural information processing systems*, 1994, pp. 737–744.

[22] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2, 2015.

[23] D. Droghini, S. Squartini, E. Principi, L. Gabrielli, and F. Piazza, "Audio metric learning by using siamese autoencoders for one-shot human fall detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–11, 2019.

[24] Z. Alaverdyan, J. Jung, R. Bouet, and C. Lartizien, "Regularized siamese neural network for unsupervised outlier detection on brain multiparametric magnetic resonance imaging: Application to epilepsy lesion screening," *Medical Image Analysis*, vol. 60, p. 101618, 2020.

[25] L. V. Utkin, Y. A. Zhuk, and V. S. Zaborovsky, "An anomalous behavior detection of a robot system by using a hierarchical Siamese neural network," *Proceedings of 2017 20th IEEE International Conference on Soft Computing and Measurements, SCM 2017*, pp. 630–634, 2017.

[26] T. Schwan, S. Schmitt, and A. Castellani, "Calibration of HVAC system models with monitoring data - Digital Twin meets measurement data," in *ESI Forum in Deutschland*. ESI, 2019.

[27] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS ONE*, vol. 10, no. 3, pp. 1–21, 2015.

[28] Y. Zhao, Z. Nasrullah, and Z. Li, "Pyod: A python toolbox for scalable outlier detection," *Journal of Machine Learning Research*, vol. 20, no. 96, pp. 1–7, 2019.

[29] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.

[30] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.



Castellani Andrea was born in Civitanova Marche, Italy, on August 1993. He received the BSc and MSc degree (cum laude) in Electronic Engineering respectively in 2016 and 2019, from Polytechnic University of Marche, UnivPM, Italy. In 2020 he started his PhD period at the CITEC excellence cluster of Bielefeld University, in collaboration with Honda Research Institute, Offenbach, Germany. His current research interests include Machine Learning and time-series analysis.



Schmitt Sebastian was born in 1975 in Germany. He received his PhD in theoretical solid state physics in 2008 from Technical University of Darmstadt, Germany. After a post-doc period from 2009 to 2011 at the Technical University of Dortmund, Germany, he joined the Honda Research Institute Europe in 2012. In his role as Senior Scientist he evaluates recent technologies in optimization, data analytics and machine learning for their use in real-world engineering settings relevant to R&D departments within Honda.



Squartini Stefano (IEEE Senior Member, IEEE CIS Member) was born in Ancona, Italy, on March 1976. He got the Italian Laurea with honors in electronic engineering from Polytechnic University of Marche, UnivPM, in 2002. He obtained his PhD at the same university in 2005. He joined the Department of Information Engineering as Assistant Professor in Circuit Theory in 2007. He is now Full Professor at UnivPM since 2020. His current research interests are in the area of computational intelligence and digital signal processing, with special focus on audio processing and energy management. He is author and coauthor of more than 210 international scientific papers. He is Associate Editor of the IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Cybernetics and IEEE Transactions on Emerging Topics in Computational Intelligence. He joined the Organizing and the Technical Program Committees of more than 80 International Conferences and Workshops.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.