

Autoencoder based Anomaly Detection and Explained Fault Localization in Industrial Cooling Systems

Stephanie Holly ¹, Robin Heel ², Denis Katic ³, Leopold Schoeffl ⁴, Andreas Stiftinger ⁵, Peter Holzner ⁶, Thomas Kaufmann ⁷, Bernhard Haslhofer ⁸, Daniel Schall ⁹, Clemens Heitzinger ¹⁰, and Jana Kemnitz ¹¹

^{1,2,6,7,9,11} *Siemens Technology, Vienna, 1210, Austria*

stephanie.holly@siemens.com

robin.heel@siemens.com

peter.holzner@siemens.com

thomas.kaufmann@siemens.com

daniel.schall@siemens.com

jana.kemnitz@siemens.com

^{1,6,7,10} *Vienna University of Technology, Vienna, 1040, Austria*

clemens.heitzinger@tuwien.ac.at

^{3,8} *Austrian Institute of Technology, Vienna, 1210, Austria*

denis.katic@ait.com

bernhard.haslhofer@ait.com

^{4,5} *Hauser, Linz, 4040, Austria*

leopold.schoeffl@hauser.com

andreas.stiftinger@hauser.com

ABSTRACT

Anomaly detection in large industrial cooling systems is very challenging due to the high data dimensionality, inconsistent sensor recordings, and lack of labels. The state of the art for automated anomaly detection in these systems typically relies on expert knowledge and thresholds. However, data is viewed isolated and complex, multivariate relationships are neglected. In this work, we present an autoencoder based end-to-end workflow for anomaly detection suitable for multivariate time series data in large industrial cooling systems, including explained fault localization and root cause analysis based on expert knowledge. We identify system failures using a threshold on the total reconstruction error (autoencoder reconstruction error including all sensor signals). For fault localization, we compute the individual reconstruction error (autoencoder reconstruction error for each sensor signal) allowing us to identify the signals that contribute most to the total reconstruction error. Expert knowledge is provided via look-up table enabling root-cause analysis and assignment to

the affected subsystem. We demonstrated our findings in a cooling system unit including 34 sensors over a 8-months' time period using 4-fold cross validation approaches and automatically created labels based on thresholds provided by domain experts. Using 4-fold cross validation, we reached a F1-score of 0.56, whereas the autoencoder results showed a higher consistency score (CS of 0.92) compared to the automatically created labels (CS of 0.62) – indicating that the anomaly is recognized in a very stable manner. The automatically created labels, however, detected anomaly earlier. The main anomaly was found by the autoencoder and automatically created labels, and was also recorded in the log files. Further, the explained fault localization highlighted the most affected component for the main anomaly in a very consistent manner.

1. INTRODUCTION

Malfunctions or even a failure of refrigeration systems are a risk with very high damage potential for food wholesalers. In the course of Industry 4.0 and digitization, sensors and instrumentation drive the central forces of innovation. New potentials for monitoring and machine learning based predictive maintenance of the cold stores are opening up. However,

Stephanie Holly et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

the training and updating of such machine learning based models poses several challenges. First, damage and outage reports, which represent the required ground-truth for a supervised learning task, are not yet collected in a systematic and consistent manner. Second, a refrigerating system is a large, complex system including several hundreds of sensors presenting a widely varying data domain such as temperature, vibration or engine speed (Weerakody, Wong, Wang, & Wendell, 2021). Third, the sensors are often retrofitted or upgraded successively in the course of digitization. Therefore the question remains open how a machine learning model can be scaled from one component to an entire system or several systems.

The challenge of missing ground truth data and therefore learning useful representations with little or no supervision is a key challenge in machine learning. In the context of predictive maintenance, unsupervised learning has shown to be successful in identifying system failures without supervision. However, identifying the root cause of the failure employing unsupervised learning procedures remains an open task. For this reason we transform our unsolvable task into a closely related solvable task, which aims to achieve a similar benefit and business impact. Therefore, we have started with a requirements analysis. The outcomes of the requirements analysis showed us that the localization of the affected sensors, associated with the root cause of the detected failures are an important step. As the affected system is very complex including several hundreds of sensors, the maintenance employee can save valuable time and effort if the affected sensors and thereby the affected subsystem can be localized. Therefore we propose an algorithm for fault localization in large cooling systems with no supervision or little supervision.

- C1:** We define a real-world learning task based on industrial requirements and provide a 18 months ground truth data set for an entire cooling system unit including 34 sensor signals.
- C2:** We provide an autoencoder based anomaly detection workflow suitable for multivariate and increasingly upgraded time series data.
- C3:** Our workflow includes an algorithm for explained fault localization based on the individual reconstruction error for each sensor signal.
- C4:** Our workflow includes a root cause analysis enabled by integrated expert knowledge.
- C5:** Our workflow was compared against automatically created labels showing an F1-score of 0.56 and a consistency score of 0.92. The explained fault localization highlighted the most affected component in a very consistent manner.

2. RELATED WORK

Industrial cooling systems (ICS) are widely deployed in large supermarkets and storage warehouses to preserve per-

ishables with a global market valued to over USD 5 billion (Reportlinker, 2020). ICS are subject to faults, such as compressor failures and bearing damage, that can degrade the operational efficiency and even result in their breakdown. Accurate and timely detection of faults and degradation is critical to prevent food spoilage, customer inconvenience, maintenance costs, and other related losses. Automated fault diagnosis in ICS has been explored for many years (Grimmelius, Klein Woud, & Been, 1995), ranging from Kalman-filter based methods (Yang, Rasmussen, Kieu, & Izadi-Zamanabadi, 2011), random forest (Kulkarni, Devi, Sirighee, Hazra, & Rao, 2018) and neural network based approach. AI based predictive maintenance is estimated to decrease breakdowns by up to 70 % and lowers maintenance costs by 25 % (Deloitte, 2017) in the next years.

Prognostics and health management approaches have been studied extensively across industrial applications, such as aircraft systems (Bieber, Verhagen, & Santos, 2021), hard drives (Barelli & Ottaviani, 2021). While some traditional methods for fault detection require feature engineering (Kato, Yairi, & Hori, 2001; Su, Sun, Gao, Qiu, & Tian, 2019; J. Wang et al., 2020), recent work has shown that end-to-end autoencoders can outperform traditional approaches (Zong et al., 2018; Maleki, Maleki, & Jennings, 2021; Heistracher, Jalali, Suendermann, et al., 2021). Autoencoders are promising and have been for minimal-configuration fault detection (Heistracher, Jalali, Suendermann, et al., 2021; Hood et al., 2021), however this unsupervised methods lack the ability for fault classification or fault location.

Fault classification is typically based on supervised learning (Ismail Fawaz, Forestier, Weber, Idoumghar, & Muller, 2019; Dempster, Petitjean, & Webb, 2020; Z. Wang, Yan, & Oates, 2017; Karim, Majumdar, Darabi, & Chen, 2018) and required a large number of ground truth data, often lacking in industrial applications. Furthermore, these supervised approaches are often difficult to scale and difficult to transfer to other, similar systems (Kemnitz, Bierweiler, Grieb, von Dosky, & Schall, 2021; Heistracher, Jalali, Strobl, et al., 2021). As the affected system is very complex including several hundreds of sensors, the maintenance employee can save valuable time and effort if the affected sensors and thereby the affected subsystem can be localized. Further, explaining and quantifying the individual contribution helps increase trust and interpretability (Grezmak, Wang, Sun, & Gao, 2019). The knowledge about individual localization and contribution can be combined with expert knowledge and thereby enables root cause analysis.

Therefore we propose an end-to-end autoencoder based workflow for fault localization in large cooling systems with no supervision or little supervision. This proposed workflow is inspired by heat and saliency maps (Simonyan, Vedaldi, & Zisserman, 2014) applied in computer vision (Goebel et al., 2018).

3. INDUSTRIAL REQUIREMENTS

Hauser aims to build a machine learning based monitoring, alarm, and remote maintenance systems for industrial refrigeration systems, which are deployed in hundreds of locations around the world. Since many of these systems are of the same type, a model-driven approach that could predict damages or outages to cold storage’s, would scale from a business perspective and could also be offered as a service to customers. The following requirements result from this vision:

- R1: Employable without ground truth data
- R2: Scalable to similar refrigeration systems
- R3: A holistic approach employing a cockpit view
- R4: Employable for fault detection
- R5: Employable for fault localization
- R6: Data agnostic
- R7: Dealing with retrofitted or upgraded sensors

It is important to start with an approach that is expandable. Each step should add substantial business value.

4. MACHINE LEARNING WORKFLOW

4.1. Overall Workflow

We propose a workflow for preprocessing, anomaly detection, explained fault localization and root cause identification of multivariate time series data, see Fig. 1. For anomaly detection, we use a LSTM autoencoder. We identify system failures using a threshold on the total reconstruction error $RE_{total}(S)$ Eq. (7) of sensor signals $S := [S_1, \dots, S_n]$. For fault localization, we compute the individual reconstruction error $RE_{ind}(S_i)$ Eq. (9) for each sensor signal S_i allowing us to identify the signals that contribute the most to the total reconstruction error $RE_{total}(S)$. Provided an expert knowledge "look-up table", we can thus locate the affected subsystem in the cooling system.

4.2. Preprocessing

Data preprocessing is a crucial task in machine learning pipelines (Leukel, González, & Riekert, 2021). Here, data preprocessing denotes the process of preparing raw data for the machine learning model including data cleaning, signal selection, resampling, missing values treatment and data normalization. In industrial systems, data preprocessing is a major challenge due to its high dimensionality and complex structure (Bekar, Nyqvist, & Skoogh, 2020). In industrial cooling systems, we find numerous components with various data sources including several hundreds of sensors, a widely varying data domain, and retrofitted and upgraded sensors requiring distinct approaches in the mentioned preprocessing steps.

In the following, we want to present our approach to data preprocessing in an industrial cooling system with time series

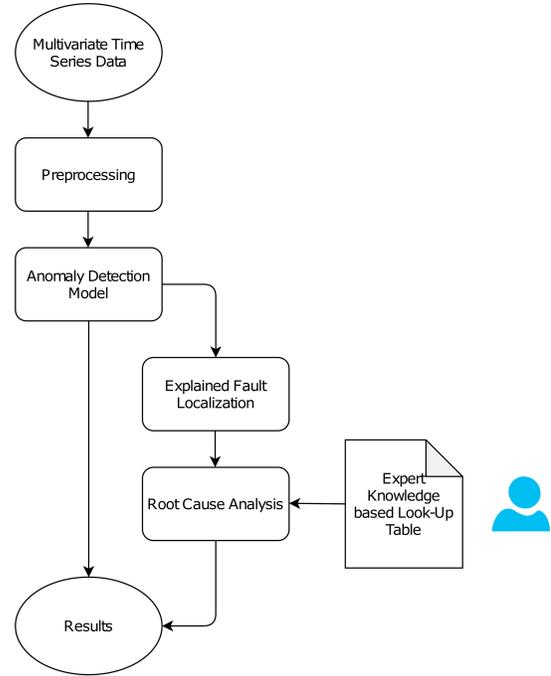


Figure 1. Overall workflow: preprocessing, anomaly detection, explained fault localization, and root cause analysis.

data. Let S_i denote the i^{th} sensor signal in the cooling system, $1 \leq i \leq n$. The data is acquired without a defined frequency, that is, the data is not available at the same frequency and each sensor signal S_i is recorded with an individual frequency or irregularly.

4.2.1. Data Cleaning

In the course of digitization, sensors are often retrofitted or upgraded successively. We therefore removed early time periods lacking sensor signals considered important by Hauser experts.

4.2.2. Signal Selection

The industrial cooling system consists of numerous components with a wealth of data sources providing a huge amount of sensor signals. The massive amount of data is a key challenge in data preprocessing (Bekar et al., 2020). In order to decrease the number of signals, we calculated the correlation coefficient and removed highly correlated signals (Nahian et al., 2021). Let X and Y denote random variables with covariance $cov(X, Y)$ and standard deviation σ_X, σ_Y respectively. Then, the correlation coefficient $\rho_{X,Y}$ is given by

$$\rho_{X,Y} := \frac{cov(X, Y)}{\sigma_X \sigma_Y}. \tag{1}$$

Extracting information from the timestamp can increase the quality of prediction models (Latyshev, 2018). Thus, we

added the signals month, time (hour) and weekday to our observations.

4.2.3. Resampling

The acquired data consists of irregularly sampled time series data. With the growth of multi-sensor systems, the preprocessing of irregular time series data is becoming increasingly important (Weerakody et al., 2021). Due to numerous components with various data sources including several hundreds of sensors and a widely varying data domain, we cannot expect to find all sensor signals sampled at a constant sampling rate with common timestamps. For modeling, however, we need to resample the data to a regular frequency. Let x_t^i denote an observed value of sensor signal S_i at time t . Let r denote a regular sampling rate. We then obtain new timestamps \mathcal{T} by equidistant time intervals of length r . Depending on the signal type, we compute a new value \hat{x}_t^i of signal S_i at timestamp $t \in \mathcal{T}$. In general, the acquired sensor signals represent numerical values, e.g. measuring temperature, vibration or engine speed. However, the cooling system also includes signals of boolean values giving information about the system's health and up-counting signals giving information about the pause time of components in the system. When the value is counting, the minimum value is the most representative one, as it was the initial position of the counter. And using the maximum boolean value, was the same result as using the logical OR, which means that it is zero if and only if all samples of the corresponding interval are zero. In the general case, we simply average over all observed values x_u^i in the interval $[t, t+r)$, that is,

$$\hat{x}_t^i := \frac{1}{\#\{x_u^i | u \in [t, t+r)\}} \sum_{u \in [t, t+r)} x_u^i. \quad (2)$$

In the case of boolean values $x^i \in \{0, 1\}$, we define

$$\hat{x}_t^i := \max_{u \in [t, t+r)} x_u^i, \quad (3)$$

and in the case of constantly (by $c \in \mathbb{N}$) increasing values $x^i \in \{k+c | k \in \mathbb{N}\}$, we define

$$\hat{x}_t^i := \min_{u \in [t, t+r)} x_u^i. \quad (4)$$

4.2.4. Missing Values Treatment

Resampling irregularly sampled time series data will result in missing values for one or more sensor signals S_i at a given timestamp $t \in \mathcal{T}$. Simple statistical techniques include forward-filling and zero imputation (Weerakody et al., 2021). We used the fill-forward method for missing values. Before the first appearance of a value, we initialized a default value by computing the median of all available values.

4.2.5. Feature Representation

In contrast to the acquired signals in the cooling system, let features define the representation of data feed into the machine learning model. Let w denote the window size. Applying a window of size w to the sensor signals S_i at timestamp t , the features F^t at timestamp t are given by

$$F^t := \begin{pmatrix} \hat{x}_t^1 & \hat{x}_t^2 & \dots & \hat{x}_t^n \\ \hat{x}_{t+1}^1 & \hat{x}_{t+1}^2 & \dots & \hat{x}_{t+1}^n \\ \vdots & \vdots & & \vdots \\ \hat{x}_{t+w-1}^1 & \hat{x}_{t+w-1}^2 & \dots & \hat{x}_{t+w-1}^n \end{pmatrix} \in \mathbb{R}^{w \times n}. \quad (5)$$

For modeling, we reshaped the features and obtained

$$F^t = \begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ F_{wn} \end{pmatrix} \in \mathbb{R}^{wn}, \quad (6)$$

where F_j with $j = iw + k$ corresponds to \hat{x}_{t+k}^i .

4.2.6. Data Normalization

Using min-max normalization, we standardized the features by scaling each feature individually to the range $(0, 1)$. The training data is scaled, and then the scaling parameters are applied to the test data. In the domain of machine learning, normalization plays a key role in the preprocessing of data including variables of different scale. In normalization, each variable is scaled individually to the range $(0, 1)$ avoiding a variable dominating the machine learning model.

4.3. Anomaly Detection Model

We applied a LSTM autoencoder with the following settings: an input layer of size 1×370 (37 signals and a window of size 10), the first encoding layer with output size 1×370 , the second encoding layer with output size 1×185 , a repeat vector with output size 1×185 , the first decoding layer with output size 185, the second decoding layer with output size 370, and a time distributed layer of size 1×370 , see Fig. 2.

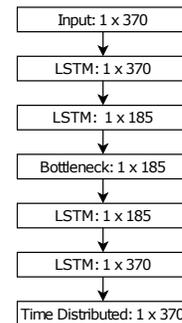
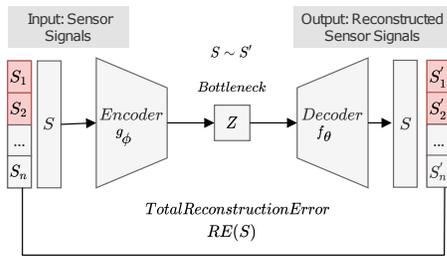


Figure 2. Model Architecture

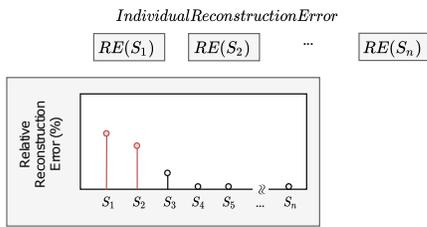
For hyper-parameter tuning, we used Bayesian optimization. Turner et al. (Turner et al., 2020) demonstrated decisively the benefits of Bayesian optimization over random search and grid search for tuning hyperparameters of machine learning models. Bayesian optimization benefits from previous evaluations of hyper-parameter configurations by including past hyper-parameter configurations in the decision of choosing the next hyper-parameter configuration. Therefore, it avoids unnecessary evaluations of the expensive objective function and requires fewer iterations to find the best hyper-parameter configuration.

We searched for the sampling rate, number of layers, dropout rate, activation function, optimizer, learning rate and batch size. We obtained a sampling rate of 60 seconds, 2 autoencoder layers, a dropout rate of 0.2, the tanh activation function, the rmsprop optimizer, a learning rate of 0.001, and a batch size of 16.

① Anomaly Detection Model



② Explained Fault Localization



③ Root Cause Analysis

Expert Knowledgebased Look - Up Table

Sensor	Component	Failure Type
Sensor 1	Component 1	1
Sensor 2	Component 1	1
Sensor 3	Component 2	2

← Domain Expert

Figure 3. Graphical Abstract of Paper, including 1) anomaly detection (identify system failures using a threshold on the total reconstruction error of all sensor signals), 2) explained fault localization (compute the the individual reconstruction error for each sensor signal and identify affected signals), 3) and root cause analysis (locate the affected subsystem based on a look-up table and the affected signals).

4.4. Explained Fault Localization

We propose an end-to-end autoencoder based workflow for fault localization in a large cooling system not providing ground truth data. The total reconstruction error is given by Eq. (7). Motivated by heat and saliency maps (Simonyan et al., 2014) applied in computer vision (Goebel et al., 2018), we derive from the individual reconstruction error Eq. (9) how much each sensor signal S_i contributes to the total reconstruction error Eq. (7). Thus the individual reconstruction error Eq. (9) can be used to identify affected signals and thereby locate the affected subsystem.

The total reconstruction error of feature F^t at timestamp t is given by

$$RE_{total}(F^t) := \frac{1}{wn} \sum_{j=1}^{wn} (F_j - F'_j)^2, \quad (7)$$

where

$$F'^t := \begin{pmatrix} F'_1 \\ F'_2 \\ \vdots \\ F'_{wn} \end{pmatrix} \in \mathbb{R}^{wn} \quad (8)$$

is the prediction of feature F^t at timestamp t .

We define the individual reconstruction error of signal S_i , $1 \leq i \leq n$, at timestamp t by

$$RE_{ind}(S_i) := \frac{1}{w} \sum_{k=0}^{w-1} (\hat{x}_{t+k}^i - \hat{x}'_{t+k}{}^i)^2, \quad (9)$$

where \hat{x}_{t+k}^i and $\hat{x}'_{t+k}{}^i$ correspond to F_j and F'_j with $j = iw+k$ respectively.

In a k -fold cross validation approach, we compute for each test dataset l , $1 \leq l \leq k$, a threshold

$$T_l := \mu_l + c\sigma_l, \quad (10)$$

where μ_l and σ_l are the mean and the standard deviation of the total reconstruction error RE_{total} on the remaining training data and c is a constant found by plotting ROC curves, see Fig. 5.

In algorithm 1, we give the pseudocode for the Fault Localization algorithm. The algorithm takes the threshold T , the feature F^t at timestamp t , the predicted feature F'^t at timestamp t and the number of significant signals m , $1 \leq m \leq n$, as an argument, and returns either the m most significant signals S^* or the empty set. We compute the total reconstruction error RE_{total} of feature F^t at timestamp t based on Eq. (7). If the total reconstruction error RE_{total} exceeds the threshold T , we determine the m most significant signals

$S^* \subset \{S_1, S_2, \dots, S_n\}$. Therefore, we compute the individual reconstruction error $RE_{\text{ind}}[i]$ based on Eq. (9), $i = 1 \dots n$. Then, we select iteratively the signal idx that yields the highest individual reconstruction error $RE_{\text{ind}}[idx]$, append signal idx to S^* and remove the selected signal from further calculations by setting its value to zero. The method APPEND takes a set and an element as an argument and appends the element to the set. When the iteration terminates, set S^* contains exactly the m most significant signals, that is, the signals with the highest individual reconstruction error. If the total reconstruction error RE_{total} does not exceed the threshold T , S^* is the empty set. Finally, the set S^* is returned.

Algorithm 1: Fault Localization

Input: threshold T , feature F^t at timestamp t , predicted feature F'^t at timestamp t , number of significant signals m
Output: significant signals $S^* \subset \{S_1^1, S_1^2, \dots, S_1^n\}$ with $\#S^* = m$
 $RE_{\text{total}} := \frac{1}{wn} \sum_{j=1}^{wn} (F_j - F'_j)^2$
if $RE_{\text{total}} > T$ **then**
 $RE_{\text{ind}}[i] := \frac{1}{w} \sum_{k=0}^{w-1} (\hat{x}_{t+k}^i - \hat{x}'_{t+k}^i)^2 \quad i = 1, \dots, n$
 $S^* := \emptyset$
 for $i = 1, \dots, m$ **do**
 $idx := \arg \max_{i=1, \dots, n} RE_{\text{ind}}[i]$
 APPEND(S^*, idx)
 $RE_{\text{ind}}[idx] := 0$
 end
else
 $S^* := \emptyset$
end
 return S^*

4.5. Root Cause Analysis and Integrated Expert Knowledge

The proposed workflow for fault detection and root cause identification – the identification of the affected sensor signals S_i – allows us to locate the affected subsystem. Each sensor signal S_i is assigned to a component in the cooling system. The component can then be used to determine the root-cause and failure type. Domain knowledge and physical connections are stored in the system using a look-up table. Over the years, the experts have collected which sensors are typically associated with a root-cause. In our case, a physics-aware look up table, see Fig. 3, was created by three domain experts with several years of maintenance experience. While all previous steps in the workflow are fully automatic - and salable to other components - the look-up table will remain component specific and will always require a manual step.

5. EVALUATION

5.1. Data Set Description

The acquired data consists of 34 irregularly sampled sensor signals in an industrial cooling system including numerous

components with various data sources and a widely varying data domain, measured over a period of 18 months. The cooling system is divided into numerous components, and each of them then again divided into several sub-components. Each sensor signal is assigned to a sub-component in the cooling system. We added 3 additional timestamp signals including month, time (hour) and weekday to our observations resulting in a total of 37 signals. By resampling the data to a regular frequency of 60 seconds, we obtained a time series dataset of 37-dimensional data samples. In general, the acquired sensor signals represent numerical values, e.g. measuring temperature, vibration or engine speed. However, the cooling system also includes signals of boolean values giving information about the system's health and up-counting signals giving information about the pause time of components in the system.

In the course of digitization, sensors are often retrofitted or upgraded successively and additional sensors are integrated in the system posing challenges in the preprocessing and updating of the machine learning model. In general, the number of measured data points greatly increases over the measurement period of 18 months for each sensor signal. For missing signals, we computed the mean and standard deviation and sampled from the corresponding normal distribution.

The dataset will be made public available, however, it will be anonymized due to privacy reasons.

5.2. Experimental Setup

Due to the steady increase in the number of data points in the system over 18 months, data acquired in the first 10 months is not representative and thus inadequate for testing. Therefore, we restricted our test data to the last 8 months. We evaluated the machine learning workflow in two different cross validation approaches. In scenario 1, we performed a conventional 4-fold cross validation procedure on the last 8 months (datasets 7–10). In scenario 2, we performed a 4-fold cross validation approach on the last 8 months (datasets 7–10), including a basic training dataset consisting of the first 10 months. We partitioned the data of the last 8 months into 4 equally sized folds, each fold receiving 54 days of acquired data. For each fold k , we performed the following steps: Fold k is held out for testing, and the remaining 3 folds are used for training, in scenario 2 the training data includes the basic dataset. The training data is preprocessed and prepared for modeling using the preprocessing steps described in section 4.2. We then train the LSTM autoencoder described in section 4.3 for 50 epochs with early-stopping on the training data. Finally, we test the autoencoder on the held-out test data.

Table 1. Organization of training and testing dataset in scenario 1 and 2 where 0 refers to the training dataset and 1 refers to the testing dataset

round	scenario 1				scenario 2			
	1	2	3	4	1	2	3	4
dataset 1					0	0	0	0
dataset 2					0	0	0	0
dataset 3					0	0	0	0
dataset 4					0	0	0	0
dataset 5					0	0	0	0
dataset 6					0	0	0	0
dataset 7	0	0	0	1	0	0	0	1
dataset 8	0	0	1	0	0	0	1	0
dataset 9	0	1	0	0	0	1	0	0
dataset 10	1	0	0	0	1	0	0	0

5.3. Ground Truth provided by Automatically Created Labels

We tested our proposed workflow for anomaly detection, fault localization and root cause analysis with thresholds based on expert knowledge, derived from PLC-system, failure log files, system sheets and documentation. A programmable logic controller (PLC) is a device used to control a machine or industrial system. The thresholds have been repeatedly evaluated in extensive feedback discussions with several domain experts and statistically confirmed. In case of absence of expert knowledge, we derived thresholds based on the 98% confidence interval. We obtained for each signal a threshold allowing us to provide automatically created labels. In the preprocessing procedure, after data cleaning, signal selection, resampling, missing values treatment but before data normalization, we compared the input features with the derived thresholds and obtained for each timestamp and signal a label (healthy – 0, anomalous – 1) enabling us to define a label for each sample

$$F^t = \begin{pmatrix} \hat{x}_t^1 & \hat{x}_t^2 & \dots & \hat{x}_t^{37} \\ \hat{x}_{t+1}^1 & \hat{x}_{t+1}^2 & \dots & \hat{x}_{t+1}^{37} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{x}_{t+9}^1 & \hat{x}_{t+9}^2 & \dots & \hat{x}_{t+9}^{37} \end{pmatrix} \in \mathbb{R}^{10 \times 37}. \quad (11)$$

We called a timestamp $t + j$, $0 \leq j \leq 9$, anomalous if at least 10 signals of all 37 signals were anomalous at that timestamp, that is, at least 10 of all values $\hat{x}_{t+j}^1, \hat{x}_{t+j}^2, \dots, \hat{x}_{t+j}^{37}$ exceed their thresholds. Finally, we called a sample F_t with window size $w := 10$ anomalous if $t + j$ was an anomalous timestamp for all $0 \leq j \leq 9$. Then, we applied a smoothing filter to the labels, and labelled a sample F_t anomalous if the smoothed value was greater or equal to 0.5. Fig. 4 shows the labels of data over a time period of 8 months (datasets 7–10 in Fig. 7 and Fig. 8) and compares the results of the corresponding models to the automatically created labels (ground truth).

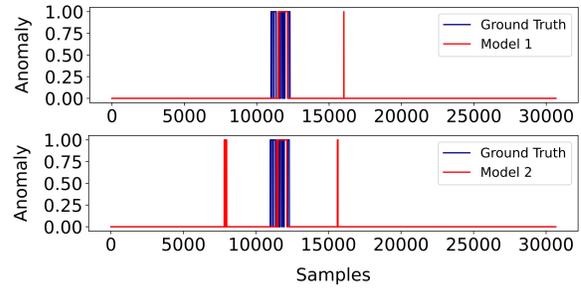


Figure 4. Comparison of anomalies (healthy – 0, anomalous – 1) defined by ground truth or found by model over a time period of 8 months, with models trained in scenario 1 and 2

5.4. Evaluation Metrics

In order to validate our model performance, we computed evaluation metrics including the F1 score, precision and recall for both scenarios, see Table 2. Further, we included ROC curves plotting the false-positive rate against the true-positive rate in several threshold settings, see Fig. 5. The true-positive rate is also known as sensitivity or recall. The ROC curves also helped us find the thresholds Eq. (10). Inspired by the imaging domain, we used the Jaccard index to measure the similarity of machine learning and threshold derived anomalies. The Jaccard index measures the similarity of two datasets comparing their intersection and union.

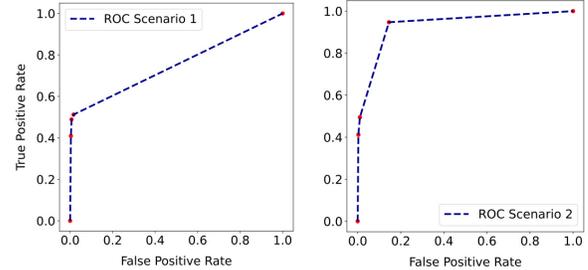


Figure 5. ROC Curves based on 3 different thresholds for each scenario respectively

Table 2. Evaluation Metrics of Scenario 1 and Scenario 2

	Scenario 1	Scenario 2	Ground Truth
f1 score	0.562	0.527	
precision	0.661	0.564	
recall	0.489	0.495	
jaccard index	0.391	0.358	
consistency score	0.920	0.773	0.619

Fig. 6 shows a part of Fig. 4. We use automatically created labels as ground truth references, however, these labels are also affected by precision and recall errors. We observe that the labels are not consistent in the occurrence of an anomaly. In order to measure the reliability of the system, we define a

consistency score indicating the consistency of an anomaly over time, see Table 2. It is based on the assumption that, in the real-world, errors often persist consistently over a longer period of time.

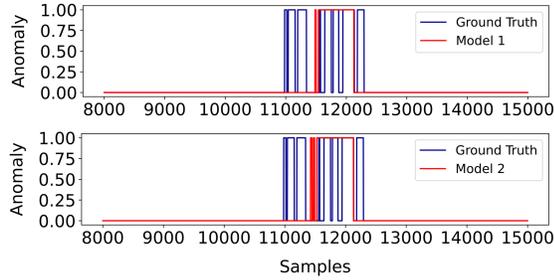


Figure 6. Comparison of an anomaly (healthy – 0, anomalous – 1) defined by ground truth or found by model in dataset 8 (54 days), with a model trained in scenario 1 and 2

Assuming an anomaly a over a time period of several closed intervals $[s_1^a, e_1^a], [s_2^a, e_2^a], \dots, [s_N^a, e_N^a]$, motivated by Fig. 6, we define the consistency score κ_a for an anomaly a as

$$\kappa_a := \frac{1}{e_N^a - s_1^a} \sum_{j=1}^N (e_j^a - s_j^a). \quad (12)$$

Further, we define the consistency score κ for a model as

$$\kappa := \frac{1}{A} \sum_a (e_N^a - s_1^a) \kappa_a \quad (13)$$

where $A := \sum_a (e_N^a - s_1^a)$.

5.5. Experimental Results

We tested our workflow for anomaly detection, fault localization and root cause analysis described in section 4.4 and 4.5 in scenario 1 and scenario 2. Scenario 1 refers to the conventional 4-fold cross validation procedure on the last 8 months (datasets 7–10). Scenario 2 refers to the 4-fold cross validation approach on the last 8 months (datasets 7–10) including a basic training dataset consisting of the first 10 months. Fig. 7 shows the results for anomaly detection in scenario 1. Fig. 8 shows the results for anomaly detection in scenario 2. The plot shows the total reconstruction error Eq. (7) and threshold computed according to Eq. (10) on the respective test dataset over a period of 54 days. Setting the sampling rate $r := 60$ (seconds) and the window size $w := 10$, we obtain 7776 ($6 \times 24 \times 54$) data samples over a period of 54 days. In both scenarios, the autoencoder detected an anomaly in dataset 8 around data sample 12000 and an anomaly in dataset 9 around data sample 16000. However, in scenario 2, the autoencoder, trained on the training data including the basic dataset, detected the anomaly in dataset 9 slightly earlier than in scenario 1. Therefore, we suggest that including the data acquired in the first 10 months

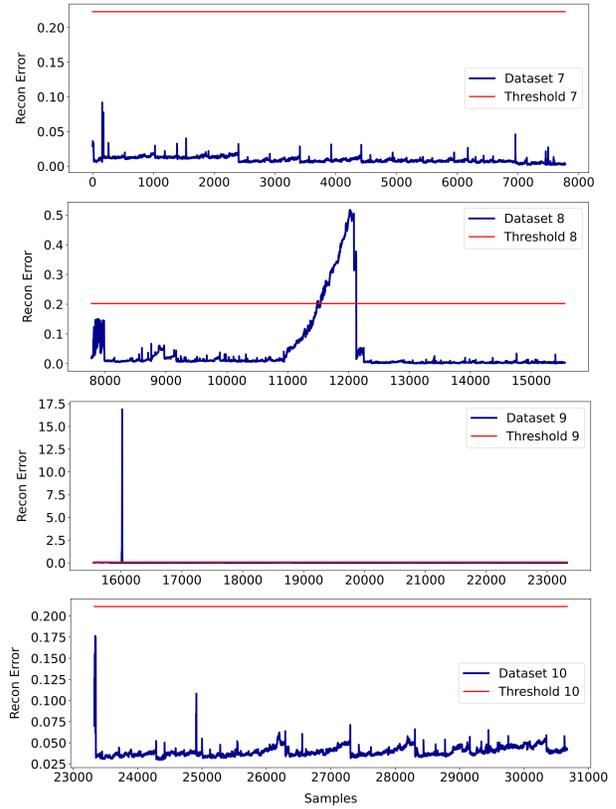


Figure 7. Anomaly Detection in Scenario 1: total reconstruction error on test datasets 7–10, each dataset of 1.8 months (54 days), with respective threshold

increases the performance of the autoencoder.

Fig. 9 shows the results for fault localization in scenario 2 on dataset 8. Fig. 10 shows the results for fault localization in scenario 2 on dataset 9. The plot shows the individual reconstruction error Eq. (9) of an anomalous data sample for all 37 signals, absolutely and relatively with the respective threshold Eq. 10. The color indicates if the individual reconstruction error of signal S_i exceeds the threshold.

Table 3. Root Cause Analysis Scenario 2: component and failure type of anomalous signals of anomalous data sample in dataset 8

RE Contribution (%)	Sensor	Component	Failure Type
11.94	Sensor 7	Component 3	2
48.86	Sensor 14	Component 1	1
8.91	Sensor 22	Component 1	1
9.33	Sensor 36	Component 13	5

Please note that the data is anonymized due to privacy reasons.

Table 3 shows the results for root cause analysis in scenario 2 on dataset 8. Table 4 shows the results for root cause analysis in scenario 2 on dataset 9. The explained fault localization and root cause analysis highlighted the most affected component

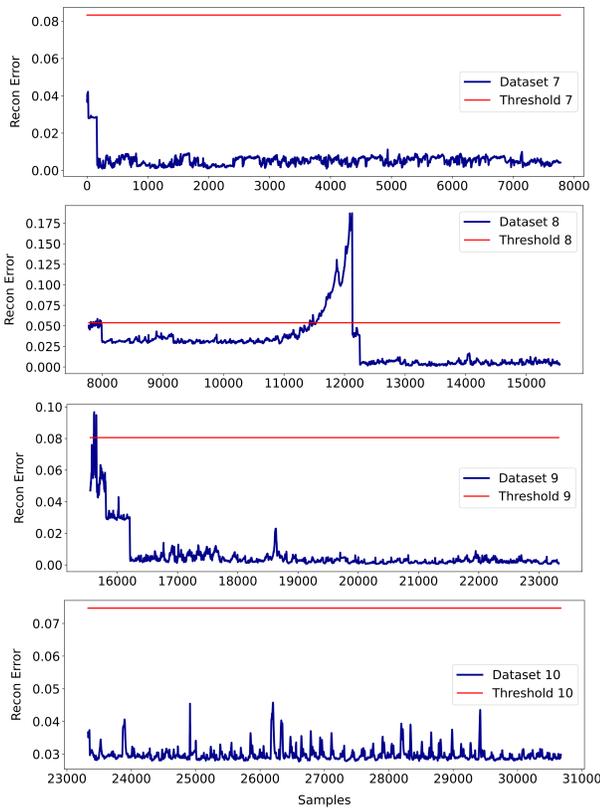


Figure 8. Anomaly Detection in Scenario 2: total reconstruction error on test datasets 7–10, each dataset of 1.8 months (54 days), with respective threshold

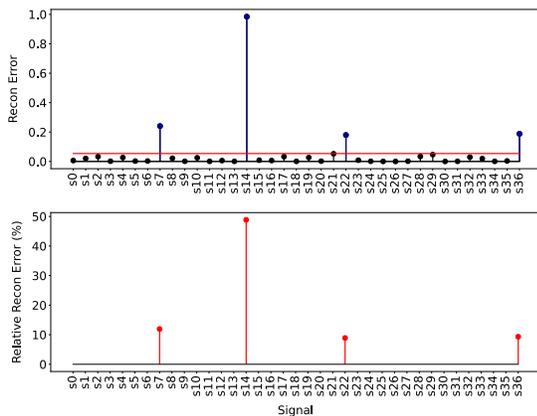


Figure 9. Explained Fault Localization in Scenario 2: individual reconstruction error of anomalous data sample in dataset 8

in a consistent manner.

6. CONCLUSION AND FUTURE WORK

We provided a 18 months dataset of multivariate time series data for an industrial cooling system including 34 sensor signals and automatically created labels based on thresholds de-

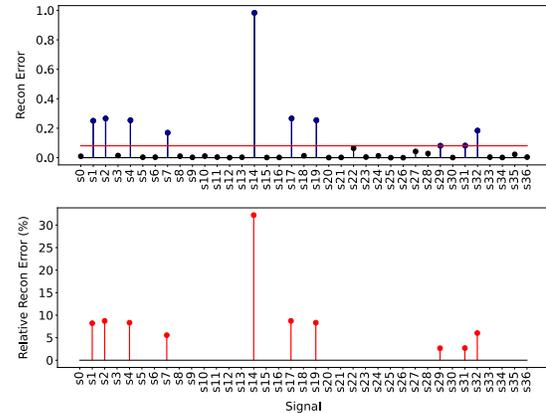


Figure 10. Explained Fault Localization in Scenario 2: individual reconstruction error of anomalous data sample in dataset 9

Table 4. Root Cause Analysis Scenario 2: component and failure type of anomalous signals of anomalous data sample in dataset 9

RE Contribution (%)	Sensor	Component	Failure Type
8.21	Sensor 1	Component 1	1
8.71	Sensor 2	Component 1	1
8.33	Sensor 4	Component 1	1
5.55	Sensor 7	Component 3	2
32.22	Sensor 14	Component 1	1
8.73	Sensor 17	Component 1	1
8.33	Sensor 19	Component 1	1
2.66	Sensor 29	Component 10	3
2.69	Sensor 31	Component 12	3
6.03	Sensor 32	Component 12	3

Please note that the data is anonymized due to privacy reasons.

rived from expert knowledge and PLC-system. We presented our machine learning workflow for anomaly detection and explained fault localization suitable for multivariate and increasingly upgraded time series data. Therefore, we presented our preprocessing steps and provided an algorithm for explained fault localization and a root cause analysis enabled by integrated expert knowledge.

We performed a conventional 4-fold cross validation approach over a time period of 8 months and a 4-fold cross validation approach including a basic dataset of 10 months and compared the model results to automatically created labels based on thresholds provided by domain experts. Using 4-fold cross validation, we reached a F1-score of 0.56, whereas the model results showed a higher consistency score (CS of 0.92) compared to the automatically created labels (CS of 0.62) – indicating that the anomaly is recognized in a very stable manner. The automatically created labels, however, detected anomaly earlier. The main anomaly was found by the model and ground truth, and was also recorded in the log files. Further, the explained fault localization highlighted the most affected component for the main anomaly in a very consistent

manner.

A limitation of this work is the comparison of our model results to automatically created labels as ground truth references. These labels are also affected by precision and recall errors. Still, automatically created labels were the best available reference for this work and are frequently considered ground truth for other real-world applications. However, a strength of our study is that we also created and provided consistency scores indicating the consistency of an anomaly over time for autoencoder results and automatically created labels. We believe that this score will also be helpful in the future for evaluation using real-world data missing ground truth. It is based on the assumption that, in the real-world, errors often persist consistently over a longer period of time.

In the future, we would like to further expand root cause analysis and increase the transparency of our proposed workflow. Further, we aim to investigate whether the increase in reconstruction error and exceeding the anomaly thresholds can be predicted for the user. A further step is to bring the developed end-to-end model into a productive environment.

Our work shows that scalable anomaly detection based on AI and explained fault localization is feasible for multivariate and increasingly upgraded time series data. Our proposed workflow provides a satisfying performance regarding the F1 score. We were also able to show that a root cause analysis enabled by integrated expert knowledge can be carried out without supervised learning highlighting the most affected component in a consistent manner. The integrated expert knowledge enabled ground truth references. We are sure that our results can also be transferred to other applications and will enable the monitoring of large systems in the future.

DATA AVAILABILITY STATEMENT

The dataset will be shared upon reasonable request. Please contact the senior author of the paper. Please cite this paper if you are using the dataset.

ACKNOWLEDGEMENTS

We thank the Hauser experts their for the valuable contribution of domain knowledge, for their time and helpful discussions.

REFERENCES

Barelli, E., & Ottaviani, E. (2021). Unsupervised anomaly detection for hard drives. *Proceedings of the European Conference of the PHM Society 2021*, 6(1), 7. doi: <https://doi.org/10.36001/phme.2021.v6i1.2795>

Bekar, E. T., Nyqvist, P., & Skoogh, A. (2020). An intelligent approach for data pre-processing and analysis in predictive maintenance with an industrial case study. *Advances in Mechanical Engineering*, 12. doi:

10.1177/1687814020919207

Bieber, M., Verhagen, W. J., & Santos, B. F. (2021). An adaptive framework for remaining useful life predictions of aircraft systems. *Proceedings of the European Conference of the PHM Society 2021*, 6(1), 11. doi: <https://doi.org/10.36001/phme.2021.v6i1.2868>

Deloitte. (2017). Predictive maintenance – taking pro-active measures based on advanced data analytics to predict and avoid machine failure. *Deloitte*.

Dempster, A., Petitjean, F., & Webb, G. I. (2020). Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5), 1454–1495. doi: 10.1007/s10618-020-00701-z

Goebel, R., Chander, A., Holzinger, K., Lecue, F., Akata, Z., Stumpf, S., ... Holzinger, A. (2018). Explainable AI: The new 42? *Machine Learning and Knowledge Extraction*, 295–303.

Grezmak, J., Wang, P., Sun, C., & Gao, R. X. (2019). Explainable convolutional neural network for gearbox fault diagnosis. *Procedia CIRP*, 80, 476-481.

Grimmelius, H., Klein Woud, J., & Been, G. (1995). On-line failure diagnosis for compression refrigeration plants. *International Journal of Refrigeration*, 18(1), 31–41. doi: [https://doi.org/10.1016/0140-7007\(94\)P3709-A](https://doi.org/10.1016/0140-7007(94)P3709-A)

Heistracher, C., Jalali, A., Strobl, I., Suendermann, A., Meixner, S., Holly, S., ... Kemnitz, J. (2021). Transfer learning strategies for anomaly detection in IoT vibration data. *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*, 1-6. doi: 10.1109/IECON48115.2021.9589185

Heistracher, C., Jalali, A., Suendermann, A., Meixner, S., Schall, D., Haslhofer, B., & Kemnitz, J. (2021). Minimal-configuration anomaly detection for IIoT sensors. *arXiv: 2110.04049*.

Hood, A., Valant, C., Horney, P., Jones, A., Lantner, J. S., Martuscello, J., & Nenadic, N. (2021). Autoencoder based anomaly detector for gear tooth bending fatigue cracks. *Proceedings of the Annual Conference of the PHM Society 2021*, 13, 10.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4), 917–963. doi: 10.1007/s10618-019-00619-1

Karim, F., Majumdar, S., Darabi, H., & Chen, S. (2018). Lstm fully convolutional networks for time series classification. *IEEE Access*, 6, 1662–1669. doi: 10.1109/ACCESS.2017.2779939

Kato, Y., Yairi, T., & Hori, K. (2001). Integrating data mining techniques and design information management for failure prevention. *Annual Conference of the Japanese Society for Artificial Intelligence*, 475–480.

Kemnitz, J., Bierweiler, T., Grieb, H., von Dosky, S., & Schall,

- D. (2021). Towards robust and transferable IIoT sensor based anomaly classification using artificial intelligence. *arXiv:2110.03440*.
- Kulkarni, K., Devi, U., Sirighee, A., Hazra, J., & Rao, P. (2018). Predictive maintenance for supermarket refrigeration systems using only case temperature data. *2018 Annual American Control Conference (ACC)*, 4640–4645. doi: 10.23919/ACC.2018.8431901
- Latyshev, E. (2018). Sensor data preprocessing, feature engineering and equipment remaining lifetime forecasting for predictive maintenance. *DAMDID/RCDL*.
- Leukel, J., González, J., & Riekert, M. (2021). Adoption of machine learning technology for failure prediction in industrial maintenance: A systematic review. *Journal of Manufacturing Systems*, 61, 87–96. doi: 10.1016/j.jmsy.2021.08.012
- Maleki, S., Maleki, S., & Jennings, N. R. (2021). Unsupervised anomaly detection with lstm autoencoders using statistical data-filtering. *Applied Soft Computing*, 108, 107443. doi: <https://doi.org/10.1016/j.asoc.2021.107443>
- Nahian, J. A., Ghosh, T., Banna, H. A., Aseeri, M. A., Uddin, M. N., Ahmed, M. R., ... Kaiser, S. (2021). Towards an accelerometer-based elderly fall detection system using cross-disciplinary time series features. *IEEE Access*, 9, 39413–39431. doi: 10.1109/ACCESS.2021.3056441
- Reportlinker. (2020). Global evaporative cooling market – growth, trends, covid-19 impact, and forecasts (2021 – 2026). *Reportlinker*. Retrieved from <https://www.reportlinker.com/p06179097>
- Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*.
- Su, S., Sun, Y., Gao, X., Qiu, J., & Tian, Z. (2019). A correlation-change based feature selection method for IoT equipment anomaly detection. *Applied Sciences*, 9(3). doi: 10.3390/app9030437
- Turner, R., Eriksson, D., McCourt, M., Kiili, J., Laaksonen, E., Xu, Z., & Guyon, I. (2020). Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. *NeurIPS 2020 Competition and Demonstration Track*.
- Wang, J., Tang, Y., He, S., Zhao, C., Sharma, P. K., Alfarraj, O., & Tolba, A. (2020). Logevent2vec: Logevent-to-vector based anomaly detection for large-scale logs in internet of things. *Sensors*(9). doi: 10.3390/s20092451
- Wang, Z., Yan, W., & Oates, T. (2017). Time series classification from scratch with deep neural networks: A strong baseline. *2017 International Joint Conference on Neural Networks (IJCNN)*, 1578–1585. doi: 10.1109/IJCNN.2017.7966039
- Weerakody, P., Wong, K. W., Wang, G., & Wendell, E. (2021). A review of irregular time series data handling with gated recurrent neural networks. *Neurocomputing*, 441. doi: 10.1016/j.neucom.2021.02.046
- Yang, Z., Rasmussen, K. B., Kieu, A. T., & Izadi-Zamanabadi, R. (2011). Fault detection and isolation for a supermarket refrigeration system – part one: Kalman-filter-based methods. *IFAC Proceedings Volumes*, 44(1), 13233–13238. (18th IFAC World Congress) doi: <https://doi.org/10.3182/20110828-6-IT-1002.03115>
- Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., & Chen, H. (2018). Deep autoencoding gaussian mixture model for unsupervised anomaly detection. *ICLR 2018 Conference*.