Enhancing Anomaly Detection Methods for Energy Time Series Using Latent Space Data Representations

Marian Turowski
Institute for Automation and Applied Informatics
Karlsruhe Institute of Technology
Germany
marian.turowski@kit.edu

Benedikt Heidrich
Institute for Automation and Applied Informatics
Karlsruhe Institute of Technology
Germany
benedikt.heidrich@kit.edu

Kaleb Phipps
Institute for Automation and Applied Informatics
Karlsruhe Institute of Technology
Germany
kaleb.phipps@kit.edu

Kai Schmieder*
Fraunhofer Institute for Industrial Engineering IAO
Germany
kai.schmieder@iao.fraunhofer.de

Oliver Neumann
Institute for Automation and Applied Informatics
Karlsruhe Institute of Technology
Germany
oliver.neumann@kit.edu

Ralf Mikut
Institute for Automation and Applied Informatics
Karlsruhe Institute of Technology
Germany
ralf.mikut@kit.edu

Veit Hagenmeyer
Institute for Automation and Applied Informatics
Karlsruhe Institute of Technology
Germany
veit.hagenmeyer@kit.edu

ABSTRACT

The increasing number of recorded energy time series calls for an automated operation of smart grid applications such as load forecasting and load management. While these applications require anomaly-free data to perform well, the recorded data often contains anomalies. The numerous methods to detect these anomalies are typically applied directly to the recorded data. However, for other tasks such as forecasting, promising performance has been achieved when directly applying methods to a meaningful feature space of the data, i.e., the latent space data representation. We, therefore, propose a novel approach to generally enhance anomaly detection methods for energy time series by taking advantage of their latent space representation. We create latent space data representations using a conditional Invertible Neural Network (cINN) and a conditional Variational Autoencoder (cVAE) and directly apply existing supervised and unsupervised detection methods to this representation. We evaluate the latent space data representation qualitatively by visualizing the separation of anomalies and non-anomalous data. We also quantitatively evaluate our approach by applying supervised and unsupervised detection methods to real-world load data containing two groups of artificially inserted anomalies: technical faults and unusual consumption. We show that our approach generally improves the anomaly detection performance of the considered methods while only moderately increasing computational cost.

CCS CONCEPTS

• **Computing methodologies** → **Anomaly detection**; **Learning latent representations**; **Neural networks**.

KEYWORDS

anomaly detection, latent space data representation, energy time series

ACM Reference Format:
Marian Turowski, Benedikt Heidrich, Kaleb Phipps, Kai Schmieder, Oliver Neumann, Ralf Mikut, and Veit Hagenmeyer. 2022. Enhancing Anomaly Detection Methods for Energy Time Series Using Latent Space Data Representations. In *The Thirteenth ACM International Conference on Future Energy Systems (e-Energy '22), June 28–July 1, 2022, Virtual Event, USA.* ACM, New York, NY, USA, 20 pages. https://doi.org/10.1145/3538637.3538851

1 INTRODUCTION

The transition to renewable energy sources in energy supply goes hand in hand with the implementation of the so-called smart grid [20]. In the smart grid, one key element is digital metering, where smart meters and sensors record power or energy consumption and generation as time series [1]. Due to the growing number of installed smart meters, the number of recorded time series increases, calling for an automated operation of smart grid applications such as customer profiling, load analysis, load forecasting, and load management [30, 38]. Since these applications use the recorded

*Also with Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology.

e-Energy '22, June 28–July 1, 2022, Virtual Event, USA

ACM ISBN 978-1-4503-9397-3/22/06.
https://doi.org/10.1145/3538637.3538851

time series as input, their performance depends on input data that accurately reflects the typical behavior of the underlying system.

However, these recorded time series mostly contain anomalies [38]. Anomalies are patterns that deviate from "a well defined notion of normal behavior" [5, p. 15:2] and can arise for many causes such as atypical user behavior [25], smart meter failures [37], and energy theft [40]. These deviations can result in data points or patterns in the time series that represent wrong or misleading information and may be especially problematic for down-stream applications [38]. For example, anomalies such as positive or negative spikes violate the underlying distribution corresponding to normal behavior and thus can have a great impact. Data containing these anomalies may result in incorrect forecasts, leading to inappropriate energy schedules and ultimately affecting energy system stability in an automated smart grid setting. Therefore, detecting anomalies in recorded time series is an important recent issue in energy systems [12]. To detect anomalies of various types, a large variety of methods, often categorised as supervised or unsupervised methods, are employed [12]. These methods are typically applied directly or after scaling to the data containing anomalies.

However, for other tasks, machine learning methods recently demonstrated promising performance when directly applied to the so-called latent space representation of the data. This latent space is an abstract multi-dimensional space containing a meaningful representation of features that is often not directly interpretable. Such latent space data representations have been successfully applied in forecasting [14, 24], offline reinforcement learning [29], photo upsampling [22], path planning [13], and trajectory adjustment [19]. Furthermore, with regards to anomaly detection, there is evidence for a medical application that the latent space better separates the representation of anomalies and non-anomalous data [28].

In order to separate anomalous and non-anomalous data in energy time series, a latent space that follows a known and traceable latent space distribution could be particularly useful. If this distribution has clearly defined mathematical properties, as is the case with the Gaussian distribution, these properties will help define how anomalies are represented. Given these considerations, we can use the representation of data in the latent space to enhance anomaly detection.

The present paper, therefore, proposes a novel approach to generally enhance anomaly detection methods for energy time series by taking advantage of their latent space representation. For this approach, we first train a generative method to learn a mapping from the original data to the latent space. Given the learned mapping, the generative model is used to create the latent space representation of an input time series containing anomalies. The resulting latent space data representation serves then as an input for an arbitrary existing supervised or unsupervised anomaly detection method.

To evaluate the proposed approach, we firstly qualitatively examine its benefit by visualizing how latent space data representations and common data representations separate anomalies and non-anomalous data. Secondly, we quantitatively evaluate how the proposed approach improves the detection performance. For this purpose, we apply a selection of existing supervised and unsupervised detection methods to real-world load data, where we insert artificial anomalies of two groups. Anomalies of the first group represent technical faults derived from real-world data that violate the underlying distribution corresponding to normal behavior and can be easily recognized by a human. Anomalies of the second group comprise unusual consumption that remain in the underlying distribution and are hard to recognize. By generally improving the detection performance of arbitrary existing detection methods, the proposed approach helps to provide high-performance anomaly detection in automated settings.

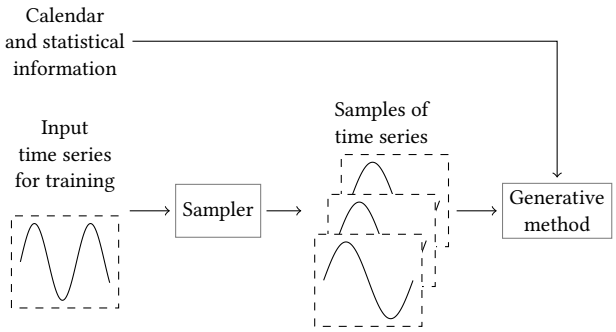


Figure 1: To train the generative method in the proposed approach, samples of an input time series as well as calendar and statistical information are used.

The remainder of the present paper is organized as follows. Section 2 introduces the proposed approach to enhance anomaly detection method by directly using the latent space data representation created with a generative model. In Section 3, we describe the experimental setting of the performed evaluation. In Section 4, we then report the evaluation results. Finally, we discuss the proposed approach in Section 5 and conclude the paper in Section 6.

2 ANOMALY DETECTION USING LATENT SPACE DATA REPRESENTATIONS

This section explains how latent space data representations can be used to enhance anomaly detection methods[1]. First, we describe how latent space data representations of time series can be created with a generative method and how this method is trained in both supervised and unsupervised anomaly detection settings. We then present how the trained generative method is applied to detect anomalies contained in a time series.

2.1 Create Latent Space Data Representations With a Generative Method

To create a latent space representation of a time series $\mathbf{z} \in \mathbb{Z}$, we need to realize a mapping $f : \mathbb{X} \to \mathbb{Z}$ from the original realization space $\mathbb{X}$ to the latent space $\mathbb{Z}$. To ensure this mapping represents a known and tractable latent space distribution P_Z in the latent space, it should be realized with either a Variational Autoencoder (VAE) [17] or an Invertible Neural Network (INN) [16]. Both methods can be extended with a conditioning mechanism to a conditional VAE (cVAE) [32] or a conditional INN (cINN) [2], allowing them to process conditional inputs. With calendar and statistical information as conditional inputs, these methods can consider typical properties of energy time series, i.e., daily, weekly, and yearly patterns. While

[1] https://github.com/KIT-IAI/EnhancingAnomalyDetectionMethods

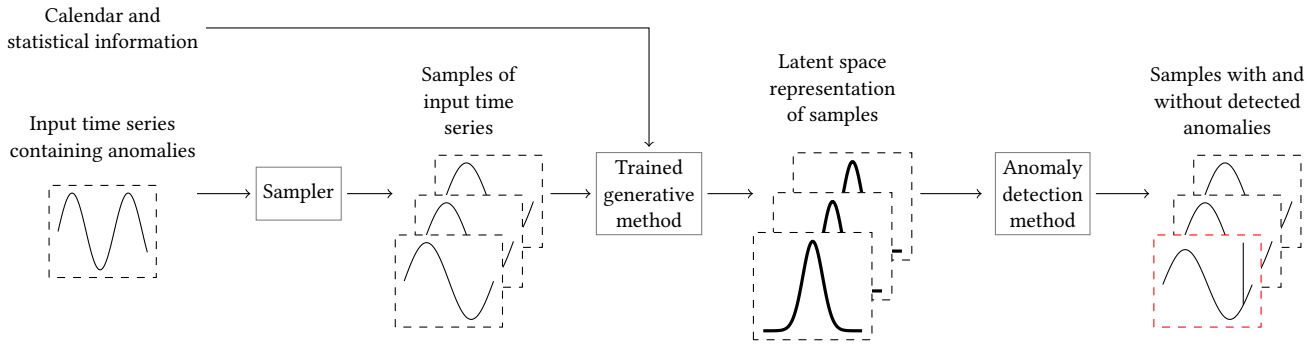


Figure 2: In the proposed approach, the previously trained generative method uses samples of an input time series containing anomalies as well as calendar and statistical information as inputs. Based on these inputs, the trained generative method provides the latent space representation of the samples. Using this latent space data representation, an arbitrary anomaly detection method detects the samples of the considered time series that contain anomalies.

both methods realize the mapping f, they differ in their structure. cVAEs consist of a jointly trained encoder and decoder with the encoder realizing the mapping f and the generator reconstructing the original representation of the time series, i.e., the mapping $g : \mathbb{Z} \rightarrow \mathbb{X}$. In contrast, cINNs only comprise a single bijective mapping $f^{-1} = g$ that realizes both the encoding and decoding.

However, since we are only interested in the latent space representation and not the reconstructed representation, we only focus on learning the mapping f. To this means, we use a cVAE or cINN to create this latent space representation using the mapping

$$f : \mathbb{X} \rightarrow \mathbb{Z}, \ \mathbf{x} \mapsto f(\mathbf{x}; \mathbf{d}, \mathbf{s}, \theta) = \mathbf{z}, \tag{1}$$

where $\mathbf{x} \in \mathbb{X}$ is the time series with arbitrary but fixed length L, $\mathbf{d}$ is calendar information of length L, $\mathbf{s}$ is statistical information of arbitrary but fixed length, and θ is the set of all trainable parameters.

2.2 Training of the Generative Method

As shown in Figure 1, the training of the selected generative method is based on samples of an input time series as well as calendar and statistical information. The training process itself differs for supervised and unsupervised anomaly detection as follows:

Supervised anomaly detection. For supervised anomaly detection, we take advantage of the labeled anomalies and train the selected generative method with fixed-length samples of an anomaly-free time series. For each fixed-length sample, we calculate the loss $\mathcal{L}_i$, which varies depending on the generative method selected. For a cINN, we use a maximum likelihood optimization based on the *change of variable formula*. This results in the maximum likelihood loss for a sample $\mathbf{x}_i$ defined as

$$\mathcal{L}_i = \frac{\| f(\mathbf{x}_i; \mathbf{d}_i, \mathbf{s}_i, \theta) \|_2^2}{2} - \log \mid J_i \mid, \tag{2}$$

where $J_i = \det(\partial f / \partial \mathbf{x}|_{\mathbf{x}_i})$ is the determinant of the Jacobian evaluated for the i-th sample [2]. For a cVAE, we use a reconstruction loss with regularization, resulting in the loss for a sample $\mathbf{x}_i$

$$\mathcal{L}_i = \mathbb{E}\left[(\hat{\mathbf{x}}_i - \mathbf{x}_i)^2\right] + \mathbb{KL}\left(\mathbf{x_i}, P_Z\right), \tag{3}$$

where $\hat{\mathbf{x}}_i$ is the reconstructed time series sample from the cVAE and $\mathbb{KL}$ is the Kullback-Leibler divergence [18]. This loss function $\mathcal{L}_i$ ensures that the generative methods learn a standard normal distribution of a non-anomalous time series as latent space distribution P_Z. Therefore, when applying to a time series containing anomalies, anomalies are likely to be mapped to the outer regions of the latent space distribution and thus are easy to detect.

Unsupervised anomaly detection. For unsupervised anomaly detection, the training process of the selected generative method has to cope with non-existent anomaly labels for the data points. This is realized on the fair assumption that the minority of the used training data is anomalous and that the training errors are higher for anomalous data points than for non-anomalous data points. We take advantage of these expected higher errors for anomalies by defining a contamination c for the training process. The contamination c represents the assumed share of anomalous data points in the considered time series and is used to calculate the threshold quantile Q_c for the training errors of each sample of a batch in the training process. Each sample with a training error above this threshold quantile Q_c is excluded from the loss function $\mathcal{L}_i$. The resulting adapted loss for a sample x_i is

$$\mathcal{L}_i' = \begin{cases} \mathcal{L}_i, & f(\mathbf{x}_i; \mathbf{d}_i, \mathbf{s}_i, \theta) < Q_c \\ 0, & \text{else} \end{cases}, \tag{4}$$

where $\mathcal{L}_i$ is the loss from Equation (2) or Equation (3), depending on the generative method used. Using this loss ensures that the selected generative model is also capable of learning an anomaly-free latent space data representation with the latent space distribution P_Z in an unsupervised manner.

2.3 Detecting Anomalies in Time Series Using the Latent Space Data Representation

Given the trained generative method, anomalies contained in a time series are detected as shown in Figure 2. Firstly, from an input time series containing anomalies, a sampler draws samples which serve as input for the trained generative method. As additional inputs, the trained generative method uses calendar and statistical information associated with this time series. Secondly, given these inputs, the trained generative method creates a latent space representation of the input time series' samples. Thirdly, an arbitrary anomaly

detection method is directly applied to the created latent space data representation to detect the samples that contain anomalies.

This approach differs for supervised and unsupervised anomaly detection methods. For a supervised anomaly detection method, two steps are involved: Firstly, it is trained on the created latent space data representation using a training set with labeled anomalies. Secondly, it classifies the samples from a test set. An unsupervised anomaly detection methods, however, is applied to a complete data set without labeled anomalies.

3 EXPERIMENTAL SETTING

In this section, we present how we evaluate the proposed approach. After describing the data set and the inserted artificial anomalies, we introduce the used generative models, the compared data representations, and the applied anomaly detection methods. Finally, we describe the evaluation criteria and the hard- and software.

3.1 Data Set and Inserted Anomalies

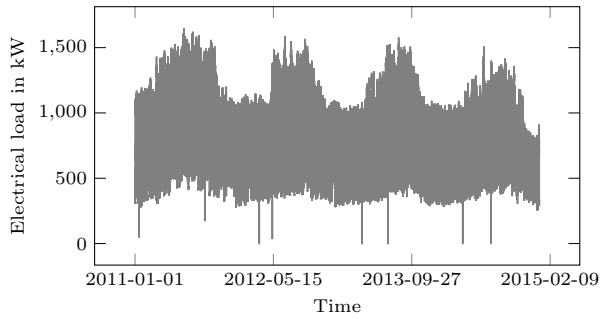
For the evaluation, we select the publicly available "ElectricityLoadDiagrams20112014 Data Set"[2] from the UCI Machine Learning Repository [8]. This data set has a quarter-hourly temporal resolution and contains electrical load time series of 370 clients, which are mostly available for the period from the beginning of 2011 until the end of 2014. To cover the complete period of four years and to consider the electrical load of a typical client, we select the time series `MT_200` for the evaluation (see Figure 3a).

Since the selected time series does not contain labeled anomalies, we insert artificial anomalies of two groups (see e.g., Figure 3b), namely technical faults in the metering infrastructure and unusual consumption. In the following, we briefly introduce the anomaly types of each group (see Figure 4 and Equations (6) to (13) in Appendix A).

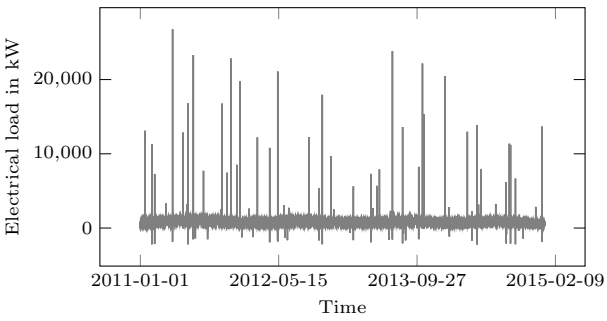
As anomalies of the first group, we select the four types of anomalies that are identified in real-world load time series in [34] and that violate the underlying distribution corresponding to normal behavior. While the values of anomaly types 1 and 3 are not in the valid range, anomaly types 2 and 4 comprise values from the valid range that are not part of typical patterns in load time series.

- Anomaly type 1 refers to a negative power spike followed by zero power values and a positive spike (see Figure 4a). This characteristic can be based on a load time series whose values are derived from an energy time series containing missing values due to a communication error.
- Anomaly type 2 comprises several zero power values followed by a positive spike (see Figure 4b). This characteristic can be a result of an interruption in the transmission of power values from smart meters.
- Anomaly type 3 is a negative power spike (see Figure 4c). It could be caused by external recalibration of a smart meter reading so that, together with the readings of other smart meters, the meter reading matches a certain amount of load.
- Anomaly type 4 is a positive power spike (see Figure 4d). It may be due to, for example, the change from daylight saving time to standard time, where power values of five time steps are recorded as the value of one time step.

[2]https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014



(a) Without inserted anomalies.



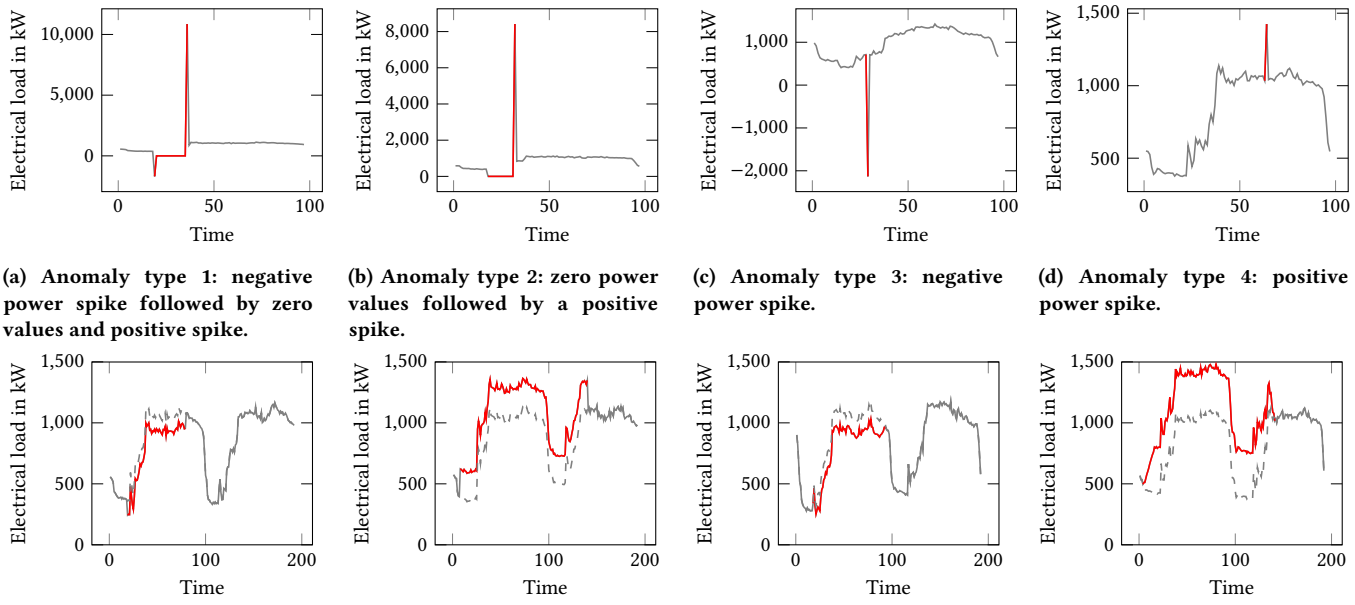
(b) With 20 inserted anomalies of each of the four types.

Figure 3: Overview of the selected data without inserted anomalies and with inserted anomalies of all four types of the first group.

As artificial anomalies of the second group, we insert four types of anomalies representing unusual behavior: Anomaly types 5 and 7 represent unusually low power consumption, while anomaly types 6 and 8 illustrate unusually high consumption. These four anomaly types comprise values from the valid range and represent in themselves typical patterns.

- Anomaly type 5 is an abrupt small temporary reduction in the power values (see Figure 4e). This characteristic can be caused by a large device temporarily shutting down, resulting directly in lower consumption.
- Anomaly type 6 is an abrupt small temporary increase in the power values (see Figure 4f). This characteristic can be the result of switching on a rarely used large device for a short period of time.
- Anomaly type 7 is also a period of temporary reduction in the power values, however with a gradual start and end (see Figure 4g). It could be caused by a large device in an unusual operating mode gradually requiring less power for a period of time, before slowly returning to its usual performance.
- Anomaly type 8 is a again small temporary increase in the power values, however with a gradual start and end (see Figure 4h). Similar to anomaly type 7, it may be due to a device in an unusual operating mode that gradually requires more power, before slowly returning to its usual performance.

For the evaluation, we insert 10, 20, 30, 40, and 50 anomalies of each anomaly type into the selected time series. This corresponds



(a) Anomaly type 1: negative power spike followed by zero values and positive spike.

(b) Anomaly type 2: zero power values followed by a positive spike.

(c) Anomaly type 3: negative power spike.

(d) Anomaly type 4: positive power spike.

(e) Anomaly type 5: abrupt small temporary reduction in the power values.

(f) Anomaly type 6: abrupt small temporary increase in the power values.

(g) Anomaly type 7: small temporary reduction in the power values with a gradual start and end.

(h) Anomaly type 8: small temporary increase in the power values with a gradual start and end.

Figure 4: Examples of the anomaly types 1 to 4 from the technical faults (top row) and anomaly types 5 to 8 from the unusual consumption (bottom row) that we insert as artificial anomalies into the selected data.

to 3, 5, 8, 10, and 12 % of the data for the technical faults and 6, 11, 16, 21, and 26 % of the data for the unusual consumption.

3.2 cINN and cVAE as used Generative Models

Since cINNs and cVAEs can be used as generative method in the proposed latent space-based approach, we perform the evaluation with a representative from both generative methods. After introducing the selected cINN and the selected cVAE, we describe the input data used for both generative models.

cINN. The selected cINN consists of 10 GLOW coupling layers [16] that implement a type of generative flow. Each of them is followed by a random permutation and contains a subnetwork that allows the coupling layer to learn. We use a fully connected network as subnetwork. To account for conditional information, we use a conditioning network as proposed by [2]. The conditioning network processes the conditional information and is also a fully connected network as proposed in [11] (see Table 3 in Appendix A). For the training of the cINN, we apply a batch size of 512, the Adam optimizer [15], and a maximum of 50 epochs.

cVAE. The selected cVAE comprises an encoder and a decoder. Both are fully connected networks (for details, see Table 4 in Appendix A). For the training of the cVAE, we use a batch size of 512, the Adam optimizer [15], and maximum of 100 epochs.

Input data. To train both generative methods, we use the first 15000 data points of the selected time series. We standardize these data points, before creating samples with a size of 96. We also use the information contained in the time stamps of the considered time series as calendar information. It comprises the hour of the day, the month of the year, and the weekday. As statistical information, we choose the mean of the considered time series sample.

3.3 Data Representations for Comparison

Using the selected time series, we compare the two latent space representations generated by the cINN and the cVAE as proposed in our approach with two benchmark data representations. The two benchmark data representations stand for the common approach of directly applying anomaly detection methods to the given data.

The first latent space data representation is from the cINN. For this, we standardize the selected time series and create overlapping samples with a size of 96. These samples serve as an input for the trained cINN that generates the resulting latent space data representation. The second latent space data representation is from the cVAE. It is created in the same way as for the cINN.

For the first benchmark data representation, we standardize the selected time series to obtain a scaled data representation, before creating overlapping samples with a size of 96. For the second benchmark data representation, we use the unaltered time series as an unscaled data representation, from which we create overlapping samples with a size of 96.

3.4 Applied Anomaly Detection Methods

For the evaluation of our approach, we select existing anomaly detection methods and apply them to the four data representations to detect anomalous and non-anomalous data. To consider different

learning assumptions, i.e., inductive biases [23], we select seven supervised and four unsupervised anomaly detection methods, which we briefly present below including their application.

Supervised methods. The selected supervised methods consider anomaly detection as a binary classification problem, where each data point is assigned the label anomaly or non-anomalous data.

As first supervised detection method, we select the Logistic Regression (LR). In the LR for binary outcomes, the posterior probabilities of the outcomes are modeled with a logistic function [9].

The second method we choose is the Gaussian Naïve Bayes (NB). The NB estimates a conditional probability by assuming the conditional independence of the input features, given the prior probability of the output variable [8, 33].

As third method, we select the Random Forest (RF). The RF uses bagging to reduce the variance of an estimated prediction by combining the predictions of multiple decision trees [3, 9, 33].

We select XGBoost as the fourth method because it is effective and widely used in various machine learning competitions. More specifically, XGBoost is a gradient boosting machine and optimizes a regularized objective function using gradient decent [6].

As fifth method, we choose the k-Nearest Neighbor (kNN) method. It uses a proximity measure to classify a test sample based on the similarity of training instances [7].

We select the Multi-Layer Perceptron (MLP) as the sixth method. As an Artificial Neural Network, it approximates an arbitrary function through multiple hidden layers of interconnected nodes and applying activation functions between the layers [e.g. 23, 39].

As seventh method, we choose a Support Vector Machine for classification (SVC) that maximizes the hyperplane between the binary classes to classify test samples [36].

Unsupervised methods. The selected unsupervised methods analyze the data to uncover the underlying normal behavior and then identify anomalous data points that violate this behavior.

As first unsupervised detection method, we select the Local Outlier Factor (LOF). The LOF measures the distances of a sample to its k-nearest neighbors to estimate the local density. By comparing the local density to the local densities of its neighborhood, the LOF is able to identify samples as non-anomalous or anomalous [4].

The second method is the Isolation Forest (iForest). It is an ensemble of isolation trees that randomly partitions samples in randomly selected features. The averaging path length of samples in different isolation trees serves as the indicator for anomalous data [21].

As third method, we choose an autoencoder (AE). It learns a mapping to the latent representation of data and a mapping back to the reconstruction of the input [31].

The fourth method is a variational autoencoder (VAE). The VAE learns the probability distribution of the data in the latent space to reconstruct its input [17].

Application. To apply the unsupervised detection methods, we use the complete selected data with inserted anomalies. To apply the supervised detection methods, however, we split the selected data with inserted anomalies to obtain a training and a test set. We use the first 5000 data points as training set. As test set, we use all data points except the first 15000 data points because these 15000 data points are used for the training of the generative models.

To both sets, we apply the selected supervised detection methods with default hyperparameters, i.e., the hyperparameters set as default in the available implementation, and with the best performing hyperparameters. To determine the best performing hyperparameters, we choose hyperparameters and select corresponding values for each method (see Table 5 in Appendix C). Over the resulting hyperparameter grid, we perform a cross-validated grid search on the training set. We choose the parameters that yield the best F1-Score (5) for a data representation and group of anomalies as best performing hyperparameters for this data representation and group of anomalies (see Tables 6 to 17 in Appendix D).

3.5 Evaluation Criteria

To quantitatively evaluate the selected anomaly detection methods on the four data representations, we use three evaluation criteria.

The first evaluation criterion is the detection performance of the methods. For this criterion, we choose the commonly used F1-Score as metric, which is the harmonic mean of precision and recall. It is calculated on the samples and defined as

$$\text{F1-Score} = \frac{\text{TP}}{\text{TP} + \frac{1}{2} \cdot (\text{FP} + \text{FN})}, \tag{5}$$

where TP are the true positives, FP the false positives, and FN the false negatives in relation to the inserted artificial anomalies. In the calculation of the F1-Score, a sample is considered as an anomaly as soon as one of its data points is an inserted artificial anomaly.

The second evaluation criterion is the robustness of the methods' detection performance. To assess the detection robustness, we calculate the F1-Score for different shares of inserted anomalies.

The third evaluation criterion is the computational cost of the detection. To assess the computational cost, we measure run-times. For the supervised anomaly detection methods, we measure the run-times for training the supervised cINN and cVAE, finding the methods' best hyperparameters, and for training them given the best performing hyperparameters. Similarly, for the unsupervised anomaly detection methods, we determine the run-times for training the unsupervised cINN and cVAE and fitting the methods.

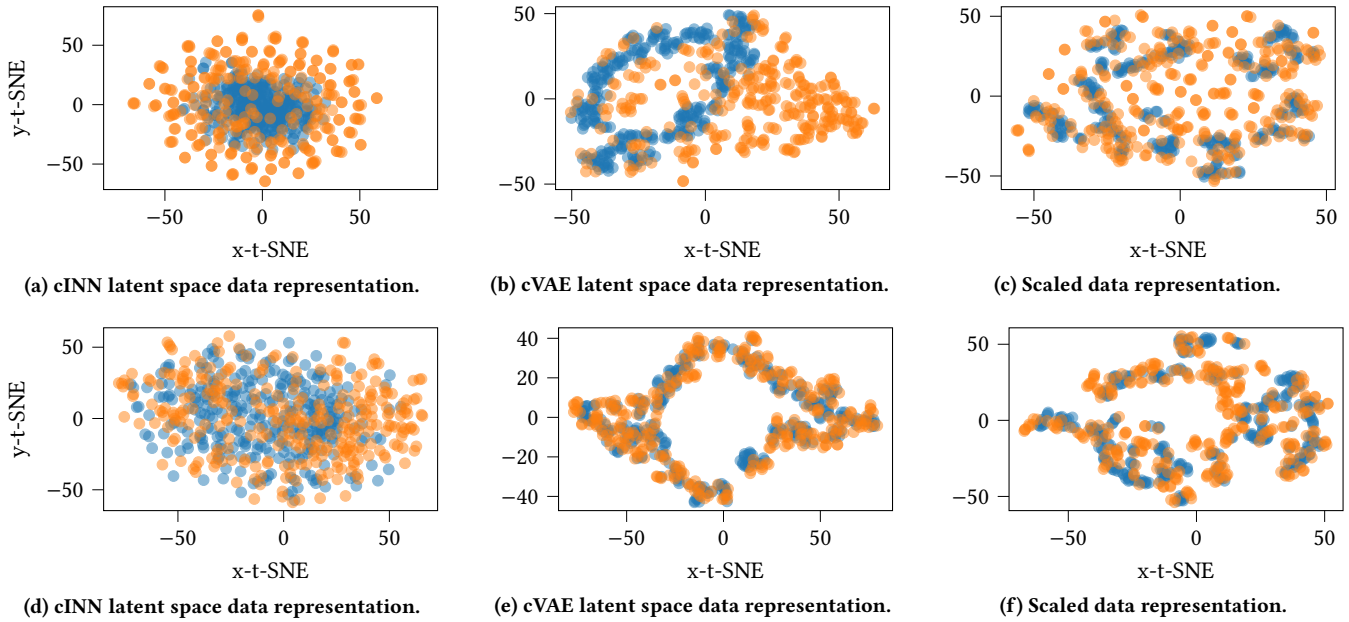
3.6 Hard- and Software

For a better comparability of the results, we use the same hardware throughout the evaluation, namely a 48 core system with 256 GB RAM, where each core has 2.1 GHz. Furthermore, all selected detection methods are implemented in Python. More specifically, for XGBoost, we use its available implementation [6]; for all others, Scikit-learn [27]. The cINN is implemented with FrEIA[3] and PyTorch [26] and the cVAE with PyTorch [26]. To automate the evaluation with these implementations, we additionally use pyWATTS[4] [10].

4 RESULTS

To qualitatively evaluate the benefit of using the latent space data representation in anomaly detection, we first visualize how the four data representations separate anomalies and non-anomalous data.

[3] https://github.com/VLL-HD/FrEIA
[4] https://github.com/KIT-IAI/pyWATTS



(a) cINN latent space data representation.
(b) cVAE latent space data representation.
(c) Scaled data representation.
(d) cINN latent space data representation.
(e) cVAE latent space data representation.
(f) Scaled data representation.

Figure 5: t-SNE visualizations of 300 random samples without anomalies (blue) and 300 random samples with anomalies (orange) in the cINN's latent space, the cVAE's latent space, and the unscaled data representations with 20 inserted anomalies of technical faults (top row) and unusual consumption (bottom row).

Second, we report the quantitative evaluation criteria when applying supervised and unsupervised anomaly detection methods to the data representations containing inserted artificial anomalies. Since the scaled and unscaled data representations perform similarly, we focus on the scaled data representation in the following.

4.1 Visualization of Anomalies in Data Representations

To analyze how the four data representations separate anomalies and non-anomalous data, we randomly choose 300 samples without anomalies and 300 samples with anomalies from the test set of the supervised detection methods. We visualize the cINN latent space, cVAE latent space, scaled, and unscaled data representations of the chosen samples with a t-distributed stochastic neighbor embedding (t-SNE) [35] using two dimensions. For an optimal anomaly detection, samples with anomalies and samples without anomalies should be clearly separated without overlapping.

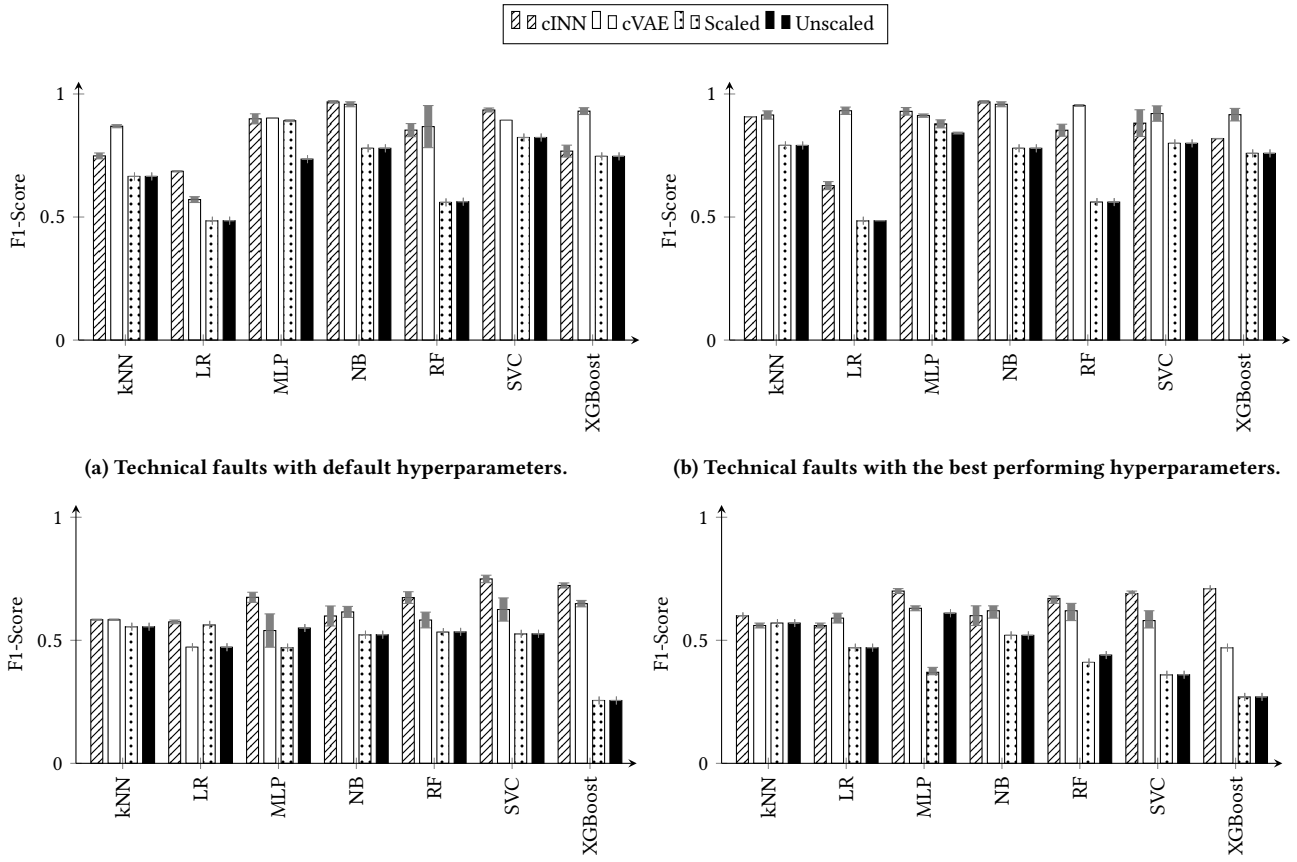
Figure 5 shows the resulting t-SNE visualization of samples with anomalies of technical faults and unusual consumption and samples without anomalies for three of the four data representations for the sake of graphical clarity (see Figure 11 in Appendix E for the unscaled). For both groups of anomalies, the t-SNE visualizes less overlap between samples with anomalies and samples without anomalies for the cINN latent space data representation than for the scaled data representation. Furthermore, for the cINN latent space data representation, the samples with anomalies are grouped around the main cluster of samples without anomalies.

4.2 Data Representations in Supervised Anomaly Detection

Detection performance. We evaluate the detection performance of the supervised anomaly detection methods with both default and best-performing hyperparameters for technical faults and unusual consumption. For both groups of anomalies, we insert 20 anomalies of each type belonging to this group. Figure 6a and Figure 6b show the resulting F1-Scores for the technical faults and Figure 6c and Figure 6d for the unusual consumption. For each unsupervised method, the bars indicate the average F1-Score for the cINN latent space, cVAE latent space, scaled, and unscaled data representations. The gray error bars show the best and the worst observed F1-Scores in multiple runs using varying random initialization for the cINN, cVAE, and the detection methods.

With default hyperparameters, all evaluated methods yield the best F1-Scores for both groups of anomalies when using a latent space data representation. Compared to the scaled data representation, the F1-Scores of the cINN latent space representation are 21 % better on average, ranging from 1 % for the MLP to 52 % for the RF. The F1-Scores of the cVAE latent space representation are 23 % better on average, ranging from 1 % for the MLP to 55 % for the RF. Note that, despite the general improvement through using a latent space data representation, the F1-Scores strongly vary between the evaluated methods. For example, considering the cINN latent space data representation and the technical faults, the LR yields a F1-Score of 0.69, while the NB achieves a F1-Score of 0.97.

With the best performing hyperparameters, all evaluated methods also perform best using the latent space data representation for



(a) Technical faults with default hyperparameters.

(b) Technical faults with the best performing hyperparameters.

(c) Unusual consumption with default hyperparameters.

(d) Unusual consumption with best performing hyperparameters.

Figure 6: The F1-Scores of the seven supervised detection methods applied to 20 anomalies of each type from the technical faults and unusual consumption. For each method, the bars indicate the average F1-Score for the cINN latent space, cVAE latent space, scaled, and unscaled data representations. The gray error bars show the best and the worst observed F1-Scores.

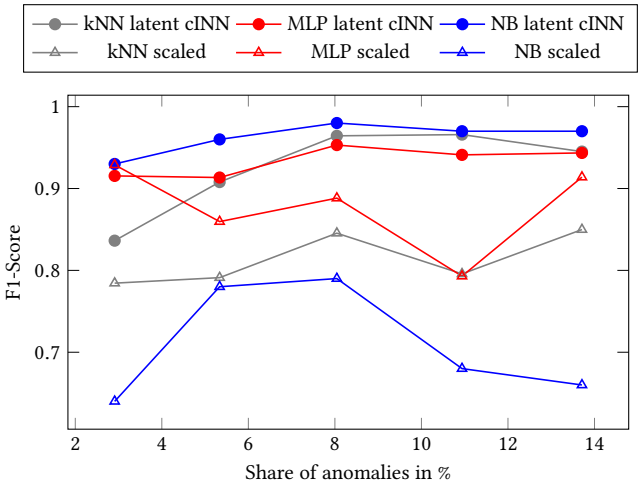
both groups of anomalies. Compared to the scaled data representation, the F1-Scores of the cINN latent space representation for the technical faults are 21 % better on average, ranging from 6 % for the MLP to 52 % for the RF. The F1-Scores of the cVAE latent space representation are 34 % better on average, ranging from 4 % for the MLP to 92 % for the LR. Note that we again observe highly varying F1-Scores across all evaluated methods.

Detection robustness. Regarding different shares of anomalies, we examine the kNN, MLP, and NB as the three best methods when using best performing hyperparameters determined for 20 anomalies of each type from technical faults. For the sake of brevity, we only consider the cINN latent space and the scaled data representations. The detection robustness for different shares of anomalies is shown in Figure 7a for the technical faults and in Figure 7b for the unusual consumption (for all other methods see Figure 13 in Appendix E). For the technical faults, the F1-Scores based on the cINN latent space data representation are consistently higher than those for the scaled data representation across all shares of anomalies. Furthermore, all anomaly detection methods perform more consistently when using the cINN latent space data representation, showing less variation than the scaled data representations. For unusual consumption, the F1-Scores when using the cINN latent space data representation are noticeably better for the MLP when compared to the scaled data representation, and similar for the kNN and NB.

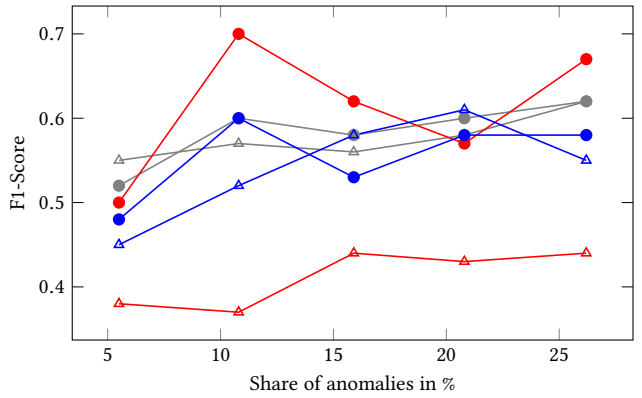
Computational cost. Concerning the computational cost reported in Table 1, we first compare the run-times required to train the supervised cINN and cVAE with the run-times to find the best performing hyperparameters and to train the supervised methods given selected hyperparameters. Afterward, we compare the run-times of the hyperparameter optimization and of the training of the supervised methods with respect to the four data representations.

The supervised training of the used cINN and cVAE takes considerably less time than the hyperparameter optimization of the MLP, SVC, and XGBoost, about the same as the hyperparameter optimization of the kNN, and more time than the hyperparameter optimization of the LR and RF. Compared to the training of the methods on all data representations, the supervised training of the cINN and cVAE, however, generally requires some more time.

The hyperparameter optimization itself requires different amounts of time depending on the data representation. On the cINN latent



(a) Technical faults



(b) Unusual consumption

Figure 7: The F1-Scores of the three best performing supervised detection methods applied to different shares of anomalies from technical faults and unusual consumption using the best performing hyperparameters. For each method, one line each indicates the resulting F1-Score for the cINN latent space and scaled data representations.

space data representation and compared to the scaled data representation, the optimization takes noticeably less time for the MLP, about the same for the kNN, LR, and RF, and more time for the SVC and XGBoost. On the cVAE latent space data representation, the optimization takes less time for the kNN, SVC, and XGBoost, about the same time for the LR and RF, and longer for the MLP. Note, however, that the run-time required for the hyperparameter optimization varies greatly across the methods.

The run-times for training the supervised methods also depend on the data representation. Using the latent space data representations for the training requires less or about the same time as using the scaled data representation for most supervised methods.

Table 1: The required run-times in seconds to train the supervised cINN and cVAE, to find the best performing hyperparameters of the supervised detection methods, and to train them given the best performing hyperparameters for the four data representations.

		cINN	cVAE	Scaled	Unscaled
Supervised training		60.77	22.24	0	0
Detection method's hyperparameter optimization	kNN	44.43	10.46	42.56	41.25
	LR	6.29	2.55	6.35	12.61
	MLP	2694.54	9116.88	9286.41	3125.75
	RF	6.04	3.71	5.20	5.00
	SVC	5906.27	4.86	899.70	11863.58
	XGBoost	1912.33	1048.95	1631.59	1631.59
Detection method's training	kNN	0.00	0.00	0.00	0.00
	LR	0.66	0.91	0.66	0.84
	MLP	3.24	7.37	7.58	1.64
	RF	0.83	0.51	0.86	0.86
	SVC	0.11	0.13	0.29	0.30
	XGBoost	1.48	1.10	1.47	1.70

4.3 Data Representations in Unsupervised Anomaly Detection

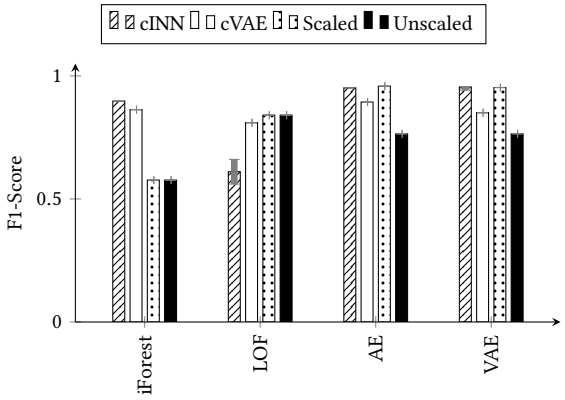
Detection performance. To evaluate the detection performance of the unsupervised anomaly detection methods, we first use latent space data representations from an unsupervised cINN and cVAE trained with a contamination of 0.05 for the technical faults and 0.1 for the unusual consumption. Afterwards, we examine the effect of different contamination values for both groups of anomalies.

For the unsupervised cINN and cVAE using data with 20 inserted anomalies of each type from an anomaly group, Figure 8 presents the F1-Scores of the unsupervised detection methods. For each method, the bars indicate the average F1-Score for the cINN latent space, cVAE latent space, scaled, and unscaled data representations. The gray error bars show the best and the worst observed F1-Scores.

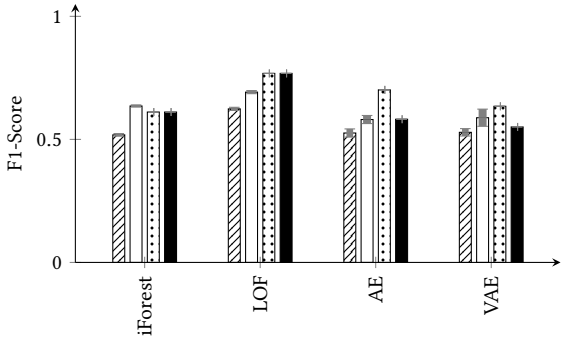
We observe that unsupervised detection methods perform differently when using the cINN and cVAE latent space data representations. Compared to the scaled data representation, the F1-Scores for the technical faults and the cINN latent space data representation show an improvement for the iForest, a similar performance for the AE and VAE, and a worse performance for the LOF. Furthermore, for the unusual consumption, the cINN and cVAE latent space data representations results in similar or lower F1-Scores than those from unsupervised anomaly detection methods using the scaled data representation.

For a contamination of 0.05, 0.1, 0.15, 0.2, and 0.25, Figure 9a and Figure 9b shows the F1-Scores on the cINN and cVAE latent space data representations of the detection methods for the technical faults and unusual consumption respectively.

For technical faults, the detection methods achieve varying F1-Scores across the different contamination values, with the best F1-Scores for all models, except for LOF, occurring with a contamination of 0.05. The performance for unusual consumption varies more, with the best F1-Scores being achieved with a contamination of 0.05, 0.1, or 0.15, depending on the considered method.



(a) Technical faults



(b) Unusual consumption

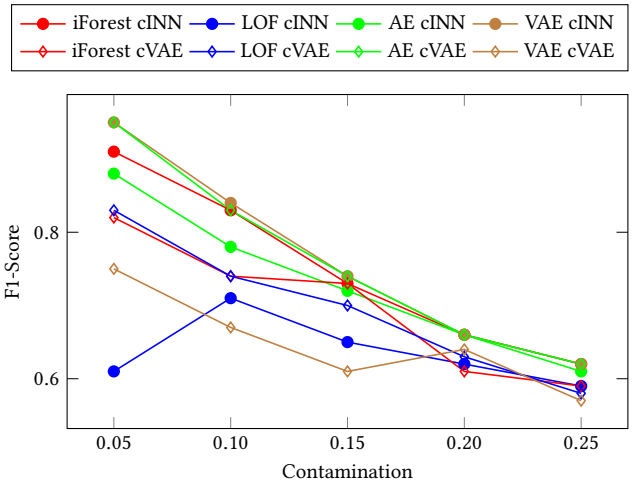
Figure 8: The F1-Scores of the four unsupervised detection methods applied to 20 anomalies of each type from technical faults and unusual consumption. For each method, the bars indicate the average F1-Score for the cINN latent space, cVAE latent space, scaled, and unscaled data representations. The gray error bars show the best and the worst F1-Scores.

Detection robustness. Regarding the analysis of different shares of anomalies from technical faults and unusual consumption, we again only consider the cINN latent space and the scaled data representations. Figure 10 shows the F1-Scores of the detection methods for these data representations. For each share of anomalies, a corresponding contamination is used for the training of the cINN.

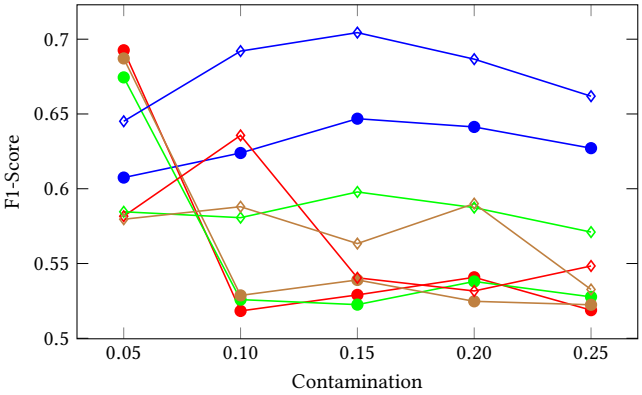
Compared to the scaled data representation, the F1-Scores of the cINN latent space data representation for technical faults are higher for the iForest, similar for the AE and VAE, and lower for the LOF. When considering unusual consumption, the latent space data representation results in lower F1-Scores across all shares of anomalies.

Computational cost. Regarding the four data representations, the run-times required to train the unsupervised cINN and cVAE and to fit the unsupervised methods are reported in Table 2.

The unsupervised training of the cINN and cVAE requires considerably more time than the fitting of most of the unsupervised methods. Additionally, fitting the AE and VAE is quicker, the iForest



(a) Technical faults

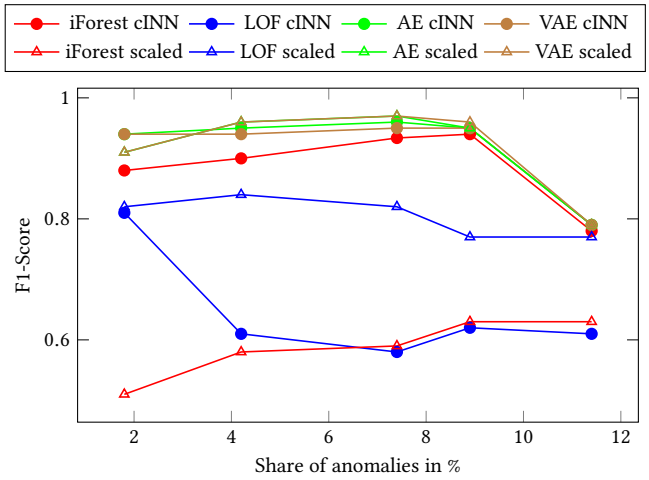


(b) Unusual consumption

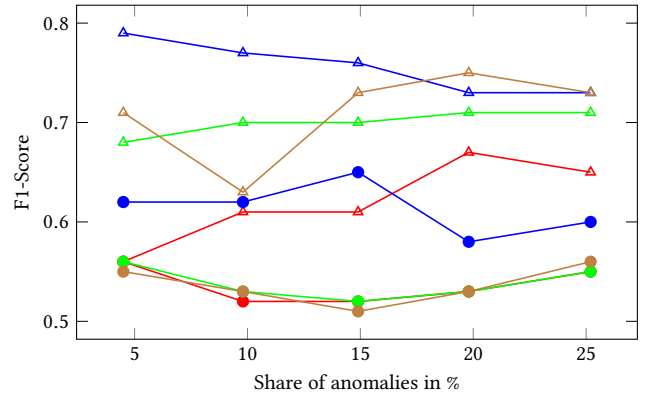
Figure 9: The F1-Scores of the four unsupervised detection methods applied to the latent space data representations created by an unsupervised cINN and cVAE with different contamination values. The data contains 20 anomalies of each type from technical faults and unusual consumption.

Table 2: The required run-times in seconds to train the unsupervised cINN and cVAE and to fit the unsupervised detection methods regarding the four data representations.

		cINN	cVAE	Scaled	Unscaled
Unsupervised training		632.96	510.22	0	0
Detection method's fitting	iForest	16.21	3.84	29.53	23.33
	LOF	370.23	269.11	207.98	206.81
	AE	443.71	99.76	579.79	320.27
	VAE	856.12	76.00	4555.05	416.99



(a) Technical faults



(b) Unusual consumption

Figure 10: The F1-Scores of the four unsupervised detection methods applied to different shares of anomalies from technical faults and unusual consumption. For each method, one line each indicates the resulting F1-Score for the cINN latent space, scaled, and unscaled data representations. For the latent space data representation, the unsupervised cINN is trained with a contamination corresponding to the share of anomalies in the data.

is similar, and the LOF is slower on the latent space compared to the scaled data representation.

5 DISCUSSION

In this section, we discuss the reported results and the benefits of the proposed approach to enhance anomaly detection methods. First, we focus on the visualization of anomalies in different data representations, before considering the three evaluation criteria for the supervised and unsupervised anomaly detection methods. Finally, we discuss the overall benefits of the proposed approach.

From the initial t-SNE visualization of anomalies in different data representations in Figure 5, we conclude that the latent space data representation helps to better separate anomalies and non-anomalous data. This separation is clearer for technical faults, but still noticeable for unusual consumption. The visualization thus confirms that using the latent space data representation as in the proposed approach is beneficial.

The detection performance and robustness of the selected supervised and unsupervised detection methods also supports this observation. For the evaluated detection methods, the results show that directly using the latent space data representation improves supervised anomaly detection methods for both technical and unusual consumption, and partly improves unsupervised anomaly detection methods when considering technical faults. For the supervised methods, this improvement occurs even without performing hyperparameter optimization and is independent of the share of inserted artificial anomalies. For unsupervised methods, the improvement is only partly noticeable for technical faults, with unusual consumption performing similarly or worse. However, since the used anomalies of unusual consumption are similarly shaped and difficult to detect even for a human, better detection performance for such anomalies might only be possible with a label. Furthermore, the LOF performs worse on the latent space data representation for both technical faults and unusual consumption. This suggests that the latent space data representation may not be suited for density based anomaly detection methods and this phenomenon should be investigated further in future work.

With regards to computational time, for some supervised detection methods, the proposed anomaly detection method reduces the time required for hyperparameter optimization and the methods' training. At the same time, for the unsupervised detection methods, the proposed anomaly detection method does not reduce the fitting time. Therefore, the proposed anomaly detection method can also be beneficial for the hyperparameter optimization and the methods' training.

Nevertheless, these improvements in the detection performance come with computational cost for the required trained cINN or cVAE. For the supervised cINN or cVAE, the required training time is considerably smaller than the time needed for hyperparameter optimization for some supervised detection methods; for the others, the time needed is in the same order of magnitude. For the unsupervised cINN or CVAE, the training takes noticeably longer than the fitting of the evaluated unsupervised detection methods, possibly due to the calculation of the quantile and the filtering of the errors for each batch. Additionally, unlike the supervised cINN and cVAE, the unsupervised cINN and cVAE are trained on all available data. Furthermore, since all available data are used, we must choose a suitable contamination to enable the cINN and cVAE to provide a beneficial data representation for unsupervised detection methods.

Considering the mentioned aspects, applying supervised detection methods without hyperparameter optimization directly on the latent space data representation as proposed is advantageous. For these methods, the training time of the cINN or cVAE is often considerably shorter than the time required to optimize their hyperparameters. At the same time, their detection performance remains high, even without hyperparameter optimization.

Overall, the proposed approach provides several benefits. The most important one is that it generally considerably enhances the detection performance of supervised and unsupervised detection

methods. This way, the latent space data representation created by a cINN or cVAE can serve as a beneficial input for any existing detection method at only moderate computational cost. This performance improvement is particularly noticeable for technical faults in both supervised and unsupervised anomaly detection methods; for unusual consumption only in supervised methods. However, the used technical faults are assumed to have more impact on downstream applications and thus should be prioritized in an automated setting. Therefore, the high performance for technical faults and improved supervised performance for unusual consumption imply that our approach is suitable to enhance anomaly detection methods in an automated setting.

6 CONCLUSION

The present paper proposes the direct use of latent space data representation to enhance anomaly detection methods. We qualitatively examine the latent space data representation created with a cINN and cVAE by visualizing the separation of anomalies and non-anomalous data. We also quantitatively evaluate anomaly detection performance using this latent space data representation by applying selected supervised and unsupervised anomaly detection methods to real-world load data containing inserted artificial anomalies of two groups, namely technical faults and unusual consumption.

Our evaluation shows that the latent space data representation enhances anomaly detection, since it results in a clearer separation between time series samples with anomalies and samples without anomalies. Furthermore, the proposed approach generally improves the detection performance of the selected supervised detection methods for both technical faults and unusual consumption with only moderate additional computational cost. We also show that this benefit is mostly observable regardless of the share of anomalies in the considered time series. For unsupervised anomaly detection methods, our approach partially improves anomaly detection methods for technical faults, but has difficulties with unusual consumption.

In future work, we will evaluate our approach with multivariate time series. We also plan to integrate the creation of latent space data representation with the training of the detection methods, and systematically evaluate hyperparameter optimization in the latent space. Moreover, we want to improve unsupervised methods for detecting unusual consumption in the latent space data representation.

ACKNOWLEDGMENTS

This project is funded by the Helmholtz Association's Initiative and Networking Fund through Helmholtz AI, the Helmholtz Association under the Program "Energy System Design", the Helmholtz Metadata Collaboration, and the German Research Foundation (DFG) as part of the Research Training Group 2153 "Energy Status Data: Informatics Methods for its Collection, Analysis and Exploitation".

REFERENCES

[1] Damminda Alahakoon and Xinghuo Yu. 2016. Smart Electricity Meter Data Intelligence for Future Energy Systems: A Survey. *IEEE Transactions on Industrial Informatics* 12, 1 (2016), 425–436. https://doi.org/10.1109/TII.2015.2414355

[2] Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. 2019. Guided Image Generation with Conditional Invertible Neural Networks. (2019). arXiv:1907.02392

[3] Leo Breiman. 2001. Random Forests. *Machine Learning* 45, 1 (2001), 5–32. https://doi.org/10.1023/A:1010933404324

[4] Markus M Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: Identifying Density-Based Local Outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. ACM, 93–104. https://doi.org/10.1145/342009.335388

[5] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly Detection: A Survey. *Comput. Surveys* 41, 3 (2009), 15:1–15:58. https://doi.org/10.1145/1541880.1541882

[6] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 785–794. https://doi.org/10.1145/2939672.2939785

[7] T. Cover and P. Hart. 1967. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* 13, 1 (1967), 21–27. https://doi.org/10.1109/TIT.1967.1053964

[8] Dheeru Dua and Casey Graff. 2019. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[9] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning* (2 ed.). Springer, New York. https://doi.org/10.1007/978-0-387-84858-7

[10] Benedikt Heidrich, Andreas Bartschat, Marian Turowski, Oliver Neumann, Kaleb Phipps, Stefan Meisenbacher, Kai Schmieder, Nicole Ludwig, Ralf Mikut, and Veit Hagenmeyer. 2021. pyWATTS: Python Workflow Automation Tool for Time Series. *arXiv:2106.10157* (2021).

[11] Benedikt Heidrich, Marian Turowski, Kaleb Phipps, Kai Schmieder, Wolfgang Süß, Ralf Mikut, and Veit Hagenmeyer. 2022. Controlling Non-Stationarity and Periodicities in Time Series Generation Using Conditional Invertible Neural Networks. *Applied Intelligence* (2022). https://doi.org/10.1007/s10489-022-03742-7

[12] Yassine Himeur, Khalida Ghanem, Abdullah Alsalemi, Faycal Bensaali, and Abbes Amira. 2021. Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives. *Applied Energy* 287 (2021), 116601. https://doi.org/10.1016/j.apenergy.2021.116601

[13] Chia-Man Hung, Shaohong Zhong, Walter Goodwin, Oiwi Parker Jones, Martin Engelcke, Ioannis Havoutis, and Ingmar Posner. 2022. Reaching Through Latent Space: From Joint Statistics to Path Planning in Manipulation. *IEEE Robotics and Automation Letters* 7, 2 (2022), 5334–5341. https://doi.org/10.1109/LRA.2022.3152697

[14] Jin-Young Kim and Sung-Bae Cho. 2021. Explainable prediction of electric energy demand using a deep autoencoder with interpretable latent space. *Expert Systems with Applications* 186 (2021), 115842. https://doi.org/10.1016/j.eswa.2021.115842

[15] Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations (ICLR 2015)*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980

[16] Diederik P. Kingma and Prafulla Dhariwal. 2018. Glow: Generative Flow with Invertible 1x1 Convolutions. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. 10215–10224. https://proceedings.neurips.cc/paper/2018/file/d139db6a236200b21cc7f752979132d0-Paper.pdf

[17] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. (2014). arXiv:1312.6114v10

[18] S. Kullback and R. A. Leibler. 1951. On Information and Sufficiency. *Annals of Mathematical Statistics* 22, 1 (1951), 79–86. https://doi.org/10.1214/aoms/1177729694

[19] K. Kutsuzawa, S. Sakaino, and T. Tsuji. 2019. Trajectory adjustment for nonprehensile manipulation using latent space of trained sequence-to-sequence model. *Advanced Robotics* 33, 21 (2019), 1144–1154. https://doi.org/10.1080/01691864.2019.1673204

[20] Fangxing Li, Wei Qiao, Hongbin Sun, Hui Wan, Jianhui Wang, Yan Xia, Zhao Xu, and Pei Zhang. 2010. Smart Transmission Grid: Vision and Framework. *IEEE Transactions on Smart Grid* 1, 2 (2010), 168–177. https://doi.org/10.1109/TSG.2010.2053726

[21] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation Forest. In *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 413–422. https://doi.org/10.1109/ICDM.2008.17

[22] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. 2020. PULSE: Self-Supervised Photo Upsampling via Latent Space Exploration of Generative Models. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2437–2445. https://doi.org/10.1109/CVPR42600.2020.00251

[23] Tom Mitchell. 1997. *Machine Learning*. McGraw-Hill, New York.

[24] Nam Nguyen and Brian Quanz. 2021. Temporal Latent Auto-Encoder: A Method for Probabilistic Multivariate Time Series Forecasting. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*. 9117–9125. https://ojs.aaai.org/index.php/AAAI/article/view/17101/16908

[25] Christian Nordahl, Marie Persson, and Håkan Grahn. 2017. Detection of residents' abnormal behaviour by analysing energy consumption of individual households.

In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 729–738. https://doi.org/10.1109/ICDMW.2017.101

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. D'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. 8024–8035.

[27] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, 85 (2011), 2825–2830. http://jmlr.org/papers/v12/pedregosa11a.html

[28] João Pereira and Margarida Silveira. 2019. Learning Representations from Healthcare Time Series Data for Unsupervised Anomaly Detection. In *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE. https://doi.org/10.1109/BIGCOMP.2019.8679157

[29] Rafael Rafailov, Tianhe Yu, Aravind Rajeswaran, and Chelsea Finn. 2021. Offline Reinforcement Learning from Images with Latent Space Models. In *Proceedings of the 3rd Conference on Learning for Dynamics and Control*. PMLR, 1154–1168. https://doi.org/v144/rafailov21a.html

[30] Bruno Rossi and Stanislav Chren. 2020. Smart Grids Data Analysis: A Systematic Mapping Study. *IEEE Transactions on Industrial Informatics* 16, 6 (2020), 3619–3639. https://doi.org/10.1109/TII.2019.2954098

[31] David E. Rumelhart, Geoffrey E. Hintont, and Ronald J. Williams. 1986. Learning Representations by Back-Propagating Errors. *Nature* 323 (1986), 533–536. https://doi.org/10.1038/323533a0

[32] Kihyuk Sohn, Xinchen Yan, and Honglak Lee. 2015. Learning Structured Output Representation using Deep Conditional Generative Models. In *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. 3483–3491.

[33] Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. 2019. *Introduction to Data Mining* (2nd global ed.). Pearson, New York.

[34] Marian Turowski, Moritz Weber, Oliver Neumann, Benedikt Heidrich, Kaleb Phipps, Hüseyin K. Çakmak, Ralf Mikut, and Veit Hagenmeyer. 2022. Modeling and Generating Synthetic Anomalies for Energy and Power Time Series. In *The Thirteenth ACM International Conference on Future Energy Systems (e-Energy '22)*. ACM.

[35] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html

[36] Vladimir N. Vapnik. 2000. *The Nature of Statistical Learning Theory* (2 ed.). Springer, New York. https://doi.org/10.1007/978-1-4757-3264-1

[37] Long Wang, Marian Turowski, Meng Zhang, Till Riedel, Michael Beigl, Ralf Mikut, and Veit Hagenmeyer. 2020. Point and contextual anomaly detection in building load profiles of a university campus. In *2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*. IEEE, 11–15. https://doi.org/10.1109/ISGT-Europe47291.2020.9248792

[38] Yi Wang, Qixin Chen, Tao Hong, and Chongqing Kang. 2019. Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges. *IEEE Transactions on Smart Grid* 10, 3 (2019), 3125–3148. https://doi.org/10.1109/TSG.2018.2818167

[39] Paul Werbos. 1975. *Beyond regression: New tools for prediction and analysis in the behavioral sciences*. Ph.D. thesis. Harvard University.

[40] Sook-Chin Yip, Wooi-Nee Tan, ChiaKwang Tan, Ming-Tao Gan, and KokSheik Wong. 2018. An anomaly detection framework for identifying energy theft and defective meters in smart grids. *International Journal of Electrical Power and Energy Systems* 101 (2018), 189–203. https://doi.org/10.1016/j.ijepes.2018.03.025

A INSERTED ARTIFICIAL ANOMALIES

For the evaluation, we create an anomaly $\hat{p}_{j,i}$ of type j with start index i. For each anomaly type mentioned above, we list the relevant parameters and specify the corresponding manipulation in a given power time series $P = p_1, p_2, ...p_N$ with length N in the following. Note that the anomaly types 1 to 4 are based on [34].

A.1 Anomaly Type 1

Parameters:

- Length $l \sim \mathcal{U}_{[5,24]}$
- Random scaling factor $r_s = 2 + r \cdot 3$ with $r \sim \mathcal{U}_{[0,1]}$

$$\hat{p}_{1,i+n} = \begin{cases} -1 \cdot mean(P) + r_s \cdot std(P), & n = 0 \\ 0, & 0 < n < l - 1 \\ \sum_{t=1}^{i+l-1} p_t, & n = l - 1 \end{cases} \quad (6)$$

A.2 Anomaly Type 2

Parameter:

- Length $l \sim \mathcal{U}_{[5,24]}$

$$\hat{p}_{2,i+n} = \begin{cases} 0, & 0 \leq n < l - 1 \\ \sum_{t=i}^{i+l-1} p_t, & n = l - 1 \end{cases} \quad (7)$$

A.3 Anomaly Type 3

Parameter:

- Random scaling factor $r_s = 0.01 + r \cdot 3.99$ with $r \sim \mathcal{U}_{[0,1]}$

$$\hat{p}_{3,i} = -r_s \cdot mean(P) \quad (8)$$

A.4 Anomaly Type 4

Parameter:

- Random scaling factor $r_s = 3 + r \cdot 5$ with $r \sim \mathcal{U}_{[0,1]}$

$$\hat{p}_{4,i} = r \cdot mean(P) \quad (9)$$

A.5 Anomaly Type 5

Parameters:

- Length $l \sim \mathcal{U}_{[48,144]}$
- Random scaling factor $r \sim \mathcal{U}_{[0.3,0.8]}$

$$\hat{p}_{5,i+n} = p_i - r \cdot p_{\min}, \quad 0 < n < l - 1, \quad (10)$$

where $p_{\min} = \min\{p_i, p_{i+1}, \ldots, p_{i+l-1}\}$.

A.6 Anomaly Type 6

Parameters:

- Length $l \sim \mathcal{U}_{[48,144]}$
- Random scaling factor $r \sim \mathcal{U}_{[0.5,1]}$

$$\hat{p}_{6,i+n} = p_i + r \cdot p_{\min}, \quad 0 < n < l - 1, \quad (11)$$

where $p_{\min} = \min\{p_i, p_{i+1}, \ldots, p_{i+l-1}\}$.

A.7 Anomaly Type 7

Parameters:

- Length $l \sim \mathcal{U}_{[48,144]}$
- Random scaling factor $r \sim \mathcal{U}_{[0.3,0.8]}$

$$\hat{p}_{7,i} = \begin{cases} p_i - r \cdot p_{\min} \cdot \frac{l}{10} \cdot i, & 0 < n < \frac{l}{10} \\ p_i - r \cdot p_{\min}, & \frac{l}{10} \leq n \leq 1 - \frac{l}{10} \\ p_i - r \cdot p_{\min} \cdot \frac{l}{10} \cdot (1 - i), & 1 - \frac{l}{10} < n < l - 1 \end{cases} \quad (12)$$

where $p_{\min} = \min\{p_i, p_{i+1}, \ldots, p_{i+l-1}\}$.

A.8 Anomaly Type 8

Parameters:

- Length $l \sim \mathcal{U}_{[48,144]}$
- Random scaling factor $r \sim \mathcal{U}_{[0.5,1]}$

$$\hat{p}_{8,i} = \begin{cases} p_i + r \cdot p_{\min} \cdot \frac{l}{10} \cdot i, & 0 < n < \frac{l}{10} \\ p_i + r \cdot p_{\min}, & \frac{l}{10} \leq n \leq 1 - \frac{l}{10} \\ p_i + r \cdot p_{\min} \cdot \frac{l}{10} \cdot (1 - i), & 1 - \frac{l}{10} < n < l - 1 \end{cases} \quad (13)$$

where $p_{\min} = \min\{p_i, p_{i+1}, \ldots, p_{i+l-1}\}$.

B IMPLEMENTATION OF CINN AND CVAE AS USED GENERATIVE MODELS

Table 3: Implementation details of the subnetwork and the conditioning network q in the used cINN.

(a) Subnetwork

Layer	Description
Input	[Output of previous coupling layer, conditional information]
1	Dense 32 neurons; activation: tanh
1	Dense horizon units; activation: linear

(b) Conditioning network

Layer	Description
Input	[Calendar information, statistical information]
1	Dense 8 neurons; activation: tanh
2	Dense 4 neurons; activation: linear

Table 4: Implementation details of encoder and decoder of the used cVAE.

(a) Encoder

Layer	Description
Input	[Normal data, conditional information]
1	Dense 64 neurons; activation: tanh
2	Dense 32 neurons; activation: tanh
3	μ: dense latent dimension; activation: linear
4	σ: dense latent dimension; activation: linear

(b) Decoder

Layer	Description
Input	[Latent data, conditional information]
1	Dense 32 neurons; activation: tanh
2	Dense 64 neurons; activation: tanh
3	Dense horizon units; activation: linear

C DEFAULT HYPERPARAMETERS

Table 5: Overview of the hyperparameters, their default values, and the evaluated values of all seven selected supervised anomaly detection methods.

Detection method	Hyperparameter	Default value	Evaluated values
kNN	n_neighbors	5	1, 3, 5, 7, 10
	p	2	1, 2, 3
	weights	uniform	uniform, distance
LR	C	1	0.01, 0.1, 1, 10, 100
	penalty	l2	l1, l2, elasticnet, none
	solver	lbfgs	newton-cg, lbfgs, liblinear, sag, saga
MLP	activation	relu	logistic, tanh, relu
	alpha	0.0001	0.00001, 0.0001, 0.001
	batch_size	auto	10, 11, 12, 13, 14, 15, 16, 32, 64, 128, 200
	hidden_layer_size	(100,)	(25,), (50,), (75,), (100,), (125,), (150,), (25, 25), (50, 50), (75, 75), (100, 100), (125, 125), (150, 150), (25, 25, 25), (50, 50, 50), (75, 75, 75), (100, 100, 100), (125, 125, 125), (150, 150, 150)
NB	*no hyperparameters*		
RF	criterion	gini	gini, entropy
	max_features	auto	sqrt, log2
SVC	C	1	0.01, 0.1, 1, 10, 100
	gamma	scale	scale, auto
	kernel	rbf	linear, sigmoid, rbf
XGBoost	booster	gbtree	gbtree, gblinear, dart
	importance_type	gain	gain, weight, cover, total_gain, total_cover
	reg_lambda	1	0, 0.1, 0.5, 1, 2, 4

D BEST PERFORMING HYPERPARAMETERS

Table 6: The best performing hyperparameters of the kNN for all data representations for technical faults.

Data	n_neighbors	p	weights
Latent cINN	1	2	uniform
	1	2	distance
	1	3	uniform
	1	3	distance
Latent cVAE	1	2	uniform
	1	2	distance
Scaled	1	2	uniform
	1	2	distance
	1	3	uniform
	1	3	distance
	3	3	distance
Unscaled	1	2	uniform
	1	2	distance
	1	3	uniform
	1	3	distance
	3	3	distance

Table 7: The best performing hyperparameters of the kNN for all data representations for unusual consumption.

Data	n_neighbors	p	weights
Latent cINN	1	1	uniform
	1	1	distance
Latent cVAE	10	2	distance
Scaled	5	3	uniform
	5	3	distance
Unscaled	5	3	uniform
	5	3	distance

Table 8: The best performing hyperparameters of the LR for all data representations for technical faults.

Data	C	penalty	solver
Latent cINN	10	none	sag
Latent cVAE	0.01	none	newton-cg
	0.1	none	newton-cg
	1	none	newton-cg
	10	none	newton-cg
	100	none	newton-cg
Scaled	0.1	l1	saga
Unscaled	0.01	none	saga
	1	l2	sag
	10	none	sag
	100	l2	saga

Table 9: The best performing hyperparameters of the LR for all data representations for unusual consumption.

Data	C	penalty	solver
Latent cINN	0.01	l1	liblineear
Latent cVAE	100	l1	liblinear
Scaled	0.01	l1	liblinear
	0.01	l1	saga
Unscaled	0.01	l1	liblinear
	0.01	l1	saga
	0.01	l2	lbgfs
	0.01	l2	liblinear
	0.01	l2	sag
	0.01	l2	saga
	0.01	none	lbgfs
	0.01	none	sag
	0.01	none	saga
	0.1	l1	saga
	0.1	l2	lbgfs
	0.1	l2	liblinear
	0.1	l2	sag
	0.1	l2	saga
	0.1	none	lbgfs
	0.1	none	sag
	0.1	none	saga
	1	l1	saga
	1	l2	lbgfs
	1	l2	liblinear
	1	l2	sag
	1	l2	saga
	1	none	lbgfs
	1	none	sag
	1	none	saga
	10	l1	saga
	10	l2	lbgfs
	10	l2	liblinear
	10	l2	sag
	10	l2	saga
	10	none	lbgfs
	10	none	sag
	10	none	saga
	100	l1	saga
	100	l2	lbgfs
	100	l2	liblinear
	100	l2	sag
	100	l2	saga
	100	none	lbgfs
	100	none	sag
	100	none	saga

Table 10: The best performing hyperparameters of the MLP for all data representations for technical faults.

Data	activation	alpha	batch_size	hidden_layer_sizes
Latent cINN	relu	0.001	15	(100,)
Latent cVAE	relu	0.0001	14	(75,75)
Scaled	relu	0.0001	14	(125,125)
	relu	0.001	12	(125,125,125)
	relu	0.001	15	(125)
Unscaled	relu	0.001	32	(125)

Table 11: The best performing hyperparameters of the MLP for all data representations for unusual consumption.

Data	activation	alpha	batch_size	hidden_layer_sizes
Latent cINN	relu	0.0001	11	(150,)
Latent cVAE	relu	0.00001	11	(75. 75)
Scaled	relu	0.0001	12	(50,50)
Unscaled	relu	0.00001	64	(100,100,100)

Table 12: The best performing hyperparameters of the RF for all data representations for technical faults.

Data	criterion	max_features
Latent cINN	gini	sqrt
Latent cVAE	entropy	sqrt
Scaled	gini	sqrt
Unscaled	gini	sqrt

Table 13: The best performing hyperparameters of the RF for all data representations for unusual consumption.

Data	criterion	max_features
Latent cINN	gini	sqrt
Latent cVAE	gini	sqrt
Scaled	entropy	log2
Unscaled	entropy	log2

Table 14: The best performing hyperparameters of the SVC for all data representations for technical faults.

Data	C	gamma	kernel
Latent cINN	100	scale	rbf
Latent cVAE	100	scale	rbf
Scaled	0.1	scale	rbf
Unscaled	0.1	scale	rbf

Table 15: The best performing hyperparameters of the SVC for all data representations for unusual consumption.

Data	C	gamma	kernel
Latent cINN	10	auto	rbf
	100	auto	rbf
Latent cVAE	1	scale	rbf
Scaled	10	scale	rbf
Unscaled	10	scale	rbf

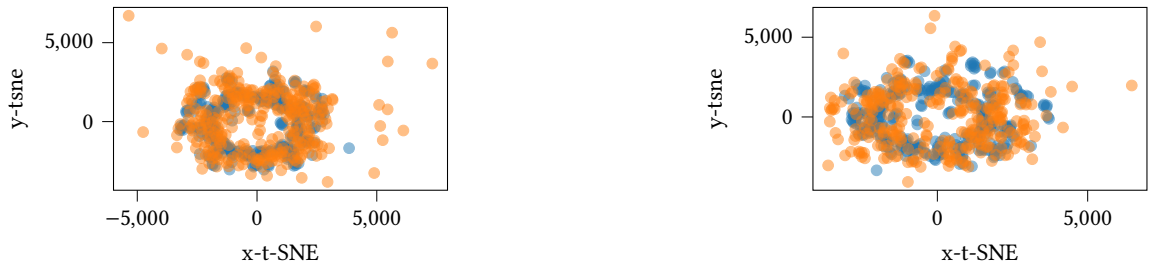
Table 16: The best performing hyperparameters of the XGBoost for all data representations for technical faults.

Data	booster	importance_type	reg_lambda
Latent cINN	gbtree	gain	0.1
	gbtree	weight	0.1
	gbtree	cover	0.1
	gbtree	total_gain	0.1
	gbtree	total_cover	0.1
	dart	gain	0.1
	dart	weight	0.1
	dart	cover	0.1
	dart	total_gain	0.1
	dart	total_cover	0.1
Latent cVAE	gbtree	gain	1
	gbtree	weight	1
	gbtree	cover	1
	gbtree	total_gain	1
	gbtree	total_cover	1
	dart	gain	1
	dart	weight	1
	dart	cover	1
	dart	total_gain	1
	dart	total_cover	1
Scaled	gbtree	gain	0
	gbtree	weight	0
	gbtree	cover	0
	gbtree	total_gain	0
	gbtree	total_cover	0
	dart	gain	0
	dart	weight	0
	dart	cover	0
	dart	total_gain	0
	dart	total_cover	0
Unscaled	gbtree	gain	0
	gbtree	weight	0
	gbtree	cover	0
	gbtree	total_gain	0
	gbtree	total_cover	0
	dart	gain	0
	dart	weight	0
	dart	cover	0
	dart	total_gain	0
	dart	total_cover	0

Table 17: The best performing hyperparameters of the XGBoost for all data representations for unusual consumption.

Data	booster	importance_type	reg_lambda
Latent cINN	gbtree	gain	1
	gbtree	weight	1
	gbtree	cover	1
	gbtree	total_gain	1
	gbtree	total_cover	1
	dart	gain	1
	dart	weight	1
	dart	cover	1
	dart	total_gain	1
	dart	total_cover	1
Latent cVAE	gblinear	gain	0
	gblinear	weight	0
	gblinear	cover	0
	gblinear	total_gain	0
	gblinear	total_cover	0
Scaled	gbtree	gain	0
	gbtree	weight	0
	gbtree	cover	0
	gbtree	total_gain	0
	gbtree	total_cover	0
	dart	gain	0
	dart	weight	0
	dart	cover	0
	dart	total_gain	0
	dart	total_cover	0
Unscaled	gbtree	gain	0
	gbtree	weight	0
	gbtree	cover	0
	gbtree	total_gain	0
	gbtree	total_cover	0
	dart	gain	0
	dart	weight	0
	dart	cover	0
	dart	total_gain	0
	dart	total_cover	0

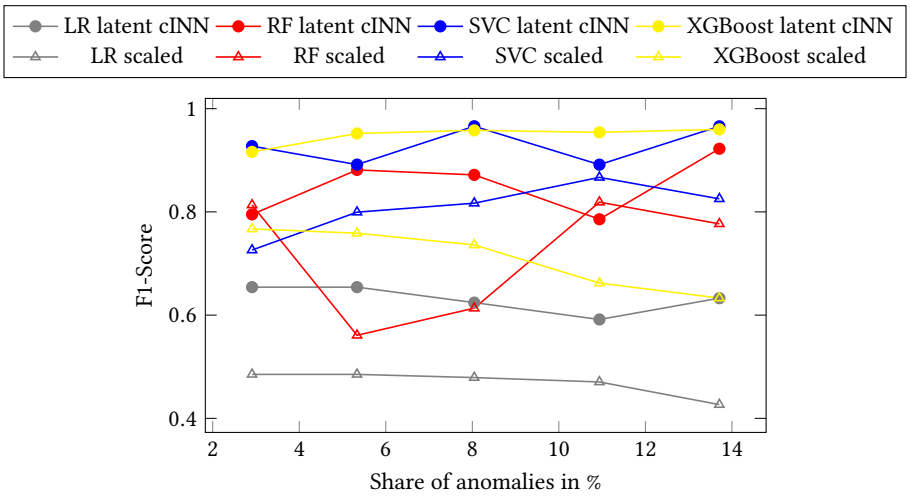
E ADDITIONAL RESULTS



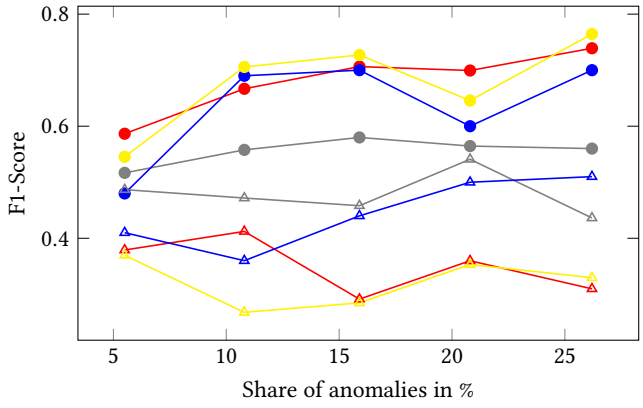
(a) Unscaled data representation for technical faults.

(b) Unscaled data representation for unusual consumption.

Figure 11: t-SNE visualizations of 300 random samples without anomalies (blue) and 300 random samples with anomalies (orange) in the unscaled latent space data representation with 20 inserted anomalies of technical faults and unusual consumption.



(a) Technical faults



(b) Unusual consumption

Figure 13: The F1-Scores of the four remaining supervised detection methods applied to different shares of anomalies from technical faults and unusual consumption using the best performing hyperparameters. For each method, one line each indicates the resulting F1-Score for the cINN latent space and scaled data representations.