

Fault Detection in Solar Thermal Systems using Probabilistic Reconstructions

Florian Ebmeier^{a,*}, Nicole Ludwig^a, Jannik Thuemmel^a, Georg Martius^{a,b} and Volker H. Franz^a

^aUniversity of Tübingen, Geschwister-Scholl Platz, Tübingen, 72072, Baden-Württemberg, Germany

^bMax Planck Institute for Intelligent Systems, Max-Planck-Ring 4, Tübingen, 72076, Baden-Württemberg, Germany

ARTICLE INFO

Keywords:

Solar thermal systems
Fault detection
Anomaly detection
Time series
Deep learning
Uncertainty estimation

ABSTRACT

Solar thermal systems (STS) present a promising avenue for low-carbon heat generation, with a well-running system providing heat at minimal cost and carbon emissions. However, STS can exhibit faults due to improper installation, maintenance, or operation, often resulting in a substantial reduction in efficiency or even damage to the system. As monitoring at the individual level is economically prohibitive for small-scale systems, automated monitoring and fault detection should be used to address such issues. Recent advances in data-driven anomaly detection, particularly in time series analysis, offer a cost-effective solution by leveraging existing sensors to identify abnormal system states.

Here, we propose a probabilistic reconstruction-based framework for anomaly detection. We evaluate our method on the publicly available PaSTS dataset of operational domestic STS, which features real-world complexities and diverse fault types. Our experiments show that reconstruction-based methods can detect faults in domestic STS both qualitatively and quantitatively, while generalizing to previously unseen systems. We also demonstrate that our model outperforms both simple and more complex deep learning baselines. Additionally, we show that heteroscedastic uncertainty estimation is essential to fault detection performance. Finally, we discuss the engineering overhead required to unlock these improvements and make a case for simple deep learning models.


1. Motivation

Domestic solar thermal systems (STS) present a promising approach to decarbonising the energy system. They require minimal natural resources, do not add strain to the electrical grid, and can be more space-efficient than photovoltaic systems. However, adoption has been hampered by low yields due to poor monitoring and undetected faults [32, 33]. Monitoring these systems can be prohibitively expensive if it involves additional sensors, complex system modelling, and continuous oversight by human experts. With increased digitalisation, substantial sensor data are now available, creating an opportunity to develop data-driven fault and anomaly detection methods that reduce costs and human supervision.

Data-driven time series anomaly detection (TSAD) has attracted considerable attention in machine learning. Among the prevailing paradigms, reconstruction- and forecasting-based methods are particularly popular, and recent years have seen a surge in the application of cutting-edge deep learning (DL) architectures to this task [1, 2, 7, 42, 35, 45]. To our knowledge, no system applies state-of-the-art deep learning anomaly detection frameworks to real-world domestic STS. This is particularly severe when considering the variety of system realisations in operational domestic STS. Generalization remains the key challenge: to build robust models that can adapt to the variety of real-world system designs, operational states, and environmental factors common in domestic solar thermal setups. Furthermore, concerns have been raised regarding the soundness of experimental protocols and benchmarking practices in many DL studies [47, 24, 38]. For example, Sarfraz et al. [38] demonstrate that simple linear methods such as principal component analysis (PCA) can outperform recent, more complex DL models; thus challenging the presumed superiority of advanced architectures.

The finding that PCA substantially outperforms DL models in TSAD is surprising, as there is no theoretical basis for this shortcoming of DL models. Previous findings for Variational Autoencoders (VAEs) show that they learn local PCA

*Corresponding author

 florian.ebmeier@uni-tuebingen.de (F. Ebmeier)

ORCID(s): 0000-0002-1518-9593 (F. Ebmeier); 0000-0003-3230-8918 (N. Ludwig); 0009-0004-4209-6239 (J. Thuemmel); 0000-0002-8963-7627 (G. Martius); 0000-0002-8457-0611 (V.H. Franz)

representations [36] and can be interpreted as non-linear PCA. Thus, we hypothesize that this performance gap is not due to the inherent shortcomings of DL models, but rather to the added complexity of optimizing overly complicated DL models. Hence, we focus on simpler methods that require little optimisation while keeping the strengths of DL models. More specifically, we show that probabilistic reconstruction-based methods make fault detection in STS feasible, and explicit heteroscedastic uncertainty estimation is a simple yet effective method to enhance fault detection accuracy.

Our experiments on the PaSTS dataset [10] will explore these hypotheses, offering insights into how data-driven techniques can be effectively integrated into domestic STS monitoring workflows.

The structure of the paper is as follows: First, we briefly introduce and discuss the relevant literature on fault detection in STS and TSAD, identifying the research gap and our research contribution (Section 2). We then introduce the methods we use to close this gap (Section 3). Following this, we introduce the Experiments, in particular the dataset and experimental setup, and then show our results (Section 4). In section 5, we interpret these results and discuss the key takeaways of the study as well as possible improvements for the future.

In a nutshell, this paper demonstrates that reconstruction-based anomaly detection methods are both qualitatively and quantitatively successful in fault detection in STS. Our method outperforms simple statistical baselines, such as PCA, as well as both simple and complex DL baselines. Additionally, uncertainty estimation further improves detection performance, with heteroscedastic uncertainty estimation leading to the best detection performance without requiring additional hyperparameters.

2. Background

2.1. Fault detection in solar thermal systems

Fault detection in STS has been approached in various ways, as discussed in several works [23, 11]. These approaches can be broadly categorized into rule-based and expert systems, modelling or parity-space approaches, and data-driven approaches, with some overlap between these categories. In practice, operational fault detection often combines elements of these approaches, leveraging domain expertise and automated detection tools to identify and diagnose failures.

Rule-based and expert systems. Rule-based and expert systems detect faults by checking if a system's behaviour deviates from predefined rules. These rules are crafted using detailed system knowledge to model specific failure cases individually tailored to the system type. As a key advantage, they often directly classify the specific fault whenever a rule is violated. A simple example is sensor fault detection: if a sensor produces highly implausible outputs, such as jumping several orders of magnitude outside its dynamic range, it is considered a faulty sensor. However, this approach only detects anticipated failures encoded in the rules. Any anomalies or failures not considered during rule creation will go undetected. Expert and rule-based systems have been employed, for example, in the IP-Solar project [9, 31, 19], the FeDet research project [26, 40, 14], as well as the Solar-Check project [39, 15].

Modelling approaches. Modelling approaches, including parity-space and physics-based methods, focus on replicating system dynamics through simulations or analytical equations. This is done either through simulation tools, simulating the expected behaviour of the system [22], or by applying physical equations to estimate the yield of the collector loop directly [32, 33]. Large deviations from the estimations are then flagged as faults. These approaches thus detect any unexpected faults or service interruptions without specifying the fault type beforehand and are more robust to unexpected behaviour. However, unlike the rule-based system, the faults are not immediately classified but require a secondary approach or manual system analysis. Additionally, modelling approaches often require detailed knowledge of the specific system design and supplementary sensors, further constraining their applicability.

Data-driven approaches. Data-driven approaches do not rely on explicit physics-based models or rules; instead, they learn patterns directly from data. Early examples in STS used small, simple neural networks to estimate sensor readings, relying on residuals between predictions and measurements to identify anomalies [20, 21, 13]. Some recent works have extended these concepts to recurrent neural network architectures and slightly more complex, modern deep learning architectures [5]. Additionally, there has been work that uses operational data from large STS to train tree-based models, similar to previous work, which uses tree regression models to estimate sensor values based on both historical data and data from highly correlated sensors [12].

Data sources and availability All of these works have been trained on nominal data and then evaluated on both nominal and faulty data. The data sources of these works have been simulations and pilot projects, which have very controlled setups and defined layouts. Typically, the faults detected were either induced by the pilot project operators or programmed into the simulation. Additionally, the models employed were for a single specific STS layout. They may struggle to generalize across different installations and sensor layouts, which is frequently encountered in domestic

STS, where configurations vary widely across systems. In contrast, we recently published a publicly available dataset of operational domestic STS, featuring a wide variety of system realizations and faults [10].

2.2. Time Series Anomaly Detection

Deep learning in TSAD Time series anomaly detection in deep learning initially centered on autoencoders or forecasting models, building on their success in related tasks, such as computer vision and speech recognition. Early efforts focused on embedding modern architectures into these frameworks, including VAEs with long short-term memory (LSTM) -based encoding and decoding [35], time-dependent latent spaces, and normalizing flows [42]. Seeking further improvements, subsequent studies explored more complex strategies, such as adversarial training [2], novel transformer-based designs [45], modified attention mechanisms [48], graph neural networks [7], and contrastive learning [6]. These approaches all assign anomaly scores to individual points or sections of a time series, which can then be either directly analysed or thresholded for automatic classification. While there are many different ways to generate these anomaly scores based on the used architecture, forecasting- and reconstruction-based methods usually take a distance in data space, such as the mean-squared error, as an anomaly score.

Critique of TSAD literature However, as previously mentioned, recent investigations have highlighted fundamental issues in the TSAD literature. Commonly used benchmarks often contain noisy or mislabeled faults, some of which are trivial or impossible to detect, thus skewing model performance. Furthermore, other biases are present in commonly used datasets, such as the run-to-failure bias and unreasonable anomaly densities [47, 28, 38]. There are also no clearly established guidelines for analysis and preprocessing. For example, Sarfraz et al. [38] reported some studies using 127 and other studies using 112 features on the WADI dataset, which itself reports to only have 125 features. In addition, the metrics favoured in the field, specifically the point-adjusted F1 score, tend to reward continuous or noisy predictions, with some studies showing almost perfect results for randomly chosen anomaly scores [24]. Other approaches to replace problematic metrics, i.e., segment-based alternatives [43], and workflows in benchmarks have either failed to achieve large-scale adoption or introduced new problems.

Simple models outperform complex models When restricting the analysis to the least problematic benchmarks and correcting for evaluation flaws, recent publications have found simpler methods to consistently outperform more elaborate models. Sarfraz et al. [38] found that for all their tested DL architectures, a simplified forecasting method outperformed the established DL method. Furthermore, both these simple methods as well as the complex DL architectures were outperformed by a PCA-based reconstruction method using simple error rescaling. Other research, such as the recent benchmark dataset by Liu and Paparrizos [28], also found classical machine learning methods as well as simpler DL methods outperform complex DL methods both for univariate and multivariate time series. This performance gap was especially pronounced in the univariate case.

2.3. Research Gap

As previously stated, despite the abundance of raw sensor data, previous research on fault detection in STS has been constrained to either synthetic or proprietary datasets, as manufacturers are often unwilling or unable to share operational data. Only the recently published PaSTS dataset [10] is publicly accessible for operational domestic STS. This dataset allows us to extend the fault detection literature in STS by a data-driven approach that leverages the insights of the recent DL-TSAD literature.

As we aim to stick closely to the established workflow from Ebmeier et al. [10], we introduce several constraints. First, this is operational data from heterogeneous, non-standardized domestic STS, requiring methods with strong generalisation. To ensure this, we validate and test only on entirely unseen systems. Second, we focus on post-hoc system analysis, rather than real-time control. This enables purely offline evaluation and aligns with existing fault-diagnosis workflows. As these workflows interpret anomalies within a broader system context, we analyse full-day sequences rather than shorter segments, localising anomalies at this coarser granularity.

We adopt a semi-supervised anomaly detection setup, common in TSAD, where training occurs only on nominal data. In this semi-supervised setup, the model is trained exclusively on data representing nominal system behaviour. Anomalies are then identified as deviations from this learned distribution during inference. This approach aligns with practical constraints, where labelled faults are scarce, unavailable or unreliable. We implement a straightforward LSTM-based VAE, informed by the recent TSAD literature. Building on findings that error rescaling improves detection by implicitly modelling uncertainty, we extend the VAE with probabilistic reconstructions, comparing homoscedastic and heteroscedastic uncertainty estimation.

3. Methods

3.1. Reconstruction-based anomaly detection

Reconstruction-based anomaly detection relies on deriving a regularized reconstruction \hat{x} of the input data x , then evaluating both via a scoring function. This function produces an anomaly score $A = f(x, \hat{x})$, which can be thresholded to classify samples as nominal or anomalous or be used directly to inform an engineer about the system's state.

Historically, such scoring functions focused on moment matching; for instance, the squared error $(x - \hat{x})^2$ or the absolute error $|x - \hat{x}|$. These remain popular in recent deep learning approaches, sometimes with rescaling techniques [38, 7] to incorporate homoscedastic uncertainty. However, moment-based methods alone do not fully exploit the probabilistic nature of our VAE framework.

Since our training set is assumed to be both anomaly-free and diverse enough to capture the full range of nominal operational conditions, identifying anomalies becomes akin to identifying out-of-distribution (OOD) samples. A well-trained VAE can learn a calibrated distribution of nominal data. Thus, if a given trace x has a low likelihood under our learned model \hat{x} , it is unlikely to belong to the nominal (in-distribution) regime. Consequently, we measure anomalies using the negative log-likelihood (NLL). Larger NLL values indicate a lower probability of x under the reconstructed distribution, hence a higher degree of anomaly.

In practice, one can determine a suitable threshold on the NLL by examining validation data or specifying acceptable risk levels. This threshold then defines a boundary between nominal and anomalous behaviour, providing a direct, probabilistically grounded metric for fault detection in STS. Additionally, one can use the anomaly scores to manually determine trends and the system's state.

3.2. Reconstruction-based Model

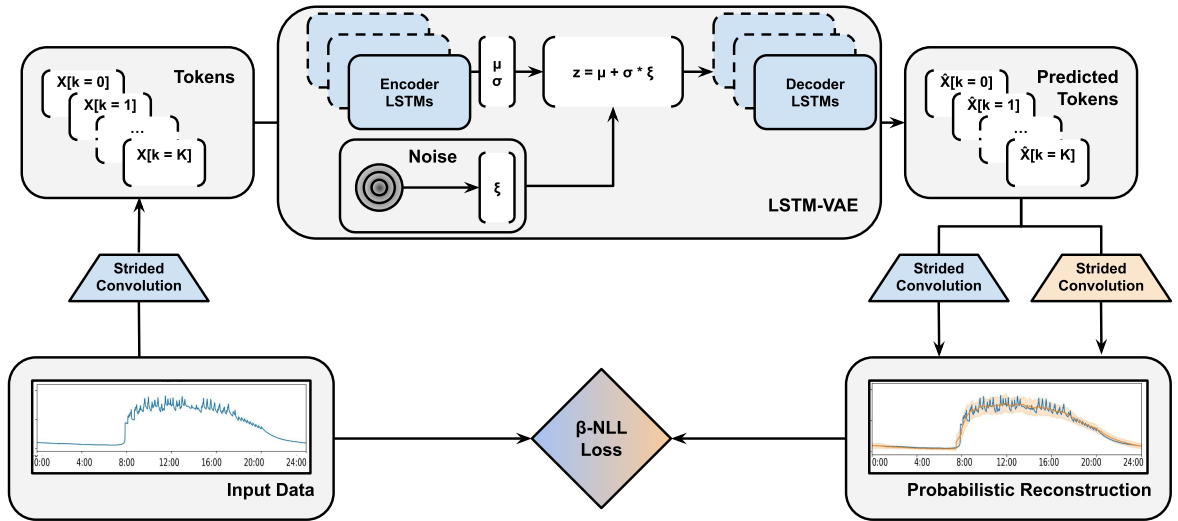


Figure 1: Schematic overview of our LSTM-based VAE for anomaly detection. The input time series is tokenized into fixed-length segments and passed through an encoder LSTM, which outputs parameters of a Gaussian latent distribution. A latent sample drawn from this distribution is then fed into a decoder LSTM, which reconstructs the tokens and maps them to a Gaussian distribution, approximating the original input data.

Figure 1 shows the core idea of an LSTM [18] VAE [25] that reconstructs tokenized sensor data. Similar work was previously performed on an online anomaly detection task in robotics [35], while other works have used similar but more complex strategies [42].

3.2.1. Variational Autoencoder

VAEs [25] are probabilistic models widely used in anomaly detection, particularly when the goal is to learn a representation of normal operating conditions and subsequently detect deviations. VAEs extend the standard autoencoder

formulation, which learns a deterministic mapping from data to a bottleneck layer and back, by incorporating a (typically Gaussian) prior distribution over the latent space.

VAEs consist of two main components: an encoder that maps input data \mathbf{x} to a latent representation \mathbf{z} , and a decoder that reconstructs $\hat{\mathbf{x}}$ from samples drawn in the latent space. Formally, the encoder outputs a mean and variance that parameterise a Gaussian distribution over \mathbf{z} :

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\sigma}_\phi^2(\mathbf{x})), \quad (1)$$

while the decoder models

$$p_\theta(\hat{\mathbf{x}}|\mathbf{z}) = \mathcal{N}(\hat{\mathbf{x}}; \boldsymbol{\mu}_\theta(\mathbf{z}), \boldsymbol{\sigma}_\theta^2(\mathbf{z})). \quad (2)$$

Training proceeds by minimizing a loss that balances reconstruction quality (how accurately the decoder reproduces \mathbf{x}) and a Kullback-Leibler (KL) divergence term that encourages $q_\phi(\mathbf{z}|\mathbf{x})$ to remain close to a chosen prior (often $\mathcal{N}(\mathbf{0}, \mathbf{I})$). An additional hyperparameter β is introduced to control the trade-off between reconstruction quality and latent space regularization in the loss. This results in the β -VAE loss function [17], given by:

$$\mathcal{L} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta \cdot \text{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})), \quad (3)$$

The first term, $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})]$, represents the expected reconstruction quality. In contrast, the second term, $\text{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z}))$, measures the divergence between the approximate posterior and the prior distribution. Tuning β determines the regularization strength of the model, with $\beta = 1$ recovering the standard VAE formulation, and a larger β parameter leading to stronger restrictions on the latent space.

The reconstruction term is the NLL of the data, assuming a Gaussian data distribution, it is defined as follows.

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})] = \frac{1}{2} \left(\log \sigma^2 + \frac{\mathbf{x} - \boldsymbol{\mu}}{\sigma^2} \right) + \text{const} \quad (4)$$

Where $\boldsymbol{\mu}$ is the mean of the estimated distribution $\hat{\mathbf{x}}$ and σ^2 its variance, the commonly used mean squared error (MSE) reconstruction loss is recovered when we assume homoscedastic variance. However, since we are interested in heteroscedastic uncertainty estimation, we formulate our model to predict both the σ^2 term and the mean. This is also known as mixture density networks [3].

As noted by Seitzer et al. [41], training with the standard NLL term can lead to issues with undersampling and generally poor training dynamics. To counter this, we employ their variation of the NLL, the Beta negative log-likelihood (BNLL). In the BNLL, the distance is scaled with $\sigma^{2\beta_L}$, which simplifies to the standard deviation σ for the proposed standard value of $\beta_L = 0.5$. The BNLL is equivalent to the NLL for $\beta_L = 0$ and recovers the MSE for $\beta_L = 1$.

3.2.2. Data Encoding and Decoding

Encoding and decoding the time series occur in two main steps, inspired by tokenization strategies used in computer vision [8]. Since each daily trace contains 1440 timesteps (one timestep per minute, 1440 minutes per day) and consecutive timesteps often exhibit strong autocorrelation, feeding the entire sequence at once can lead to unstable training dynamics and other problems, resulting in deteriorating performance. Instead, we chunk the data into fixed-length tokens to reduce sequence length while retaining information content.

Specifically, a day's data consists of T timesteps across F features, which we reshape into K chunks of length L ($K \times L = T$). Each chunk is then mapped by a linear layer into an embedding of size E , producing K tokens

$$\mathbf{X} \in \mathbb{R}^{T \times F} \xrightarrow{\text{Reshape}} \mathbf{X}' \in \mathbb{R}^{K \times L \times F} \xrightarrow{\text{Linear}} \mathbf{Z} \in \mathbb{R}^{K \times E}.$$

Here, each token represents a segment of L timesteps for a given day. Notably, the original ordering of tokens is maintained, preserving the temporal structure.

Once the data has been processed into a series of tokens, we use LSTM networks [18] to further process this information and map it to a latent representation. For simplicity of notation, we will now assume that the number of hidden units in the LSTM is equal to the embedding dimension. After processing with the LSTM, the hidden units of

the last layer are then linearly mapped to parametrize D Gaussian distributions per token, from which we sample to get a latent vector.

$$\mathbf{Z} \in \mathbb{R}^{K \times E} \xrightarrow{\text{LSTM}} \mathbf{H} \in \mathbb{R}^{K \times E} \xrightarrow{\text{Linear}} \mathbf{\Theta} \in \mathbb{R}^{K \times D \times 2} \xrightarrow{\text{sample}} \mathbf{\Theta}' \in \mathbb{R}^{K \times D}.$$

The first dimension of $\mathbf{\Theta}$ represents the mean of each Gaussian distribution, and the second dimension represents its variance. This retention of a time-dependent latent space is one of the key findings in the Omnianomaly architecture [42] and also proved beneficial for stable training dynamics.

Decoding reverses the process. The latent vector is linearly mapped to the embedding dimension and processed using the decoder LSTM. A single-layer multi-layer perceptron (MLP) then produces a mean and standard deviation for each feature and timestep within each token. Finally, we reshape the tokens to restore the original daily format, yielding a probabilistic reconstruction of size $T \times F \times 2$:

$$\mathbf{\Theta}' \in \mathbb{R}^{K \times D} \xrightarrow{\text{Linear}} \mathbf{H}' \in \mathbb{R}^{K \times E} \xrightarrow{\text{LSTM}} \mathbf{H}'' \in \mathbb{R}^{K \times E} \xrightarrow{\text{MLP}} \mathbf{R} \in \mathbb{R}^{K \times (L \cdot F) \times 2} \xrightarrow{\text{reshape}} \hat{\mathbf{X}} \in \mathbb{R}^{T \times F \times 2}.$$

Here, the first channel of $\hat{\mathbf{X}}$ corresponds to the reconstructed mean of each feature at each timestep, while the second channel represents the associated variance, providing a probabilistic estimate of the underlying dynamics.

3.3. Uncertainty Estimation

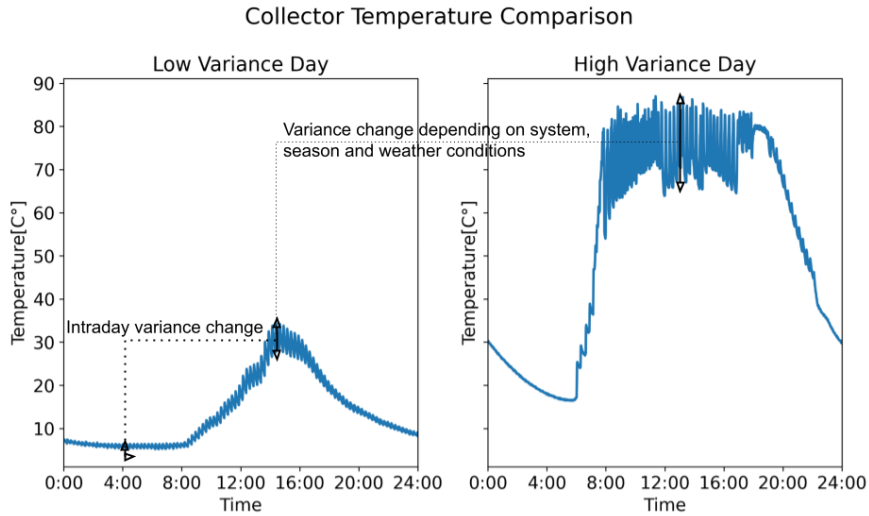


Figure 2: Collector temperature for a high variance and low variance day of the same system. Illustrating the large difference in variance both within a day and between different days.

Sensor data in STS exhibit variability that depends on both intrinsic (sensor-specific) and extrinsic (state-dependent) factors. For example, the collector's temperature during the day has a broader dynamic range than at night. This variability can further differ based on the system's state and seasonal changes, as shown in Figure 2. Capturing this variability accurately is critical for identifying anomalies, as it enables differentiating between normal fluctuations and genuine faults.

To account for these types of variability, we consider two uncertainty estimation approaches. The first, **homoscedastic uncertainty estimation**, assumes variability is independent of the data trace and relies solely on the feature-time index. The second, **heteroscedastic uncertainty estimation**, accounts for data-dependent variability arising from system state, seasonal effects, or other external factors.

3.3.1. Homoscedastic Uncertainty Estimation

Homoscedastic uncertainty estimation can be implemented as a post-processing step. Here, the error vector $E_{i,t}$ for sensor i and time step t is calculated using the observed data \mathbf{x} and its reconstruction $\hat{\mathbf{x}}$:

$$E_{i,t} = |x_{i,t} - \hat{x}_{i,t}|. \quad (5)$$

The error vector $E_{i,t}$ can then be normalized using z-normalization or the inter-quartile transform:

$$\hat{E}_{i,t} = \frac{E_{i,t} - \mu_{i,t}}{\sigma_{i,t}}, \quad (6)$$

where $\mu_{i,t}$ is the mean and $\sigma_{i,t}$ is the standard deviation over the training set for z-normalization, or $\mu_{i,t}$ is the median and $\sigma_{i,t}$ is the inter-quartile range for the inter-quartile transform [38, 7]. This approach assumes a constant, data-independent variance for every feature-time combination, thereby aligning with the definition of homoscedastic uncertainty. When training a model that gives a deterministic output, we use the standard mean squared error loss.

3.3.2. Heteroscedastic Uncertainty Estimation

While post-processing methods provide a simple way to estimate uncertainty, they fail to capture data-dependent variability. For example, Figure 2 demonstrates that the dynamic range of collector temperature varies substantially across different system states and seasons. To address this, we incorporate uncertainty estimation into the model by using a mixture density network (MDN) [3]. In this approach, the model predicts both the mean and variance for each feature and timestep from the input data, thereby learning heteroscedastic (data-dependent) variance.

An MDN-based approach can adapt to evolving conditions, such as seasonal shifts or changing operational states, by providing a probabilistic reconstruction at each timestep. This is particularly relevant for domestic STS, where day-to-day behaviour can vary widely depending on ambient conditions, system state, and usage patterns. We employ a mean-variance approach for simplicity, allowing the model to output a Gaussian parameterized by (μ, σ) per timestep and feature, without requiring any additional hyperparameters. When training a model outputting a Gaussian distribution, it is necessary to choose an appropriate loss function. As discussed in Section 3.2.1, we choose the BNLL here for improved training dynamics. Note that we only use the BNLL for training the model and not for calculating the anomaly scores, where we use the standard NLL.

4. Experiments

4.1. Dataset

We use the PaSTS dataset [10], consisting of 83 different STS with varying installations and operational durations, for a total of 39878 days (corresponding to app. 110 system-years). The data is taken from the manufacturer's, Ritter Energie GmbH & Co. KG, service team and thus contains many anomalies. Only a few systems are considered faultless or mostly faultless under normal operational conditions. Four data types are provided: index, sensor, control signals, and status. The index data specify the system and timestep, sensor data records direct readings, and control signals and status data document the controller's behaviour, including outputs from an internal expert system. All measurements are at a one-minute resolution. For a more detailed summary, see Appendix C.

Dataset reannotation The dataset contains data from multiple types of installations, various old and new controllers, fault detection algorithms, and system types. Certain systems exhibit multi-year gaps in data collection or have undergone renovations and modifications over time. Because this dataset primarily reflects service operations, many annotations from the built-in fault detection system are either incomplete or inaccurate, as properly identified and resolved faults do not require additional analysis by the service team and are handled by technicians or system operators. Consequently, we treated these annotations as unreliable. In collaboration with experts from the manufacturer, some data were reannotated by examining sensor readings and comparing them against baseline methods and our anomaly detection models. These reannotations only recast days initially labelled as fault-free into fault conditions whenever experts identified likely, yet initially unrecognized, anomalies. Not all systems were revisited, and false annotations remain. However, the most problematic examples, which severely distorted the summary statistics, have been corrected.

Dataset split along systems We can separate our data into traces of a single day, with a total of 1440 minutes of data per trace. From an operational perspective, we assume that a fault detection method should be able to handle unseen systems from their first day of data onward. We separate training, validation, and test data in our dataset along different systems, such that the entire evaluation is always done on unseen data traces from unseen systems. As previously discussed, our model is trained only on nominal data, thereby constraining our training set to contain systems in a nominal operational state, with all potential anomalies and faults filtered out. We then separate the dataset into training, validation and test set, with the training and validation set only containing mostly faultless systems and the test set containing systems with both nominal and faulty behaviour. The split into these different sets is always done along systems, such that a system is always fully contained in only one of the three sets. The splits can be found in Appendix B.

4.2. Baselines

We implement several baselines to compare our work to. We will use the PCA reconstruction method as demonstrated by Sarfraz et al. [38], an LSTM-based VAE utilising homoscedastic uncertainty estimation as well as several baselines as implemented in the recent benchmark publication by Liu and Paparrizos [28].

PCA Reconstruction As previously discussed, Sarfraz et al. [38] have shown that PCA can outperform the state-of-the-art deep learning models, while being very easy to tune and train. Analog to their implementation, we fit the principal components on the training set. Then we calculate the error vector between the data and the reconstructed data using the fitted PCA and fit a z-transformation of this vector, as described earlier. On the test set, we transform the data according to the fitted PCA model, reconstruct it to calculate the error vector, and then rescale the error vector using the fitted z-transformation. The average of the error vector then yields a single anomaly score per day. Later, we refer to this method as Rescaled PCA-R and the method without z-transforming the error vector as Unscaled PCA-R. A key aspect of this PCA reconstruction method is that it involves only a single hyperparameter, namely the number of principal components used for reconstruction. To compare the ease of use, we sweep this hyperparameter and evaluate the difficulty of optimisation. We then compare it to the parameter sweep of β of the main model and discuss the optimisation difficulty for the main model.

Benchmark baselines As additional baselines, we utilize the implemented baselines from the recent benchmark publication by Liu and Paparrizos [28]. As part of their work, they implement several multivariate anomaly detection methods. We use the hyperparameters and setup they provide in the GitHub repository accompanying their publication. We then integrate the results into our problem definition, analysing the aggregated anomaly score for an entire day. The implemented methods are several statistical methods, as well as neural network-based methods, based on density, reconstruction, or forecasting. It is worth noting that the two forecasting-based methods (CNN and LSTMAD) are comparatively simple and, according to the findings by Sarfraz et al. [38], we would expect them to perform well. The PCA method implemented in the paper is not a reconstruction-based method but a density-based method within principal component space. Furthermore, we update the TranAd [45] implementation provided in the GitHub repository and remove the Omnianomaly [42] implementation, as it lacked essential components of the original implementation.

4.3. Experimental Setup

We train our models using only nominal STS, as identified in Appendix B, and validate them on a separate nominal dataset. The input variables include five temperature sensors (TSA1, TSE, TW, TSV, TAM), a volumetric flowrate sensor (VF), and two control signals (pwm for pump operation and ctr for frost protection). Temperature sensors undergo z-normalization based on training statistics, while the remaining sensors are rescaled to (0, 1) and smoothed with a 15-minute Gaussian window to stabilize training. Further details on the data distribution can be found in Ebmeier et al. [10].

A set of practical hyperparameters is chosen, as summarized in Table A1, and the model is trained on a single NVIDIA A100 GPU for 40,000 update steps. The primary hyperparameter sweep concerns the latent space regularization factor β . We evaluate the model quantitatively using 4 different metrics. We use the standard optimal F1 score, AUC-PR and AUC-ROC, as it is commonly used in time series anomaly detection (compare i.e. [38, 28]). Additionally, we calculate the system-wise F1 score, by taking the threshold leading to the optimal threshold for all systems except the one we look at and then applying this threshold to the individual system. We then calculate the F1 score per system and average the F1 score over all systems. In the following we will refer to this as the system-wise F1 score. This would be the metric, which mirrors our application the most, as it quantifies the detection performance of a new unseen system being evaluated. For a short discussion of the metrics, see Appendix F.

Our evaluation will then focus on three main hypotheses.

1. Reconstruction-based anomaly detection can adequately detect faults in STS.
2. Our method can match or exceed the anomaly detection performance of both simple baselines, such as PCA, and more complex established DL methods. This would be surprising given the findings of Sarfraz et al. [38].
3. Uncertainty estimation improves anomaly detection performance. Here, we want to evaluate specifically whether heteroscedastic uncertainty estimation is superior to homoscedastic uncertainty estimation.

4.4. Quantitative Results

Table 1 compares the performance of (Our Model) against several baselines. Some example reconstructions by Our Model can be seen in Appendix D. We evaluate all methods on both the original dataset annotations (see Table A2) and the reannotated dataset, which incorporates expert feedback from the manufacturer (see Table 1). Besides listing each

Table 1

Comparison of anomaly detection performance on the reannotated dataset. Values are the mean over all seeds \pm the standard deviation. For non-stochastic methods– like PCA– no standard deviation is given. Baselines are as discussed in Section 4.2. Baselines marked with † are forecasting-based deep learning models and forecast a single step. Best performing models and models without a statistically significant difference to the best performing model are marked in bold.

Method	System-wise F1	Optimal F1	AUC-PR	AUC-ROC
Unscaled PCA-R	0.30	0.69	0.74	0.86
Rescaled PCA-R	0.40	0.76	0.82	0.91
LSTM-VAE	0.32 \pm 0.02	0.72 \pm 0.01	0.75 \pm 0.01	0.89 \pm < 0.01
Our Model	0.46 \pm 0.02	0.77 \pm 0.01	0.79 \pm 0.01	0.90 \pm < 0.01
HBOS [16]	0.15	0.50	0.52	0.74
LOF [4]	0.29	0.48	0.47	0.77
KMeansAD [49]	0.24 \pm 0.02	0.57 \pm 0.01	0.62 \pm 0.02	0.82 \pm 0.01
PCA [34]	0.38	0.75	0.79	0.88
IForest [27]	0.15	0.57	0.60	0.75
AnomalyTransformer [48]	0.29 \pm 0.02	0.36 \pm 0.04	0.33 \pm 0.06	0.64 \pm 0.05
AutoEncoder [37]	0.24 \pm < 0.01	0.30 \pm < 0.01	0.22 \pm < 0.01	0.57 \pm < 0.01
CNN [30] †	0.36 \pm 0.01	0.75 \pm 0.01	0.78 \pm 0.01	0.88 \pm < 0.01
Donut [44]	0.30 \pm 0.03	0.69 \pm 0.01	0.71 \pm 0.01	0.84 \pm 0.01
LSTMAD [29] †	0.30 \pm 0.01	0.65 \pm 0.02	0.71 \pm 0.02	0.84 \pm 0.01
TimesNet [46]	0.25 \pm 0.01	0.36 \pm 0.01	0.38 \pm 0.01	0.56 \pm 0.01
TranAd [45]	0.32 \pm 0.02	0.68 \pm 0.03	0.72 \pm 0.03	0.85 \pm 0.01
USAD [2]	0.25 \pm 0.02	0.37 \pm 0.02	0.29 \pm 0.01	0.67 \pm 0.01

method’s optimal F1 score and the previously defined system-wise F1 score, we report the area under the precision-recall curve (AUC-PR) and the area under the receiver operator characteristic (AUC-ROC).

The Unscaled PCA-R approach serves as a simple baseline, achieving a system-wise F1 score of 0.3 and an optimal F1 score of 0.69. Rescaling the error vector (Rescaled PCA-R) leads to a substantial increase in performance over all metrics, outperforming all other baselines that we compare to. This indicates, that even using homoscedastic uncertainty estimation greatly improves performance and aligns with previous studies, which have also observed a clear performance improvement when employing this method [38, 7].

Transitioning from PCA to a mean-only VAE (LSTM-VAE) yields an improved performance level over the Unscaled PCA-R baseline. However, even though this method also used the error-rescaling, it did not improve performance substantially over the unscaled method, falling behind Rescaled PCA-R.

However, when comparing to Our Model, a more substantial gain becomes apparent. As the model also estimates variance, the anomaly score is computed using the NLL score here. Our model achieves an optimal F1 score of 0.77 and a system-wise F1 score of 0.46. This was especially true in the application-focused system-wise F1 score, Our Model outperforms all other models by a substantial margin. This mirrors our approach, where we focus specifically on generalisation capabilities. Furthermore, our model substantially outperforms all the baselines we adapted from Liu and Paparrizos [28] on all metrics.

In Figure 3, we show the F1 score of Our Model and the Rescaled PCA-R model on all systems. This is the exact case, from which we calculate the system-wise F1 score by averaging over all systems. The systems are sorted by their F1 score w.r.t. each method for better visualisation. Figure 3 illustrates the good performance of Our Model, when evaluating the model on a completely unseen system and shows what performance level to expect. For a more detailed comparison between Our Model and the second best model (Rescaled PCA-R) as well as additional qualitative analysis, see Appendix E.

Overall, these results are in relatively good agreement with our key hypotheses. We find that reconstruction-based anomaly detection gets solid quantitative results on our dataset. Our Model clearly outperforms the baselines, which include reconstruction-based, forecasting-based, and density-based models. Additionally, both heteroscedastic and homoscedastic uncertainty estimation substantially enhances anomaly detection performance. Additionally, Our Model has the strongest performance gain in the metric that aligns closest with the application. While the model performs comparably across many systems, there are some systems where our model strongly outperforms all baselines. We

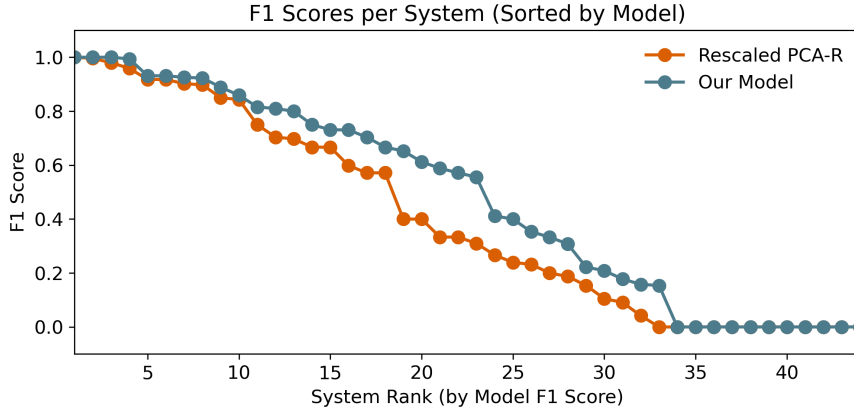
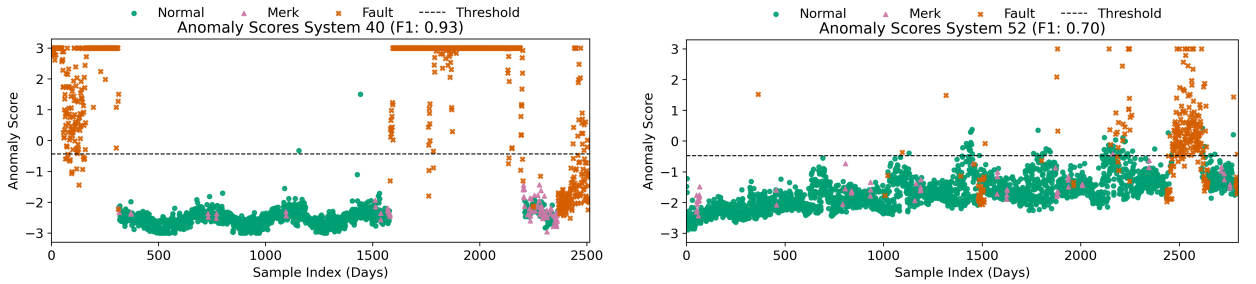


Figure 3: Performance of Our Model and the Rescaled PCA-R model on different systems. The systems are sorted by performance of that model. The systems are sorted by their F1 score, and on the x-axis is the rank of the model. On the y-axis is the F1 score the model achieves on the corresponding system. The average of these is the system-wise F1 score. Note that the system ranks can differ between the two models.

show this more clearly in the comparison to the Rescaled PCA-R in Appendix E. The performance of our model compared to the LSTM-VAE also demonstrates that, in this case, heteroscedastic uncertainty estimation may be necessary for reconstruction-based anomaly detection.

4.5. Qualitative Results



(a) System 40 exhibits a long, previously undetected fault, leading to reduced yield and eventual degradation of multiple components.

(b) System 52 shows no clear boundary between fault and nominal states, though the anomaly score indicates a run-to-failure trend near the end of its life cycle.

Figure 4: Qualitative results for two sample systems. The x-axis represents the day index, and the y-axis is the anomaly score (values above 3 are capped for visibility). Individual dots denote a day with a fault annotation (Fault), a potential anomaly marked by the expert system (Merk) or nominal behaviour (Normal) according to its colour.

To validate our quantitative findings, we investigate two example systems in the dataset that are known to alternate between nominal and faulty behaviour. In Figure 4a, we show System 40 from the dataset. The x-axis represents the index of the days, while the y-axis displays the anomaly score, capped at 3 for clearer visualization. Days without any fault annotation, with a potential anomaly, or with nominal behaviour are marked with their respective colours in the figure. A clear separation emerges between nominal and faulty behaviour. After a manual analysis of this system, in collaboration with the manufacturer, we discovered a previously undetected, long-persisting fault that lasted over 600 days (around day indices 1 529 to 2 130). This fault further degraded the system, leading to multiple faults towards the end of the time series. Our model thus clearly contributed to uncovering these hidden issues and provided valuable insights into the system's state.

An additional example can be seen in Figure 4b, illustrating System 52. Here, the separation between nominal and faulty states is less pronounced. Nonetheless, the system exhibits a run-to-failure pattern, with more anomalous or

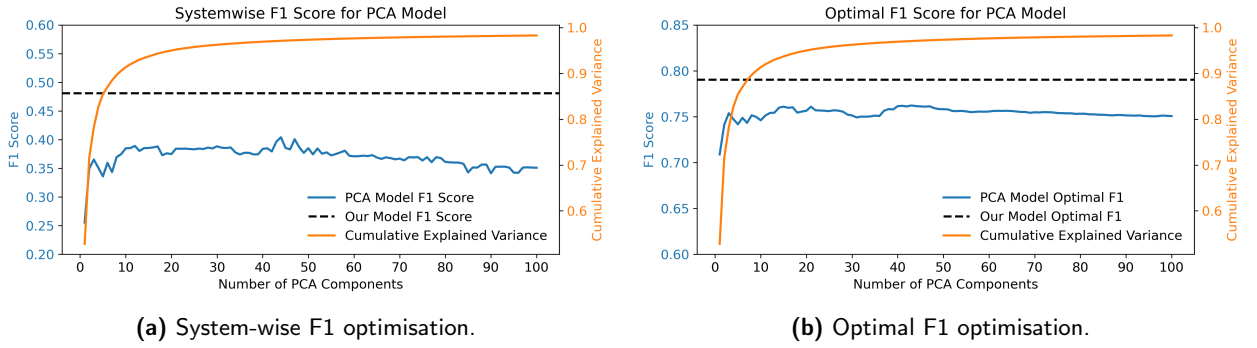


Figure 5: Impact of the number of principal components on anomaly detection performance. The x-axis depicts the number of components, and the y-axis shows both the resulting F1 score and the cumulative variance explained by those components.

faulty days towards the end of its lifecycle. A qualitative inspection of the anomaly score time series still helps reveal the gradual degradation into an anomalous state, even if automated detection does not yield perfect results for all days. Hence, visual inspection of the anomaly score remains a valuable tool for monitoring the system’s status and has proven informative in our analysis of the dataset.

In Appendix E, we compare the qualitative performance of Our Model with Rescaled PCA-R. Notably, we observe qualitatively that the early detection performance, i.e., through the observed trend in the anomaly score, is absent in Rescaled PCA-R (compare Figure A3 and Figure A4). Additionally, we observe that Rescaled PCA-R fails on some systems, while Our Model achieves excellent performance (compare Figure A5, Figure A6, and Figure A7). This mirrors the performance of Our Model in the system-wise F1 metric. Given the usefulness of the anomaly score itself, Our Model is currently being implemented into the manufacturer’s service team workflows.

4.6. Optimisation Complexity

Figure 5 illustrates how the single hyperparameter in the Rescaled PCA-R baseline, the number of principal components, affects anomaly detection performance. The x-axis shows the number of components, while the y-axis tracks both the F1 score and the cumulative variance explained. Notably, identifying this near-optimal range requires under 30 minutes of runtime, even with unoptimized code. As shown, using only three principal components already yields an optimal F1 score of 0.75, with the maximum at 0.76. This simple parameter sweep underscores how straightforward it is to tune PCA in practice, making it unlikely to miss the optimal setting.

By contrast, our VAE-based approach introduces multiple hyperparameters and more intricate engineering requirements. Figure 6 illustrates how just one of these hyperparameters, the latent space regularization parameter β , influences the F1 score. Each dot corresponds to a single seed training run, while the red line marks the best Rescaled PCA-R baseline. Even for the same β value, the range of observed F1 scores is considerably larger than the variation in Rescaled PCA-R. In many cases, training a VAE multiple times with the same hyperparameter configuration is necessary to obtain consistent results. Additionally, the position of the shown optimum can be affected by other hyperparameters such as the size of the latent space, making optimisation over multiple parameters even more compute- and time-intensive.

We also note that stable training for long time series relies on careful weight initialization, normalization layers, and other deep learning techniques that may not be immediately familiar to engineers without machine learning expertise. In contrast, the Rescaled PCA-R baseline requires none of these specialized considerations or high-end hardware. While deep learning models can surpass simple baselines, increasing model complexity increases both the engineering overhead and computational cost, emphasizing the trade-off between potential performance gains and ease of deployment.

5. Conclusion

In this work, we pose three key questions: Can reconstruction-based anomaly detection effectively identify faults in STS? Can VAEs match or surpass simple and complex baselines in time series anomaly detection? Can uncertainty estimation boost fault detection performance? Our qualitative findings show distinct separations between nominal and faulty states, even when faults persist for hundreds of days. These observations are further confirmed by the

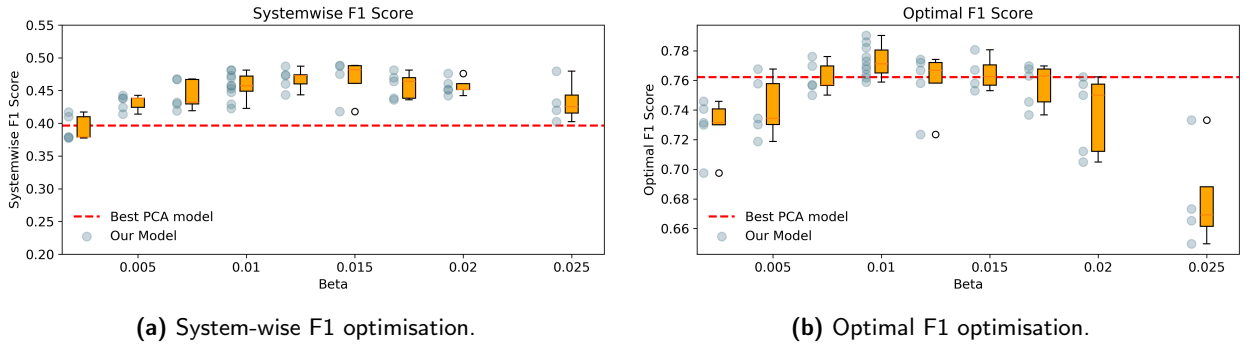


Figure 6: Impact of the latent space regularization parameter β on the F1 score. Each dot represents a single training run, highlighting how both hyperparameter choices and initialization affect model performance. The box extends from the first quartile to the third quartile. The line inside the box corresponds to the median. The whiskers extend from the box to the largest datapoint within 1.5 inter-quartile ranges from the box. The flier points are the points beyond the ends of the whiskers.

manufacturer's experts, who indicate that a day-by-day reconstruction-based workflow can reliably uncover anomalies in real operational data. This is also the reason that the model is currently being implemented into the service workflows of the manufacturer.

Quantitative results reinforce this view, demonstrating that our model outperforms simple and complex baselines, with a significant portion of its performance attributed to explicitly modelling uncertainty. Especially in the application-focused metric—the system-wise F1—we get the strongest results, highlighting the generalisation capabilities of Our Model. This is a surprising result given the findings by Sarfraz et al. [38] and highlights that deep learning, if properly configured, can outperform simpler linear approaches. Nonetheless, Rescaled PCA-R remains a robust baseline that often achieves good performance with minimal tuning.

Our evaluation shows the benefits of modelling heteroscedastic uncertainty within a VAE. Incorporating heteroscedastic uncertainty substantially improves detection and enables the model to adapt to varying system states and seasonal changes. Notably, these probabilistic reconstructions do not increase the overall model complexity, whether implemented via homoscedastic or heteroscedastic strategies.

At the same time, deep learning architectures demand considerable engineering effort. Multiple hyperparameters can critically affect performance. Finding an optimal configuration may require extensive parameter sweeps and retraining, which can be prohibitive in practice. In contrast, simple linear methods like PCA often involve only a few key hyperparameters, offering a more straightforward route to acceptable results with minimal domain knowledge.

Consequently, we want to make a case for simple models to increase the performance and usability of anomaly detection models. Particularly, model choices such as the heteroscedastic uncertainty estimation via MDN, which does not add additional hyperparameters. While deep learning can achieve higher performance, the gain comes at the cost of greater engineering overhead and computational expense. Many advanced models in time series anomaly detection, featuring novel architectures and arbitrary parameterizations, might be too complex to feasibly reach their parametric optimum. As a result, they may underperform against baseline or “simplified” methods—an outcome consistent with the results by Sarfraz et al. [38] and Liu and Paparrizos [28]. Thus, it seems that elaborate network architectures that increase complexity can be a hindrance that should be avoided during development. Nevertheless, we want to emphasize that deep learning methods for anomaly detection also enable other time series analysis tasks and may be further extended into various downstream tasks.

In summary, these findings confirm the feasibility of using probabilistic reconstruction-based anomaly detection for STS. Future research should emphasize approaches that balance performance gains with ease of deployment, ensuring that hyperparameter optimization does not become prohibitively complex and that new methods can be assessed on a level playing field against other methods.

Data availability

The data used for this study is available under <https://zenodo.org/records/11093493>, see also [10].

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this work the authors used LLMs(like ChatGPT, Grammarly) for writing assistance, including grammar correction and text polishing. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the published article.

Acknowledgements

This project was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC number 2064/1 - Project number 390727645. The authors thank the international Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Florian Ebmeier. We want to thank Ritter Energie for providing us with the data on which this dataset is based. We want to additionally thank Ritter Energie for their expertise in reannotating the dataset and evaluating our models.

CRedit authorship contribution statement

Florian Ebmeier: Writing – original draft, Software, Data curation, Methodology, Investigation. **Nicole Ludwig:** Supervision, Funding acquisition, Writing – review & editing, Methodology Conceptualization. **Jannik Thuemmel:** Software, Writing – review & editing, Methodology, Investigation. **Georg Martius:** Funding acquisition, Writing – review & editing, Methodology, Conceptualization. **Volker H. Franz:** Supervision, Funding acquisition, Writing – review & editing, Methodology, Conceptualization.

References

- [1] An, J., Cho, S., 2015. Variational autoencoder based anomaly detection using reconstruction probability. Special Lecture on IE 2, 1–18. URL: <http://dm.snu.ac.kr/static/docs/TR/SNUDM-TR-2015-03.pdf>.
- [2] Audibert, J., Michiardi, P., Guyard, F., Marti, S., Zuluaga, M.A., 2020. USAD: Unsupervised anomaly detection on multivariate time series, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, New York, NY, USA. pp. 3395–3404. URL: <https://dl.acm.org/doi/10.1145/3394486.3403392>, doi:10.1145/3394486.3403392.
- [3] Bishop, C.M., 1994. Mixture Density Networks. Technical Report NCRG/94/004. Neural Computing Research Group, Aston University.
- [4] Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J., 2000. LOF: identifying density-based local outliers, in: Proceedings of the 2000 ACM SIGMOD international conference on Management of data, pp. 93–104.
- [5] Correa Jullian, C., Cardemil, J., López Droguett, E., Behzad, M., 2019. Assessment of deep learning algorithms for fault diagnosis of solar thermal systems, in: Proceedings of the ISES Solar World Congress 2019, Santiago, Chile. pp. 1–12. URL: <http://proceedings.ises.org/citation?doi=swc.2019.08.03>, doi:10.18086/swc.2019.08.03.
- [6] Darban, Z.Z., Webb, G.I., Pan, S., Aggarwal, C.C., Salehi, M., 2025. CARLA: Self-supervised contrastive representation learning for time series anomaly detection. Pattern Recognition 157, 110874. URL: <https://www.sciencedirect.com/science/article/pii/S0031320324006253>, doi:10.1016/j.patcog.2024.110874.
- [7] Deng, A., Hooi, B., 2021. Graph neural network-based anomaly detection in multivariate time series. Proceedings of the AAAI Conference on Artificial Intelligence 35, 4027–4035. URL: <https://ojs.aaai.org/index.php/AAAI/article/view/16523>, doi:10.1609/aaai.v35i5.16523.
- [8] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houshy, N., 2021. An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations. URL: <https://openreview.net/forum?id=YicbFdNTTy>.
- [9] Dröschner, A., Ohnewein, P., Haller, M., Heimrath, R., 2009. Modular specification of large-scale solar thermal systems for the implementation of an intelligent monitoring system, in: ISES Solar World Congress 2009, International Solar Energy Society. pp. 682–688.
- [10] Ebmeier, F., Ludwig, N., Martius, G., Franz, V.H., 2024. PaSTS: An operational dataset for domestic solar thermal systems, in: Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems, ACM, New York, NY, USA. pp. 529–534. URL: <https://doi.org/10.1145/3632775.3662159>, doi:10.1145/3632775.3662159.
- [11] Faure, G., Vallée, M., Paulus, C., Tran, T.Q., 2020. Fault detection and diagnosis for large solar thermal systems: A review of fault types and applicable methods. Solar Energy 197, 472–484. URL: <https://www.sciencedirect.com/science/article/pii/S0038092X20300359>, doi:10.1016/j.solener.2020.01.027.

- [12] Feierl, L., Unterberger, V., Rossi, C., Gerardt, B., Gaetani, M., 2023. Fault detective: Automatic fault-detection for solar thermal systems based on artificial intelligence. *Solar Energy Advances* 3, 100033. URL: <http://www.scopus.com/inward/record.url?scp=85168215031&partnerID=8YFLogxK>, doi:10.1016/j.seja.2023.100033.
- [13] Ferreira Garcia, R., Rolle, J.L.C., Castelo, J.P., Gomez, M.R., 2014. On the monitoring task of solar thermal fluid transfer systems using NN-based models and rule based techniques. *Engineering Applications of Artificial Intelligence* 27, 129–136. URL: <https://www.sciencedirect.com/science/article/pii/S0952197613001127>, doi:10.1016/j.engappai.2013.06.011.
- [14] Georgii, M., Schmelzer, C., Braas, H., Orozaliev, J., Vajen, K., 2017. A flexible software framework for self-adapting algorithm-based fault detection and diagnosis in solar heating systems, in: *Proceedings of the ISES Solar World Conference and the IEA SHC Solar Heating and Cooling Conference for Buildings and Industry*, Abu Dhabi. pp. 1–8. doi:10.18086/swc.2017.19.05.
- [15] Georgii, M., Schmelzer, C., Sauer, C., Orozaliev, J., Vajen, K., 2022. Digital representation of heating systems for fault detection purposes, in: *Proceedings of EuroSun 2022*, pp. 1–6.
- [16] Goldstein, M., Dengel, A., 2012. Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm. *KI-2012: poster and demo track 1*, 59–63.
- [17] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A., 2017. beta-VAE: Learning basic visual concepts with a constrained variational framework, in: *International Conference on Learning Representations*. URL: <https://openreview.net/forum?id=Sy2fzU9gl>.
- [18] Hochreiter, S., 1997. Long short-term memory. *Neural Computation* URL: <https://ieeexplore.ieee.org/abstract/document/6795963/>.
- [19] Holter, C., Gerardt, B., Ohnewein, P., Dröscher, A., Feichtner, F., Schgaguler, K., Meißner, E., Luidolt, P., Köstinger, A., Heimrath, R., et al., 2012. Development of the prototype IP-Solar: A web-based monitoring and diagnostics tool for solar thermal systems. *Energy Procedia* 30, 134–143. doi:10.1016/j.egypro.2012.11.017.
- [20] Kalogirou, S., Lalot, S., Florides, G., Desmet, B., 2008. Development of a neural network-based fault diagnostic system for solar thermal applications. *Solar Energy* 82, 164–172. URL: <https://www.sciencedirect.com/science/article/pii/S0038092X07001296>, doi:10.1016/j.solener.2007.06.010.
- [21] Kalogirou, S.A., Mathioulakis, E., Belessiotis, V., 2014. Artificial neural networks for the performance prediction of large solar systems. *Renewable Energy* 63, 90–97. URL: <https://www.sciencedirect.com/science/article/pii/S0960148113004655>, doi:10.1016/j.renene.2013.08.049.
- [22] de Keizer, C., Kuethe, S., Jordan, U., Vajen, K., 2013. Simulation-based long-term fault detection for solar thermal systems. *Solar Energy* 93, 109–120. URL: <https://www.sciencedirect.com/science/article/pii/S0038092X13001308>, doi:10.1016/j.solener.2013.03.023.
- [23] de Keizer, C., Vajen, K., Jordan, U., 2011. Review of long-term fault detection approaches in solar thermal systems. *Solar Energy* 85, 1430–1439. URL: <https://www.sciencedirect.com/science/article/pii/S0038092X11001083>, doi:10.1016/j.solener.2011.03.025.
- [24] Kim, S., Choi, K., Choi, H.S., Lee, B., Yoon, S., 2022. Towards a rigorous evaluation of time-series anomaly detection, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 7194–7201. URL: <https://ojs.aaai.org/index.php/AAAI/article/view/20680>.
- [25] Kingma, D.P., Welling, M., 2014. Auto-encoding variational bayes, in: *2nd International Conference on Learning Representations (ICLR 2014)*, Banff, AB, Canada. URL: <http://arxiv.org/abs/1312.6114v10>, arXiv:1312.6114.
- [26] Kuethe, S., de Keizer, C., Shahbazfar, R., Vajen, K., 2011. Implementation of data processing and automated algorithm based fault detection for solar thermal systems, in: *Proceedings of the ISES Solar World Congress 2011*, Kassel, Germany. pp. 1–8. doi:10.18086/swc.2011.28.16. presented at the ISES Solar World Congress 2011.
- [27] Liu, F.T., Ting, K.M., Zhou, Z.H., 2008. Isolation forest, in: *2008 eighth IEEE international conference on data mining, IEEE*. pp. 413–422.
- [28] Liu, Q., Paparrizos, J., 2024. The elephant in the room: Towards a reliable time-series anomaly detection benchmark, in: *Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C. (Eds.), Advances in Neural Information Processing Systems*, Curran Associates, Inc.. pp. 108231–108261. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/c3f3c690b7a99fba16d0efd35cb83b2c-Paper-Datasets_and_Benchmarks_Track.pdf.
- [29] Malhotra, P., Vig, L., Shroff, G., Agarwal, P., et al., 2015. Long short term memory networks for anomaly detection in time series, in: *Proceedings*, p. 94.
- [30] Munir, M., Siddiqui, S.A., Dengel, A., Ahmed, S., 2018. DeepAnT: A deep learning approach for unsupervised anomaly detection in time series. *Ieee Access* 7, 1991–2005.
- [31] Ohnewein, P., Dröscher, A., Schgaguler, K., Feichtner, F., Meißner, E., Luidolt, P., Köstinger, A., Heimrath, R., Jaendl, M., Streicher, W., 2010. IP-Solar: Development of a web-based monitoring and diagnostics tool for solar thermal systems, in: *Proceedings of Eurosun*.
- [32] Paerisch, P., Vanoli, K., 2006. Quality assurance of solar thermal systems with the ISFH input/output-procedure. *Institut für Solarenergieforschung Hameln, Germany*.
- [33] Paerisch, P., Vanoli, K., 2007. Wissenschaftlicher Schlussbericht zu dem Forschungsvorhaben: Wissenschaftlich-technische Untersuchung des ISFH-Input-Output-Verfahrens zur Ertragskontrolle solarthermischer Systeme sowie Entwicklung und Erprobung von Input/Output-Controllern. *Final Project Report*. Institut für Solarenergieforschung Hameln, Germany.
- [34] Paffenroth, R., Kay, K., Servi, L., 2018. Robust PCA for anomaly detection in cyber networks. *arXiv e-prints*, arXiv–1801.
- [35] Park, D., Hoshi, Y., Kemp, C.C., 2018. A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder. *IEEE Robotics and Automation Letters* 3, 1544–1551. URL: <http://ieeexplore.ieee.org/document/8279425/>, doi:10.1109/LRA.2018.2801475.
- [36] Rolinek, M., Zietlow, D., Martius, G., 2019. Variational autoencoders pursue PCA directions (by accident), in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12406–12415. URL: http://openaccess.thecvf.com/content_CVPR_2019/papers/Rolinek_Variational_Autoencoders_Pursue_PCA_Directions_by_Accident_CVPR_2019_paper.pdf.

- [37] Sakurada, M., Yairi, T., 2014. Anomaly detection using autoencoders with nonlinear dimensionality reduction, in: Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis, pp. 4–11.
- [38] Sarfraz, M.S., Chen, M.Y., Layer, L., Peng, K., Koulakis, M., 2024. Position: Quo vadis, unsupervised time series anomaly detection?, in: Forty-first International Conference on Machine Learning. URL: <https://openreview.net/forum?id=W7Vqx1Jvc2>.
- [39] Schmelzer, C., Georgii, M., Kusyy, O., Sauer, C., Orozaliev, J., Vajen, K., 2021. SolarCheck: Entwicklung eines einheitlichen Verfahrens und eines anschaulichen Indikators zur Feststellung der Funktionsfähigkeit thermischer Solaranlagen für Trinkwarmwasserbereitung und kombinierte Heizungsunterstützung. Abschlussbericht. Universität Kassel.
- [40] Schmelzer, C., Georgii, M., Vajen, K., 2015. FeDet – Automatische Fehlerdetektion und Diagnose thermischer Solaranlagen. Abschlussbericht. Universität Kassel. Kassel, Germany.
- [41] Seitzer, M., Tavakoli, A., Antic, D., Martius, G., 2022. On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks, in: International Conference on Learning Representations (ICLR 2022). URL: <https://openreview.net/forum?id=aP0pX1nV1T>.
- [42] Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., Pei, D., 2019. Robust anomaly detection for multivariate time series through stochastic recurrent neural network, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA. pp. 2828–2837. URL: <https://dl.acm.org/doi/10.1145/3292500.3330672>, doi:10.1145/3292500.3330672.
- [43] Tatbul, N., Lee, T.J., Zdonik, S., Alam, M., Gottschlich, J., 2018. Precision and recall for time series, in: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/8f468c873a32bb0619eae2050ba45d1-Paper.pdf.
- [44] Tran, L., Fan, L., Shahabi, C., 2016. Distance-based outlier detection in data streams. Proceedings of the VLDB Endowment 9, 1089–1100.
- [45] Tuli, S., Casale, G., Jennings, N.R., 2022. TranAD: deep transformer networks for anomaly detection in multivariate time series data. Proceedings of the VLDB Endowment 15, 1201–1214.
- [46] Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M., 2022. Timesnet: Temporal 2d-variation modeling for general time series analysis. arXiv preprint arXiv:2210.02186 .
- [47] Wu, R., Keogh, E.J., 2021. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. IEEE Transactions on Knowledge and Data Engineering 35, 2421–2429. URL: <https://ieeexplore.ieee.org/abstract/document/9537291>.
- [48] Xu, J., Wu, H., Wang, J., Long, M., 2022. Anomaly Transformer: Time series anomaly detection with association discrepancy. URL: <http://arxiv.org/abs/2110.02642>. arXiv:2110.02642 [cs].
- [49] Yairi, T., Kato, Y., Hori, K., 2001. Fault detection by mining association rules from house-keeping data, in: proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space, Citeseer. p. 21.

Table A1

Hyperparameters used in our model.

Hyperparameter	Value
Latent Space Regulation β	0.01
BNLL Parameter β_L	0.5
Learning Rate	1e-4
LSTM Layers	4
Hidden Dimension	64
Latent Dimension	8
Dropout	0.1
Token Length	30
Update Steps	40 000

A. Hyperparameters

Table A1 summarizes the hyperparameter settings used for our model. These include both architectural choices (e.g., network depth, hidden dimensions) and optimization parameters (e.g., learning rate, dropout). Although the main text provides an overview of the most critical parameters, we include this comprehensive table here to facilitate reproducibility.

B. Train, Validation, and Test Splits

Our dataset is divided into three disjoint sets: training, validation, and testing. Each set comprises entire systems, ensuring that no system appears in more than one set. This setup reflects our goal of testing fault detection on systems unseen during training.

Training Systems

[001, 013, 023, 025, 033, 035, 041, 043, 049, 053, 058, 060, 062, 069, 075, 077, 105, 106, 110, 116, 126, 138, 146, 150, 173, 175, 193, 902, 904, 905]

Validation Systems

[026, 038, 046, 063, 074, 076, 114, 118, 136]

Test Systems

[002, 003, 005, 007, 012, 014, 015, 016, 027, 028, 029, 030, 031, 034, 036, 037, 040, 044, 045, 047, 048, 050, 051, 052, 054, 059, 061, 064, 065, 067, 068, 070, 072, 107, 119, 144, 149, 154, 158, 166, 169, 179, 181, 903]

This partition ensures that each subset contains entire systems, preventing data leakage across training, validation, and test phases. By testing on systems that were never seen during training, we simulate a real-world scenario in which the model encounters previously unknown systems.

C. PaSTS Dataset

In this work, we use the PaSTS dataset [10]. The dataset is taken from the manufacturer's service team and consists of systems in a completely nominal state, systems degrading from a nominal state towards a fault state, and systems only in fault states. The dataset consists of 83 different STS with varying installations and operational durations between 1 and 3329 days of operation, for a total of 39878 days. We use the preprocessed dataset, which includes only full days, with a time resolution of 1 datapoint per minute, for a total of 1440 datapoints per day. For every timestep, there are a total of 17 fields. These fields are index data (the timestamp and the system identifier), sensor data (temperature, yield and volumetric flow data), control signals to the pump, and status indicators. Which of the variables we use for our model is discussed in Section 4.3.

The status indicators include the output of an internal expert system, which is a rules- and physics-based system, that classifies the current system state into nominal, anomalous (*Merk*) or a fault (*sto*). The controller propagates anomalous and fault states over time and automatically runs tests that confirm or reject these states. Additionally, for some classified states, the system will shut itself down, leading to passive, nominal behaviour.

D. Example Reconstructions

Here we give some example reconstructions of a system in a nominal state and a system in a faulty state. It is noted that the reconstruction quality can be tuned to be arbitrarily good. The reconstruction quality is only interesting for us in the context of anomaly detection, and reconstructions should only be directly used to analyse the behaviour of the model, looking for unexpected behaviour.

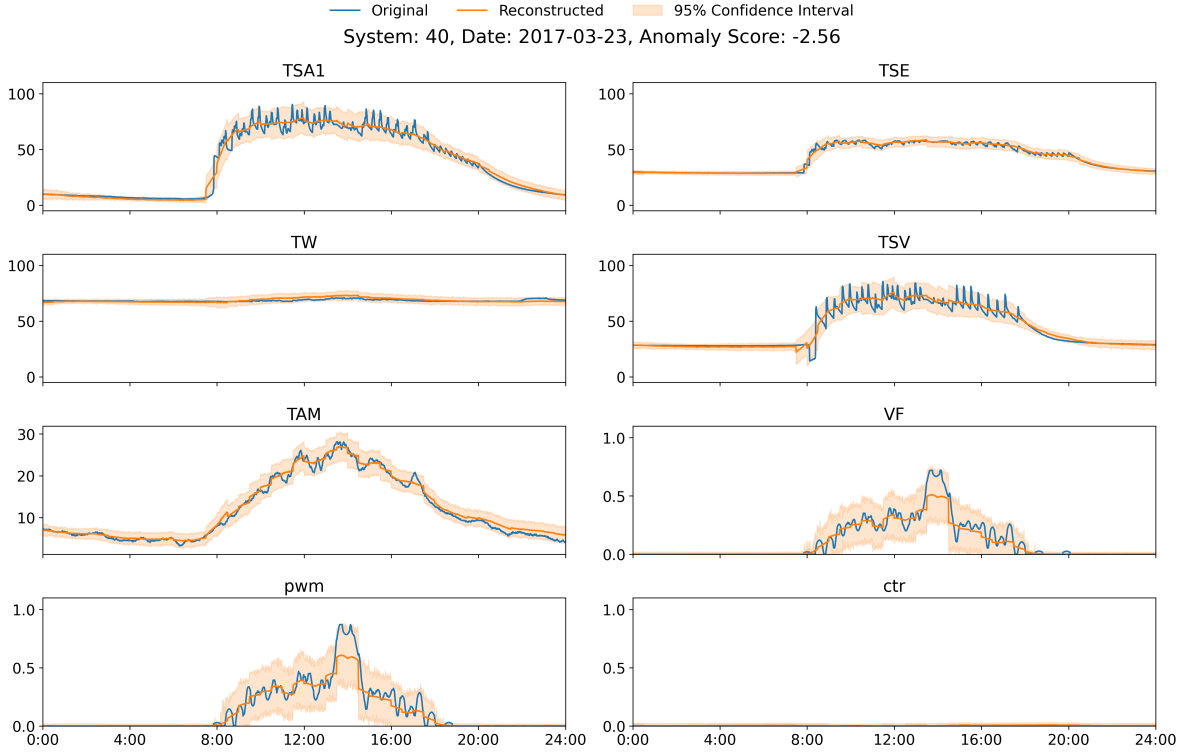


Figure A1: Example of the reconstruction of a normal spring day. Here the system is working exactly as intended, which results in good reconstructions and low anomaly scores.

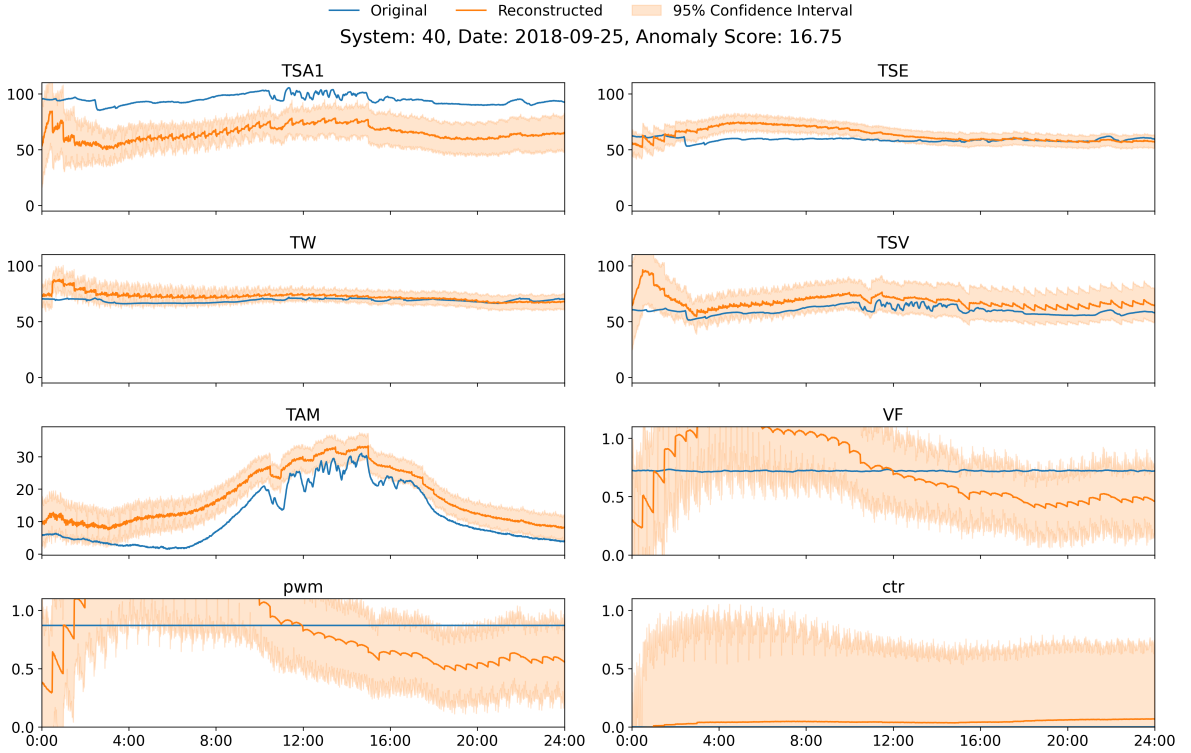


Figure A2: Due to an electrical issue, the system entered a state in which the TSA1 sensor is consistently reading the wrong temperature. This leads to a locked state, in which the pump remains permanently running, resulting in increased heating demand and more severe faults down the line. As this is completely out of distribution, the reconstructions are off, and we observe high anomaly scores.

E. Comparison of PCA and our Model

Comparison of the best performing baseline— Rescaled PCA-R model— with Our model.

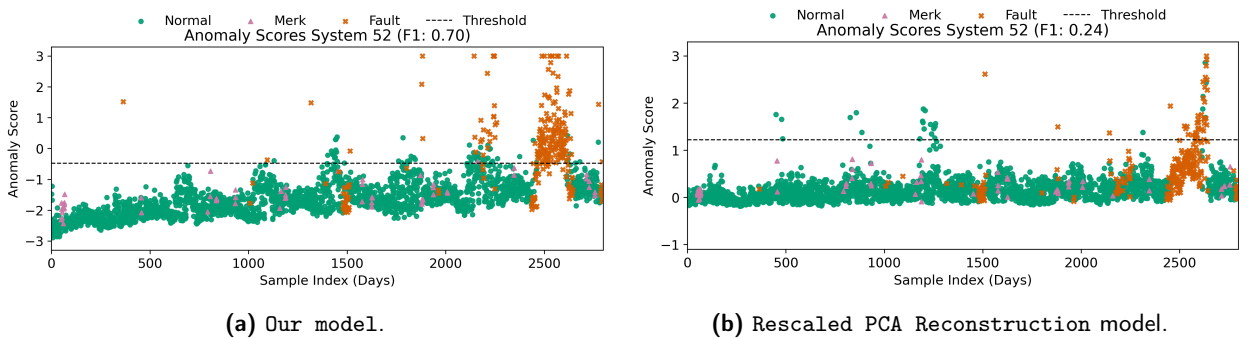
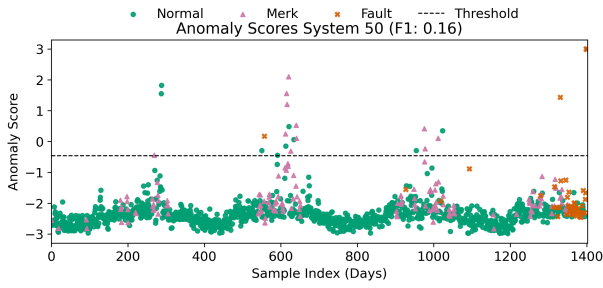


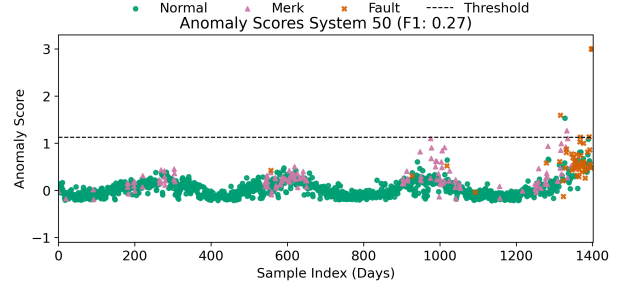
Figure A3: Our model (left) shows a trend in the anomaly score, indicating a consistent degradation of the system, leading to the complete failure of the system. The same trend is not observable in the PCA model (right), even though detection performance is comparable.

In Figure A3 and Figure A4, we see the qualitative early detection capabilities of our model. As discussed in the main text, in Figure A3, we can see a trend in the anomaly score time series, indicating a degradation of the system. This trend is already present before the critical issues arise later in the lifetime of the system, potentially allowing for early interventions. Compared to that, the PCA model had a somewhat stable evolution of the anomaly score. In Figure A4

Fault Detection in Solar Thermal Systems



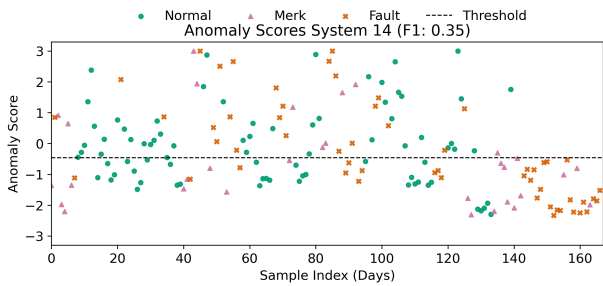
(a) Our model.



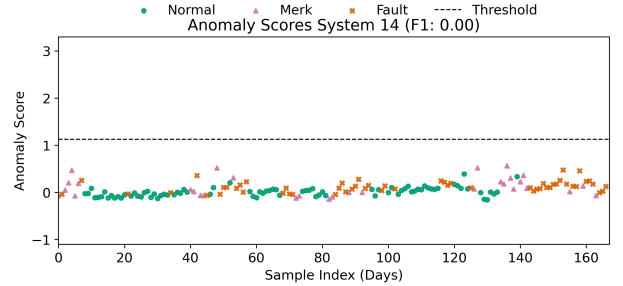
(b) Rescaled PCA Reconstruction model.

Figure A4: While detection performance is not good for either model, our model (left) does detect faults in the years prior to the system breakdown, while the PCA model (right) does not detect the problematic system early.

we additionally see that the seasonality of the anomaly score leads to potential faults that again degrade the system further, leading to faults later on. The PCA model does not detect any anomalies here, missing the faulty installation.

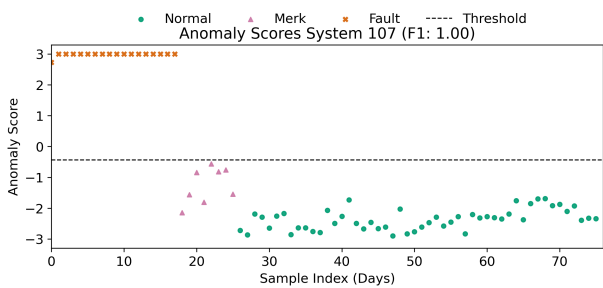


(a) Our model.

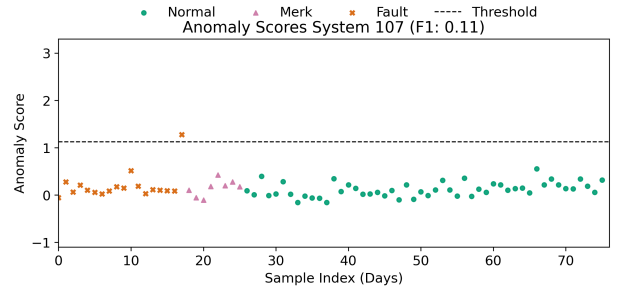


(b) Rescaled PCA Reconstruction model.

Figure A5: Qualitative analysis of the anomaly scores from our model (left) correctly identifies the system as problematic. The PCA model does not identify any faults here.



(a) Our model.



(b) Rescaled PCA Reconstruction model.

Figure A6: Our model (left) correctly identifies the system as faulty, while the PCA model (right) only detects the last day of the fault indication.

In Figure A6, Figure A7, and Figure A5, we can see that in all these cases the PCA model fails to detect the faults present in the system, while our model correctly identifies the systems as faulty.

Fault Detection in Solar Thermal Systems

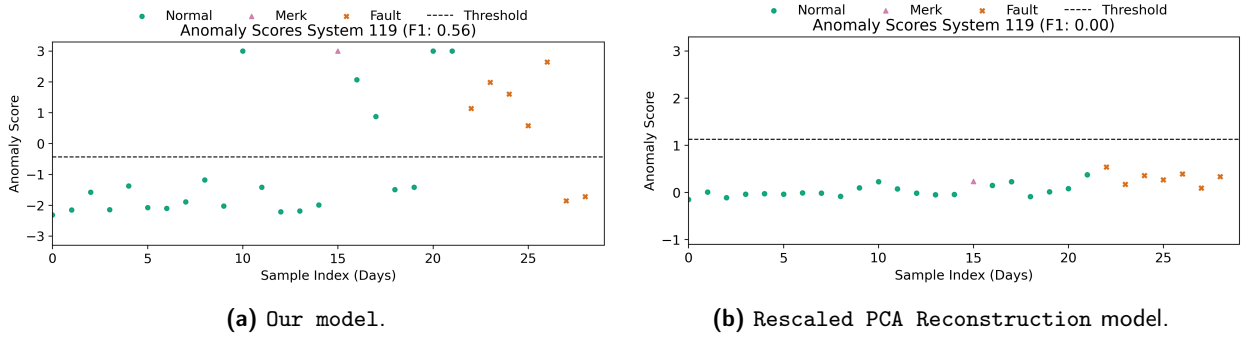


Figure A7: Our model (left) correctly identifies the system as faulty, with the first fault indications occurring before the system's internal fault indication flags the system as faulty. The PCA model (right) does not detect any system problem.

F. Metrics Discussion

As discussed in Section 4.3, we evaluate our results using the optimal F1 score, the system-wise F1 score, the AUC-ROC and the AUC-PR. The optimal F1 score can be seen as an upper bound on the F1 score using a static threshold over the entire dataset. However, here the threshold is chosen as the optimum over the testset. To avoid this, one can use K-fold cross-validation. Here, the dataset is randomly separated into K different sets of data. When doing this, we can estimate the threshold on the K-1 sets and apply it to the single set. Afterwards we average the resulting F1 score over all different configurations. In Figure A8 we see the resulting F1 score for different K. As is clear in this plot, both the F1 score as well as the threshold does not substantially change using this method. This is why we opted to use the optimal F1 score metric, which is commonly used in time series anomaly detection.

Additionally, due to the strong imbalance in system length, the 3 standard metrics — optimal F1, AUC-ROC and AUC-PR— underestimate performance in small systems. Thus, they do not reflect generalisation capabilities of the models. However, the system-wise F1 score weights every system with the same weight, better representing generalisation capabilities.

G. Original Annotation Results

In Table A2 we show the results for the original dataset annotation. Here, we observe the same general behaviour as in the reannotated dataset. In general, we want to emphasise that the reannotation is done together with industry professionals and is considered more reliable than the original notation. However, we want to point out that the forecasting-based models— forecasting a single step ahead— are performing better on this specific annotation.

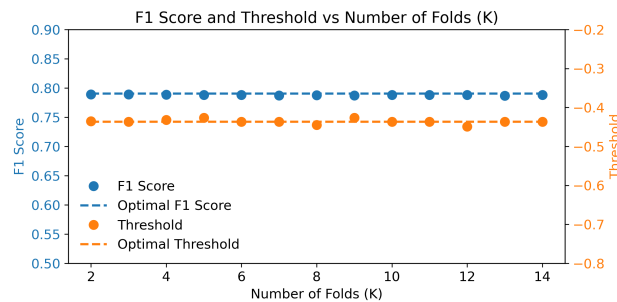


Figure A8: K-fold cross-validation for the F1 score. On the x-Axis the amount of folds is shown. The y-axis shows the average F1 score and the threshold. The dashed lines show the optimal threshold and optimal F1 score.

Table A2

Comparison of anomaly detection performance on the original dataset annotation. Values are the mean over all seeds \pm the standard deviation. For non-stochastic methods– like PCA– no standard deviation is given. Baselines are as discussed in Section 4.2. Baselines marked with † are forecasting-based deep learning models and forecast a single step. Best performing models and models without a statistically significant difference to the best performing model are marked in bold.

Method	System-wise F1	Optimal F1	AUC-PR	AUC-ROC
Unscaled PCA-R	0.24	0.37	0.37	0.76
Rescaled PCA-R	0.28	0.39	0.30	0.79
LSTM-VAE	0.24 \pm 0.03	0.37 \pm 0.01	0.36 \pm 0.02	0.79 \pm < 0.01
Our Model	0.29 \pm 0.02	0.44 \pm 0.02	0.34 \pm 0.04	0.79 \pm 0.01
HBOS [16]	0.09	0.20	0.11	0.59
LOF [4]	0.16	0.34	0.29	0.75
KMeansAD [49]	0.18 \pm 0.01	0.33 \pm < 0.01	0.30 \pm 0.01	0.76 \pm < 0.01
PCA [34]	0.25	0.35	0.18	0.73
IForest [27]	0.07	0.20	0.15	0.60
AnomalyTransformer [48]	0.18 \pm 0.03	0.26 \pm 0.03	0.21 \pm 0.04	0.66 \pm 0.02
AutoEncoder [37]	0.16 \pm < 0.01	0.20 \pm < 0.01	0.16 \pm < 0.01	0.59 \pm < 0.01
CNN [30] †	0.25 \pm 0.01	0.44 \pm 0.01	0.39 \pm < 0.01	0.79 \pm < 0.01
Donut [44]	0.17 \pm 0.02	0.36 \pm 0.02	0.31 \pm 0.01	0.71 \pm 0.01
LSTMAD [29] †	0.23 \pm 0.02	0.41 \pm < 0.01	0.40 \pm 0.01	0.78 \pm < 0.01
TimesNet [46]	0.18 \pm 0.02	0.36 \pm 0.01	0.35 \pm 0.01	0.70 \pm 0.01
TranAd [45]	0.17 \pm 0.02	0.38 \pm 0.02	0.38 \pm 0.02	0.76 \pm 0.01
USAD [2]	0.18 \pm 0.01	0.23 \pm < 0.01	0.22 \pm < 0.01	0.64 \pm 0.01