One-Class Predictive Autoencoder Towards Unsupervised Anomaly Detection on Industrial Time Series

Hongjing Zhang*
AWS AI Labs
zhhongji@amazon.com

Fangzhou Cheng
AWS AI Labs
fzc@amazon.com

Aparna Pandey
AWS AI Labs
apapan@amazon.com

ABSTRACT

Anomaly detection is a fundamental problem of data science that aims at finding instances of unusual data. In recent years, due to the rapid expansion of the Industrial Internet of Things (IIoT), substantial amounts of high-dimensional industrial time series data have been generated. Detecting potential anomalies from such data is challenging and an important research topic. In this paper, we propose *One-Class Predictive Autoencoder* (OCPAE), a novel encoder-decoder approach with additional prediction and one-class branches to enhance the performance on detection of time series anomalies from different perspectives. The prediction branch can detect anomalies by learning the local temporal dependency while the one-class branch is suitable to learn the normal patterns from a global perspective. We evaluate our proposed approach on five public datasets and demonstrate the superiority of our approach over other state-of-the-arts methods. Lastly, we conduct ablation studies and in-depth analysis to show the effectiveness, efficiency, and robustness of our proposed method.

CCS CONCEPTS

• **Computing methodologies** → **Anomaly detection**.

KEYWORDS

Anomaly detection, Unsupervised learning; Multivariate time series, Autoencoders, One class classification, Deep learning

ACM Reference Format:
Hongjing Zhang, Fangzhou Cheng, and Aparna Pandey. 2022. One-Class Predictive Autoencoder Towards Unsupervised Anomaly Detection on Industrial Time Series. In *ANDEA '22: 2nd SIGKDD Workshop on Anomaly and Novelty Detection, Explanation and Accommodation, August 14th, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 10 pages.

1 INTRODUCTION

Anomaly detection for Industrial Internet of Things (IIoT) systems is an area of growing interest in both academia [17] and industry [5]. A single hour of unplanned downtime in critical infrastructure can lead to >US $1M of operational losses [37]. The number of IIoT-enabled equipment is increasing rapidly due to the development of low-cost sensors, and as a result, there is an explosion of high-dimensional industrial sensor data. This leads to an emerging opportunity for developing advanced anomaly detection strategies to increase system availability and reduce unplanned downtime [39], especially on critical infrastructure and systems, such as water treatment and distribution networks [2, 29], spacecrafts [20], robot-assisted systems [10], and wind turbines [11].

Rule-based methods are commonly used for industrial anomaly detection tasks [8]. Domain experts carefully design rules for a specific system to identify normal and abnormal behaviors. The rules can be a simple threshold on a single sensor value, or a set of complex conditions on multiple sensors. Such rule-based methods suffer from accuracy and scalability issues. With a higher number of sensors, it becomes harder for experts to find relationships between these sensors, which makes it difficult to design the rules. In addition, rule-based methods need to be carefully reviewed whenever we introduce a new system, a new operating condition or even add a new sensor to the existing systems, which makes the rule-based methods not scalable. Thus, it is critical to develop an effective method for anomaly detection of multivariate time series sensor data. Despite the need, it is highly challenging to develop a scalable anomaly detection method for different industrial systems due to the lack of labeled anomalies, poor data quality, and few publicly available datasets.

Multivariate time series anomaly detection problem is typically treated as unsupervised learning problem due to a dearth of labeled data. In the early 2000's, many classical unsupervised approaches were used to tackle this problem, such as density-based methods [1], regression-based methods [36], and one-class methods based on support vector machines [26]. However, these approaches mainly model relationships between sensors in fairly simple ways, and as the number of sensors keeps increasing, it becomes harder for them to learn the model effectively [5]. More importantly, several of the cited work have not accounted for temporal dependencies, which are crucial for achieving better anomaly detection accuracy of time series. Recently, unsupervised anomaly detection methods based on deep learning have received much attention due to their ability to learn both feature and temporal dependencies. They can be mainly categorized into three categories: prediction-based methods, reconstruction-based methods, and Generative adversarial networks (GAN)-based methods. Prediction-based methods [20, 41] predict the data for next few steps using historical data, and compare with the true values. In this way, the models try to learn the normal behavior of the time evolution of data, and then anomalies can be detected based on the prediction error. One issue with this type of methods is they mainly focus on learning short-term temporal dependencies, while they lack the ability to learn the common global patterns that can be used for global anomaly detection. Reconstruction-based methods [30, 42] learn low-dimension

*Work done during an Amazon internship.

ANDEA '22, August 14th, 2022, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.

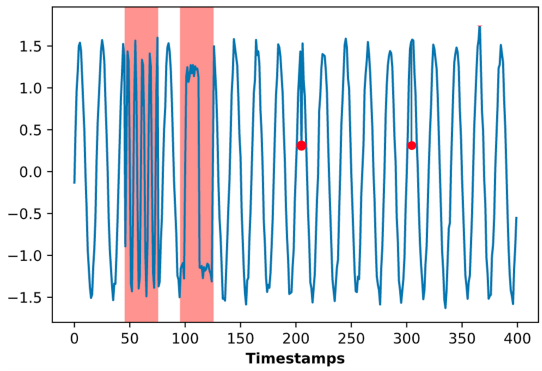


Figure 1: Illustrative example with global pattern anomalies (rectangle areas) and local pattern anomalies (dotted areas). Our proposed approach aims to enhance the autoencoder framework from both local and global perspective.

representations of normal data. The anomalies are detected based on the reconstruction error, which is the difference between the true value and the reconstructed value. There are mainly two issues with this type of methods. One is that they are weak in learning local temporal dependencies because the model is trained with window-level input and the reconstruction error is calculated from a sequence of samples and, therefore, they are not sensitive to small perturbations at specific timestamps with short duration. The other drawback is they sometimes can also reliably reconstruct anomalies due to overfitting, and thus, regularization or extra constraints are needed to mitigate this problem [16]. GAN-based methods [14, 23] are trained to learn the distributions of normal samples. During inference, both the prediction error from the discriminator and reconstruction error from the generator can be used to detect the anomalies. One limitation with GAN-based methods is training challenges due to mode collapse and non-convergence [21].

Figure 1 shows an illustrative example with global and local pattern anomalies, which are hard for reconstruction-based approaches to detect. To address the limitations of reconstruction-based method mentioned above and enable it to detect anomalies with global and local patterns such as those in Figure 1, we propose an innovative multi-branch encoder-decoder framework called one-class predictive autoencoder (OCPAE) for unsupervised anomaly detection on industrial time series data. It consists of a reconstruction branch, a prediction branch and a one-class branch. The prediction branch aims at learning the local temporal dependency with historical data, so that our model can be aware of the anomalies that only have small perturbations at some specific timestamps with short duration. The one-class branch focuses on learning the normal global patterns via mapping all input time frames to an "anomaly aware" feature space. By doing this, even if the anomalies can be reconstructed with reconstruction branch, our framework can still be aware of the anomalies if their mapping is away from the global center. Now we summarize the contributions of this paper as follows:

- We propose a novel encoding-decoding framework that contains a reconstruction branch, a one-class branch, and a prediction branch. Based on our ablation study, we demonstrate the success of our design choices for each branch.
- We propose a one-class branch that can learn the normal global patterns from the time series data by incorporating the one-class classification objectives to the learned latent representation.
- We introduce a prediction branch to our framework that can detect anomalies from local perspective by learning the short-term temporal dependencies.
- We conduct extensive experiments on five publicly available industrial time series datasets to demonstrate the superiority and effectiveness of our proposed framework. For the studied industrial datasets, our method is robust towards different parameters and is efficient in model training and inference.

2 RELATED WORK

Traditional Non-Deep Learning Methods: Over the past decades, many traditional unsupervised methods have been proposed for anomaly detection task. K-Nearest Neighbor (kNN) [18] and Local Outlier Factor (LOF) [7] are two distance-based methods that aim to find anomalies via measuring the local deviation of a given data with respect to its neighbours. Traditional statistical models such as autoregressive integrated moving average (ARIMA) have been used to capture temporal predictions [41]. One-class Support Vector Machine (SVM) [28] and Isolation forest [25] identify decision boundary between normal data and anomalies by modeling the normal data distribution. Inspired by many of these classic algorithms, more recently several deep learning based anomaly detection formulations have been introduced. We describe these approaches in following paragraphs.

Prediction-based Methods: A prediction-based method usually learns a predictive model to fit the given time series data and then uses that model to predict future values. A data point is identified as an anomaly if the difference between its predicted value and the true value exceeds a certain threshold. With the advancement of deep representation learning, [20] proposes to use long-short term memory (LSTM) recurrent neural networks to automatically extract features and predict future samples. [20] is demonstrated to achieve good performance on industrial time-series anomaly detection tasks. Besides the developments of prediction-based methods, one common limitation among these methods is the lack of global detection ability: the prediction-based methods tend to fail when one sample satisfies the local temporal dependencies but the overall pattern is rare or abnormal compared to other samples in the data. Our proposed OCPAE will take the benefits of prediction-based methods at detecting the changes at local perspective by adding a prediction branch to our framework.

Reconstruction-based Methods: Reconstruction-based methods assume anomalies lose information when mapped to a compressed space and cannot be reconstructed well. A sample with high reconstruction error suggests a high chance of being anomalous. Various reconstruction-based approaches [4, 15, 27, 38, 40] have been proposed to address the time series anomaly detection problems. To improve the robustness and performance of autoencoders, some work has been done to connect autoencoders with other learning algorithms. For example, the deep autoencoding Gaussian mixture model (DAGMM) [42] jointly considers an autoencoder and a Gaussian mixture model to learn the normal patterns in

the latent representation space. USAD [5] incorporates adversarial training with autoencoders to learn to reconstruct the most normal instances within the training set. Our work also incorporates the idea of reconstruction-based methods, but we address the limitations of autoencoders from both local and global perspectives via a prediction branch and a one-class branch.

GAN-based Methods: GAN-based methods are trained to learn the distribution of normal instances using adversarial learning. Generators can be directly used for time series reconstructions and discriminators can be used for anomaly prediction. [23] proposes to use a vanilla GAN to learn the distribution of multivariate time series and make predictions based on the combination of learned discriminator and generator. [14] introduces the cycle-consistent GAN architectures for time series so that the generators can be used for time series reconstruction. It also explores different ways of combining the discriminator's predictions and the reconstruction error to find the best possible combination to calculate the anomaly score. [6] explores using GAN to detect time series anomalies in small datasets, whose architecture contains a generator to learn the normal data distribution and an inverse mapping to map input to latent space. Although GAN-based approaches are good at learning the representation of normal time series, GAN-based models can be unstable and hard to train due to adversarial training [21].

Other Novel Methods: Recently some new types of time series anomaly detection methods have been proposed which achieved good performance. For example, [35] extends the one-class classification objective to consider multiple spheres in the latent representation space for learning the normality of the training data. Graph Neural Networks (GNNs) have emerged as successful approaches to model complex patterns in graph-structured data. [9, 13] propose to use GNNs to learn the relationships between different time series signals and use the learned structure and latent representations to do prediction tasks. The difference between the predicted values and the actual data can be adopted for calculating anomaly scores.

3 METHODS

In this section, we introduce the overall framework of our proposed OCPAE and the design of each branch. The overall framework of OCPAE is visualized in Figure 2. It consists of a basic encoder-decoder reconstruction branch and two extra branches focusing on learning the normality of global patterns and local patterns using the input sequences respectively. We will introduce the details of each part in the following subsections.

3.1 Unsupervised Anomaly Detection via Autoencoder

Autoencoder-based anomaly detection methods are widely used in recent work [4, 15, 27] to tackle the unsupervised time series anomaly detection problems. Let the encoder network be denoted as ϕ_e and decoder network as ϕ_d. Denote the input data at timestamp i as x_i, the input window $\mathbf{w_i}$ which consists of l_s timestamps can be represented as $\{\mathbf{x_i}, \mathbf{x_{i+1}}, ..., \mathbf{x_{i+l_s-1}}\}$, the encoder network maps the input window into the latent space $\phi_e(\mathbf{w_i})$.Then the decoder network ϕ_d will map the latent data $\phi_e(\mathbf{w_i})$ back to the input space as $\phi_d(\phi_e(\mathbf{w_i}))$. Let the total number of input time windows be N and the parameters for the networks be θ_e, θ_d, the training objective of the autoencoder aims to minimize the reconstruction error as the difference between the input window and the reconstructed window:

$$\ell_{AE} = \frac{1}{N}\sum_{i=1}^{N} ||\phi_d(\phi_e(\mathbf{w_i};\theta_e);\theta_d) - \mathbf{w_i}||^2 \tag{1}$$

Once the autoencoder is trained on the normal time series data with Eq. 1, instances that cannot be compressed and reconstructed well are considered to be anomalies. To make out-of-sample predictions for unseen data, the autoencoder will use the reconstruction error for unseen test window $\hat{\mathbf{w}}_\mathbf{i}$ as the anomaly score S_{AE}:

$$S_{AE}(\hat{\mathbf{w}}_\mathbf{i}) = ||\phi_d(\phi_e(\hat{\mathbf{w}}_\mathbf{i};\theta_e);\theta_d) - \hat{\mathbf{w}}_\mathbf{i}||^2 \tag{2}$$

Although autoencoder is a popular and effective approach, it also has two limitations that our proposed one-class predictive autoencoder will address. Firstly, since the reconstruction error is calculated for a sequence of samples and the autoencoder model is trained with window-level input, some small perturbations at specific timestamps may be overlooked by the model, and therefore, the anomaly can not be detected because autoencoder does not have the ability to learn the normal local patterns. Moreover, we empirically observe from our real-world experimental analysis that autoencoders can not only reconstruct the normal windows but also the anomaly windows. This case will happen more often when the training data contains anomaly data. To overcome this limitation, the autoencoder should be "anomaly aware" of some anomaly windows even if the autoencoder tends to reconstruct them well.

3.2 Detecting Anomalies from Local Perspective

In the previous subsection, we introduced the autoencoder for anomaly detection and discussed the two potential limitations of using purely reconstruction errors for anomaly detection. One of the limitations is that the autoencoder may not be sensitive enough for the small perturbations at specific timestamps (Recall the dotted areas in Figure 1). To enhance the autoencoder's ability to detect time series anomalies from a local perspective, we design an extra prediction branch to predict the one-step-ahead value at future timestamp using previous values. By adding the prediction branch, our proposed framework is able to learn temporal dependencies in between adjacent samples.

The task of the prediction branch is to learn the local temporal dependencies by minimizing prediction error between predicted future value $\mathbf{x_{i+l_s}}$ and its actual value. In the training phase, we connect the encoded latent space to a neural network g which acts as the predictor to capture the local temporal information and detect the potential local changes. To be concrete, given all the N training windows and the predictor network g with parameter θ_g, we design the following objective function to guide the local pattern learning procedure:

$$\ell_{Pred} = \frac{1}{N-1}\sum_{i=1}^{N-1} ||g(\phi_e(\mathbf{w_i};\theta_e);\theta_g) - \mathbf{x}_{i+l_s}||^2 \tag{3}$$

Note that although the last window is $\mathbf{w_N}$, our training objective's last input window will be $\mathbf{w_{N-1}}$ as the future timestamp to predict is $\mathbf{x_{N+l_s}}$. To leverage the temporal information for predictor

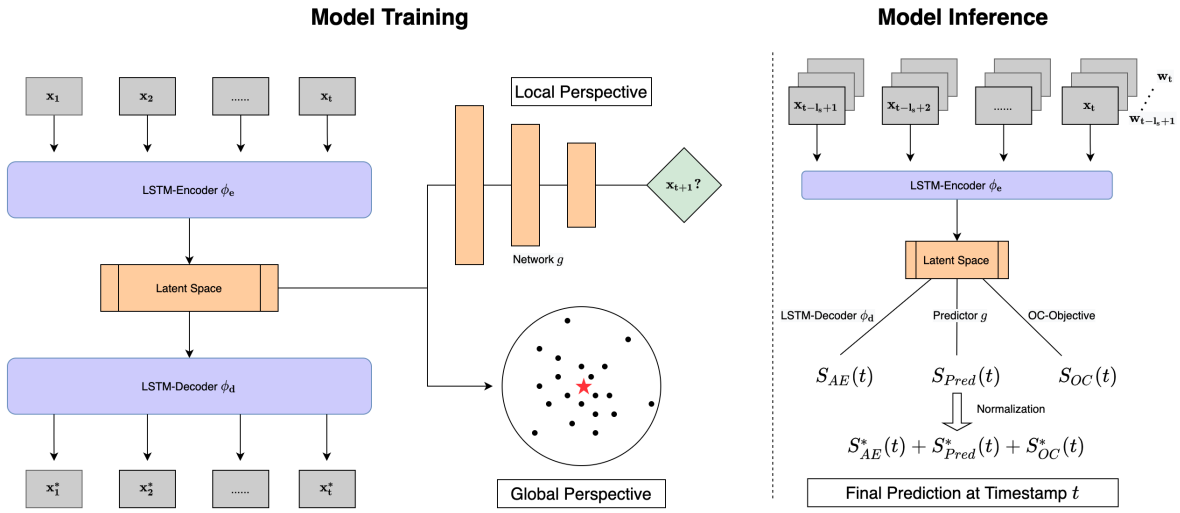


Figure 2: The overview of the proposed one-class predictive auto-encoder. It consists of an encoder-decoder branch and two extra components focusing on learning the normality of input time frames and local pattern respectively.

g, we implement our encoder ϕ_e and decoder ϕ_d with LSTM networks. For the inference stage, we calculate the local-perspective anomaly score S_{Pred} for sample at timestamp t as:

$$S_{Pred}(t) = ||g(\phi_e(\mathbf{w_{t-l_s}}; \theta_e); \theta_g) - \mathbf{x}_t||^2 \tag{4}$$

3.3 Learning the Normal Global Patterns

To address the limitation that the trained autoencoder can generalize and reconstruct the anomaly windows well especially when the input data is not anomaly free, we propose to add a one-class branch that can reshape the latent space to be "anomaly aware" so that the potential rare latent embeddings can be detected. Therefore, even if the rare input windows can be reconstructed they can still be detected based on our "anomaly aware" latent space.

To achieve this goal, the one-class classification objective is incorporated to the learned latent representation. Here we briefly summarize a deep version of one-class classifier [33] that we will adopt in our work. Unlike prediction-based and reconstruction-based anomaly detection models, one-class classification models directly optimize for a minimum hypersphere that can describe as many normal instances as possible. Deep Support Vector Data Description (Deep SVDD) [33] improves upon the traditional kernel-based one-class classification models with neural networks for better representation learning. Furthermore, it minimizes a quadratic loss for penalizing the distance of each instance's latent representation to a centroid.

Given the training data, the one-class classification objective is minimized to map the embeddings of all N input windows close to a fixed center $\mathbf{c}$, where $\mathbf{c}$ is normally set as the mean of all the latent embeddings. We formally design the objective function as:

$$\ell_{OC} = \frac{1}{N}\sum_{i=1}^{N} ||\phi_e(\mathbf{w_i}; \theta_e) - \mathbf{c}||^2 + \frac{\lambda}{2}\sum_{\ell=1}^{L} ||\theta^\ell||^2 \tag{5}$$

Note the first term of the proposed objective optimizes a simplified one-class classification problem and the second term is a network weight decay regularizer with hyper-parameter $\lambda > 0$ which prevents finding a too complex mapping function. We denote the whole network has L hidden layers and apply weight decay for all the layers with parameters $\{\theta^1, ...\theta^L\}$. Deep SVDD contracts the embedding space enclosing the points by minimizing the mean distance of all data points to the center. The embeddings closer to the center represent the global normal patterns. During the evaluation/scoring stage, given a test window $\mathbf{w_i}$, the anomaly score S_{OC} for $\mathbf{w_i}$ is calculated as follows:

$$S_{OC}(\mathbf{w_i}) = ||\phi_e(\mathbf{w_i}; \theta_e) - \mathbf{c}||^2 \tag{6}$$

Note this anomaly score is the distance between embedding and the center, and therefore, anomalies are far from the center and can be detected.

3.4 Training the Overall Framework

We have introduced the overall architecture of our proposed OCPAE and design of each branch. Here we formulate our overall learning objective to include all three branches.

$$\ell = \ell_{AE} + \alpha\ell_{Pred} + \beta\ell_{OC} \tag{7}$$

Hyperparameter $\alpha \geq 0$ controls the importance of prediction branch objective which is optimized for learning temporal dependencies from local perspective, and $\beta \geq 0$ controls the degree of one-class branch objective from global perspective. We set $\alpha = 1$ and $\beta = 1$ for OCPAE in our experiments. Our proposed framework has two variants which are predictive autoencoder (Pred-AE) ($\beta = 0$) and one-class autoencoder (OC-AE) ($\alpha = 0$).

To learn with the proposed overall objective in Eq. 7, we directly optimize the overall loss via stochastic gradient descent (Adam optimizer) with dynamic learning rate.

3.5 Model Inference

During the model inference stage, the anomaly score is calculated based on all three branches in OCPAE. For the reconstruction branch, to compute reconstruction error for the sample at timestamp t, we first find all the windows that contain timestamp t as $\{\mathbf{w_{t-l_s+1}}, ..., \mathbf{w_t}\}$. Then we calculate the average reconstructed error as the anomaly score $S_{AE}(t)$ for sample at timestamp t :

$$S_{AE}(t) = \frac{1}{l_s} \sum_{i=t-l_s+1}^{t} ||\phi_d(\phi_e(\mathbf{w_i}; \theta_e); \theta_d) - \mathbf{w_i}||^2 \tag{8}$$

The local-perspective anomaly score is the square of the error between actual and predicted values at timestamp t in the prediction branch. We denote it as $S_{Pred}(t)$ Eq. 4.

In one-class branch, we find all the windows that contain timestamp t, and then use the average of squared l_2 distance between the center $\mathbf{c}$ and the embeddings as global-perspective anomaly score:

$$S_{OC}(t) = \frac{1}{l_s} \sum_{i=t-l_s+1}^{t} ||\phi_e(\mathbf{w_i}; \theta_e) - \mathbf{c}||^2 \tag{9}$$

Since the anomaly scores from different branches have different scales, we apply z-score normalization to all three branches to combine them. Let the normalized anomaly scores for the sample at timestamp t for all three branches as $S^*_{AE}(t), S^*_{Pred}(t)$ and $S^*_{OC}(t)$. We compute the overall anomaly score as the summation among all three normalized scores:

$$S_{Overall}(t) = S^*_{AE}(t) + S^*_{Pred}(t) + S^*_{OC}(t) \tag{10}$$

4 EXPERIMENTS

In this section, we answer the following three research questions by conducting experiments[1] on five industrial datasets and comparing our proposed OCPAE with state-of-the-art methods.

- RQ1: How does OCPAE perform on industrial time-series data, in comparison with the state-of-the-art methods?
- RQ2: How effective is the design choice of each branch in OCPAE? How does our proposed model work on different types of anomalies?
- RQ3: How efficient is OCPAE in training and inference and how sensitive is it to parameter choices?

4.1 Datasets

Five real-world anomaly detection datasets are used in the experiments to evaluate our proposed OCPAE. Soil Moisture Active Passive satellite (SMAP) and Mars Science Laboratory rover (MSL) are two datasets published by NASA [20], which contain telemetry anomaly data that indicating unexpected events during spacecraft post-launch operations. The Secure Water Treatment (SWaT) dataset [29] is obtained from multiple sensors in a water treatment system that contains both normal operations and cyber attack scenarios, which are treated as the anomalies in the dataset. Water Distribution (WADI) dataset [2] is collected from a water distribution system comprising many water pipelines. Similar to SWaT, cyber attack scenarios are applied to the system as the anomalies and can last from a few minutes to several hours. Another dataset we used is recently released in the Prognostics and Health Management Society data challenge 2021 (PHM21) [31]. PHM21 dataset represents the fuse quality control pipeline that can be easily integrated in multiple different industry manufacturing lines. 29 out of 99 subsets contain different kinds of anomalous behavior in one or more components in the pipeline. The statistics of the five datasets are summarized in Table 1.

[1]Supplementary material for reproducibility and extra results can be accessed through this link.

Table 1: Dataset Description

Dataset	Subsets	Features	Train	Test	Anomalies
SMAP	55	25	2556	8071	12.8%
MSL	26	55	2160	2731	10.5%
SWaT	1	51	475200	449919	12.1%
WADI	1	118	789371	172801	5.9%
PHM21	99	247	367920	579850	29.3%

4.2 Evaluation Metric

We adopt the standard Precision, Recall and F1-Score (F1) as the evaluation metrics to evaluate and compare the performance of OCPAE and other baseline methods, where:

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN} \tag{11}$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{12}$$

TP, FP, and FN represent the truly detected anomalies, falsely detected anomalies, and the misclassified normal samples, respectively. For industrial applications, anomalous events are rare and are usually contiguous anomaly segments that contain consecutive anomalous points. Thus, it is more important for users to receive timely true alarms without too many FPs. Two major adjusted metrics are used in literature to evaluate the performance on the aforementioned five datasets. For a comprehensive comparison, we use both of the metrics in our experiments. The first metric called point-adjusted F1 is proposed in [38]. In this metric, if at least one sample in an anomalous segment is correctly detected, all the samples in that segment are also considered as TPs, regardless they are TPs or FNs. The ground-truth (GT) negative samples are treated as usual. When using this metric, we search over all possible thresholds for the best F1 during test, and denoted as $F1^*$. The second metric proposed in [20] calculates segment-wise F1 using the following rule: (1) A TP is recorded if a predicted event overlaps any GT anomalous events; (2) a FP is recorded if a predicted event does not overlap any GT anomalous events; (3) a FN is recorded if there is no sample flagged as anomaly within a GT anomalous event. This metric is used in many papers that conduct experiments on SMAP and MSL. To identify anomaly samples, $mean + 3 \cdot std$ of all anomaly scores of samples in the training set is calculated and used as the threshold. The threshold is then applied on the anomaly scores of samples in the test set to calculate F1, and we denoted this segment-wise F1 as $F1^{**}$. We also provide two illustrative examples of $F1^*$ and $F1^{**}$ in the supplementary material.

Table 2: Segment-wise $F1^{**}$ of OCPAE and baselines.

Methods	SMAP	MSL	avg.rank
ARIMA	0.420	0.492	5.5
LSTM	0.690	0.460	5.0
LSTM-AE	0.751	0.586	2.5
MAD-GAN	0.128	0.111	7.5
DeepAR	0.453	0.583	4.5
MS Azure	0.118	0.218	7.5
TadGAN	0.704	0.623	2.5
OCPAE	**0.806**	**0.625**	**1.0**

4.3 Experimental Setup

4.3.1 Baselines: The proposed approach is compared with a wide range of non-deep learning and deep learning methods in multivariate time series anomaly detection, including: ARIMA [41], Iforest [25], LSTM-AE [15], DAGMM [42], MAD-GAN [23], TadGAN [14], OmniAnomaly [38], USAD [5], LSTM [19], MSCRED [40], Deep SVDD [33], GTA [9], and THOC [35]. We also report the performance of commercial tools excerpted from [14] including Amazon DeepAR [34] and Azure Anomaly Detector [32]. We implemented some of the baselines [14, 15, 19] to get the results, while for the rest of the baselines, we directly use the results from the literature.

4.3.2 Data Preprocessing: For SWaT and WADI, In order to fairly compare with other methods, the original data and labels of SWaT and WADI are downsampled to one measurement every 10 seconds by taking the median value and the most common label in each 10-second period following [5]. For MSL and SMAP, only the telemetry value is used as the input feature. For each dataset, we normalized both the training and testing data using the standard deviation and the mean of the values in the training data. We then applied a sliding window to create input sub-sequences of the time series.

4.3.3 Training Setting: We implement our approach and several baselines using Pytorch version 1.6.0 with CUDA 10.1 and test them on an Amazon p3.2xlarge EC2 with one NVIDIA Tesla P100 GPU. The models are trained using the Adam optimizer for 100 epochs with learning rate initialized as 0.001, and decay by 0.1 at epoch 40.

4.4 RQ1. Performance and Analysis

Table 2 shows the results on SMAP and MSL using segment-wise $F1^{**}$. From the table, it can be seen that our proposed OCPAE outperforms all the baselines in both datasets. Compared to LSTM-AE, our proposed method improves the $F1^{**}$ by 7.3% and 6.7% on SMAP and MSL, respectively. This indicates that the prediction branch and one-class branch in our framework can help with the anomaly detection task. Meanwhile, most recent state-of-the-art work are adopting point-wise $F1^{*}$ as the metric, and therefore, we also compare our framework with those baselines and the results are shown in Table 3. Our method exceeds the best performing state-of-the-art methods (*i.e.*, THOC and GTA) and achieves the highest average rank (1.5) on four datasets. More specifically, OCPAE ranked the 1^{st} on SMAP and MSL datasets, and the 2^{nd} on SWaT and WADI datasets. Overall, these experimental results clearly show

Table 3: Point-adjusted $F1^{*}$ of OCPAE and baselines: OCPAE achieves the highest average rank on four datasets.

Methods	SMAP	MSL	SWaT	WADI	avg. rank
Iforest	0.474	0.617	0.831	0.620	9.0
LSTM	0.873	0.910	0.845	0.732	5.5
LSTM-AE	0.904	0.912	0.846	0.686	4.0
DAGMM	0.775	0.854	0.797	0.209	9.3
deep SVDD	0.881	0.717	0.828	N/A	8.3
MSCRED	0.860	0.775	0.863	N/A	7.0
OmniAnomaly	0.853	0.901	0.833	0.417	7.8
USAD	0.863	0.927	0.846	0.430	5.3
THOC	0.952	0.937	0.880	N/A	2.3
GTA	0.904	0.911	**0.910**	**0.840**	2.5
OCPAE	**0.975**	**0.966**	0.885	0.746	**1.5**

Table 4: Anomaly detection results of OCPAE and baselines on PHM21. OCPAE achieved the highest score among unsupervised baselines. It even surpassed the supervised learning baseline LDM which ranked 3^{rd} place in the challenge.

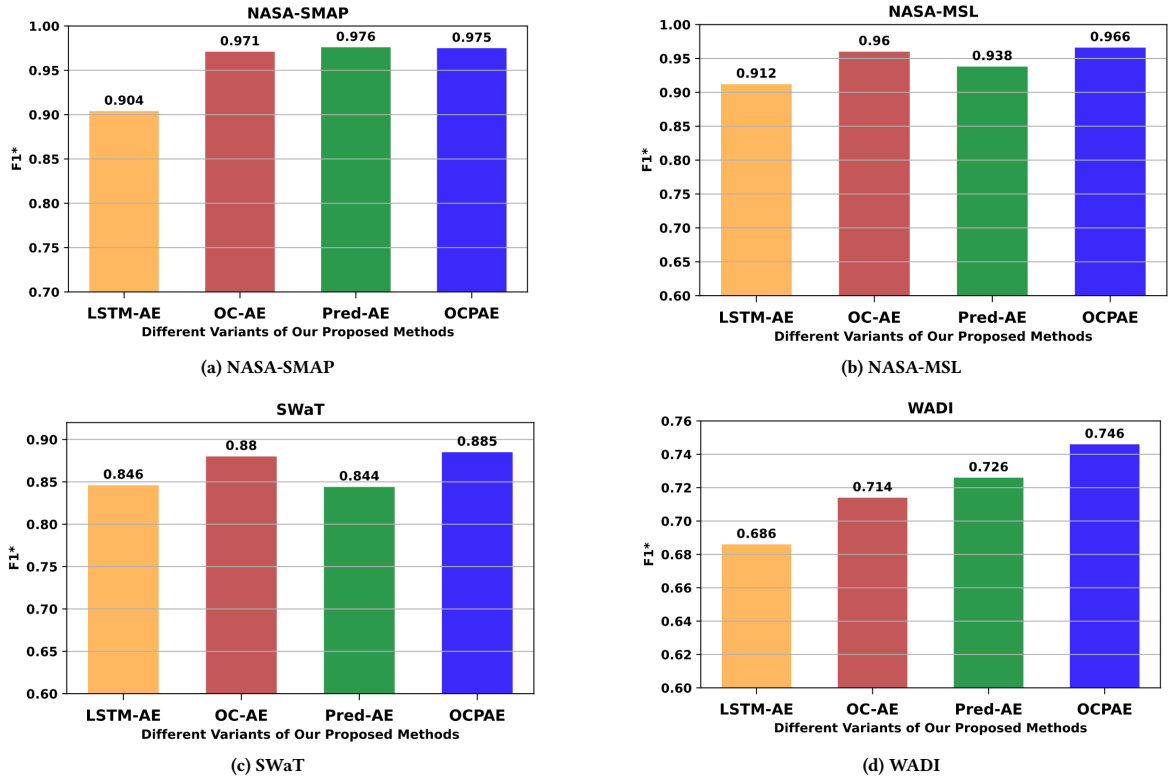
	Methods	Precision	Recall	$F1$
Unsupervised	LSTM	0.434	0.793	0.561
	LSTM-AE	0.434	0.793	0.561
	OCPAE	0.961	0.833	0.892
Supervised	HIRUTEK	1	0.897	0.945
	LDM	0.862	0.893	0.877

the effectiveness and superiority of OCPAE compared with other baselines, and more detailed analysis are provided in the following paragraphs.

LSTM [19] is a prediction-based method and mainly models the temporal dependency. However, some real-world time series are hard to predict as they may be affected by many external factors. For example, there are a wide variety of behaviors with different regularities that affect the MSL telemetry value [20], which explains why LSTM has much lower segment-wise $F1^{**}$ on MSL dataset compare to LSTM-AE and OCPAE. In OCPAE, we have the reconstruction and one-class branch that learns normal patterns of the data. As a result, it works well with unpredictable time series. Compared to LSTM-AE [15], our proposed method improves both $F1^{*}$ and $F1^{**}$ on all four datasets. Again, this shows the effectiveness of the proposed prediction branch and one-class branch in our framework, and we will provide the ablation studies in section 4.5.

Iforest [25], DAGMM [42] and MAD-GAN [23] presents the lowest overall performance using the two metrics. These are two unsupervised anomaly detection methods that mainly model the dependency in between features while are weak at modeling temporal dependencies[24, 38]. In OCPAE, the prediction branch takes a sequence of past samples as input and try to model the underlying temporal dependency to retain this information and use it to detect anomaly from local perspective.

THOC [35] and GTA [9] have an average rank of 2.3 and 2.5 respectively, which are better than LSTM-AE. Compared to THOC,



(a) NASA-SMAP (b) NASA-MSL

(c) SWaT (d) WADI

Figure 3: Ablation studies with different variants of our proposed methods on four datasets.

our proposed methods achieved a higher score on all datasets. Unlike THOC which applies hierarchical one-class network for representation learning, our proposed OCPAE can benefit from both prediction and reconstruction branches to learn a richer representation of normal data. As a result, it works well with these datasets. GTA achieves the highest $F1^*$ on SWaT and WADI because these two datasets have strong inter-feature dependencies. In OCPAE, the one-class branch can embed all the normal latent representation to one centroid, and learn the normality of global patterns. Therefore, OCPAE works well for datasets with weak inter-feature dependencies as well.

We also test our approach with PHM21 dataset and compare with the winners from the challenge. We aim at detecting time series which have anomalies at a time-series level instead of point or segment level. Thus, we first calculate the anomaly score for each sample in the subset using the trained model, and then use the median of all anomaly scores as the final anomaly score for the subset. The results are shown in Table 4. Since the labels are provided in the challenge, top teams mainly used supervised-based and rule-based methods, such as team HIRUTEK [12] and LDM [3], who rank at 1^{st} and 3^{rd} place in the challenge. From the table, OCPAE has the highest score among the unsupervised methods, and are able to improve $F1$ by 15.2 % compared to the best of other unsupervised methods. Compared with supervised methods, our results are higher than LDM but lower than HIRUTEK. The main reason is that HIRUTEK utilizes a data processing step along with supervised learning methods by manually examining the characteristic of both the normal and anomalous data, while OCPAE is trained using the normal data only.

4.5 RQ2. Ablation Studies

In this section, we conduct ablation studies to investigate the effectiveness of each branch in OCPAE. We test several variants of OCPAE by gradually excluding the branches and evaluating on SMAP, MSL, SWaT, and WADI datasets. The results are shown in Fig 3. From the figure, we can see our proposed OCPAE achieves an average of 6.8% improvement on $F1^*$ over four datasets, compared to LSTM-AE. To achieve this, both prediction branch and one-class branch play an important role in it.

One-class branch: By adding one-class branch, we see an average of 4.2% improvement on $F1^*$, comparing to LSTM-AE. Fig 4. (a) is an example from SMAP to show the effectiveness of the one-class branch. The red rectangle area is the anomalous event. From the figure, the reconstruction branch can reconstruct anomalies and has small S_{AE} for anomalous points in red rectangle area, and therefore, cannot detect the anomalies. With one-class branch, the model learns the normality of the latent space of the training set, and assigns anomalous points with large S_{OC}. This helps identify the anomalous event in this example.

Prediction branch: By adding the prediction branch, we see an average of 5.2 % improvement on $F1^*$, comparing to LSTM-AE. Fig 4. (b) is an example from SMAP to illustrate the effectiveness of the prediction branch. The normal data jumps between -1 and 1 frequently. For the anomalous event in red rectangle area, data

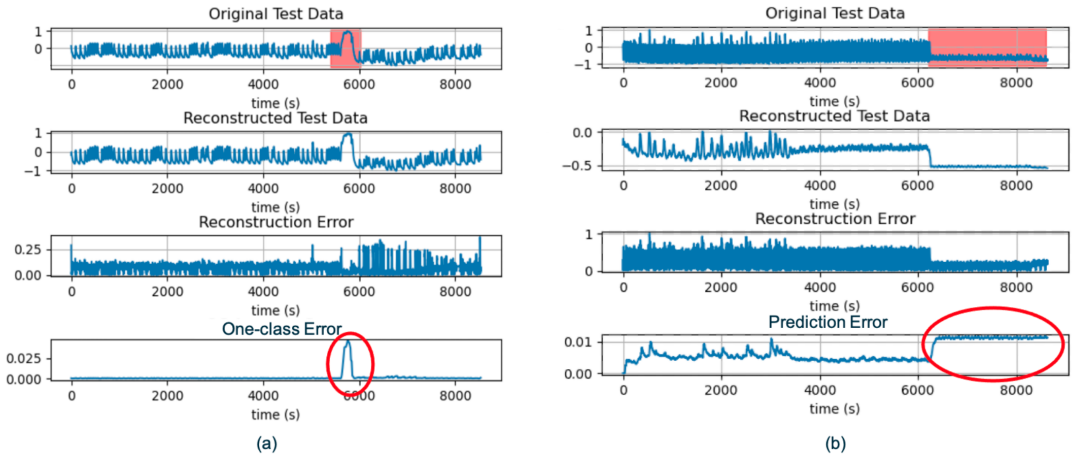
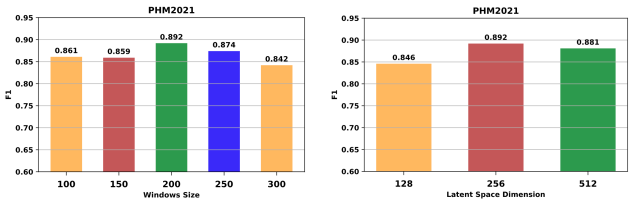


Figure 4: Case Studies on SMAP dataset. Although the reconstruction branch failed to detect the anomaly events in (a) and (b), the one-class branch and prediction branch are effective on each case respectively.



(a) Study on Window Size **(b) Study on Latent Space Dimension**

Figure 5: Parameter sensitivity study on $F1$ using PHM21 dataset. Our proposed model is robust in terms of the input window size and latent space dimension.

samples hover around -0.8. The reconstruction branch can reconstruct the anomalous samples reducing the S_{AE} for samples in the anomalous event and thereby, fails to detect the anomalies. With the prediction branch, it is clear that the model learned the local temporal dependency that normal samples will change frequently and will not be stable at any value. Thus, from the figure, the prediction branch assigns higher S_{Pred} on the samples in the anomalous event and can successfully detect the anomaly.

In summary, both prediction and one-class branches improve our performance on four datasets, and ablation of any of the branches leads to poorer results.

4.6 RQ3. Sensitivity and Efficiency

In this section, we study the parameter sensitivity and efficiency of our proposed OCPAE. These factors play a key role in determining if the model is robust and feasible to be deployed in real productions.

4.6.1 Parameter sensitivity. In this subsection, we study how window size and latent space dimension affect the results of OCPAE, which are the key parameters of concern in deployment. In general, the selection of window size and latent space dimension depends on which types the anomalous behaviors and temporal dependencies are. We evaluate how much these two parameters affect $F1$ of our model using PHM21 dataset .

In Fig. 5, five different window sizes $l_s \in [100, 150, 200, 250, 300]$ are selected and it can be observed that OCPAE is not very sensitive to the window size. The reason is OCPAE has three different branches that can deal with both short-term and long-term temporal dependencies. Thus, OCPAE is not sensitive to different window sizes. We then try three different latent space dimensions $m \in [128, 256, 512]$. Results show that with a smaller latent space dimension (*e.g.*, 128) our model has a lower $F1$, which may due to information loss at the encoding stage and affects all branches.

Table 5: Computational time on PHM21 dataset.

Methods	Training (per epoch)	Inference (per sample)
LSTM-AE	3.9 s	0.27 ms
OCPAE	6.12 s	0.41 ms

4.6.2 Computational time: Another key factor to determine if our model is feasible for deployment is the computational time. Since it has two additional branches compared to LSTM-AE, we would like to see how much more time is needed to train OCPAE and infer on the test dataset. Since PHM21 dataset is collected from an industrial setting and has the largest number of features and timestamps among five datasets we use in Table 1, we select it as an example to measure the training and inference time for OCPAE and LSTM-AE and the results are provided in Table 5. By adding additional prediction and one-class branches, we see around 57% and 52% increase in training and inference time compared with LSTM-AE. The inference time is only 0.41 ms per sample. Thus, OCPAE is a viable candidate for real-time anomaly detection tasks.

5 CONCLUSIONS AND FUTURE WORK

In this work, we proposed One-Class Predictive Autoencoder (OCPAE) for anomaly detection of time series datasets. To mitigate the limitation of the autoencoder-based reconstruction branch, we added a prediction branch and a one-class branch to detect anomalies from both local and global perspectives. We conducted extensive experiments on five real-world industrial datasets to demonstrate the superiority and effectiveness of our proposed framework. The results show that OCPAE has the highest average rank among all state-of-the-art unsupervised algorithms for five datasets, and has results close to the supervised algorithm that wins the PHM21 data challenge. We also provided ablation studies to explain how each branch helps with anomaly detection from different perspectives, and demonstrated that the combination of the three branches

allows OCPAE to take advantage from all of them while mitigating their limitations. In addition, we demonstrated its efficiency in training and inference, and robustness to parameter choices through case studies. This makes OCPAE scalable and robust for real-time anomaly detection tasks. In the future, OCPAE can be further enhanced by exploring advanced architectures to extract hierarchical features and better hyperparameter selection to support a diverse set of anomaly and data characteristics.

REFERENCES

[1] Nicola Acito, Marco Diani, and Giovanni Corsini. 2005. Gaussian mixture model based approach to anomaly detection in multi/hyperspectral images. In *Image and Signal Processing for Remote Sensing XI*, Vol. 5982. International Society for Optics and Photonics, 59820O.

[2] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P Mathur. 2017. WADI: a water distribution testbed for research in the design of secure cyber physical systems. In *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks*. 25–28.

[3] Osarenren Kennedy Aimiyekagbon, Lars Muth, Meike Wohlleben, Amelie Bender, and Walter Sextro. 2021. Rule-based Diagnostics of a Production Line. In *PHM Society European Conference*, Vol. 6(1). 10–10.

[4] Jinwon An and Sungzoon Cho. 2015. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE* 2, 1 (2015), 1–18.

[5] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. 2020. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3395–3404.

[6] Md Abul Bashar and Richi Nayak. 2020. TAnoGAN: Time series anomaly detection with generative adversarial networks. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 1778–1785.

[7] 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*. 93–104.

[8] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)* 41, 3 (2009), 1–58.

[9] Zekai Chen, Dingshuo Chen, Xiao Zhang, Zixuan Yuan, and Xiuzhen Cheng. 2021. Learning Graph Structures with Transformer for Multivariate Time Series Anomaly Detection in IoT. *IEEE Internet of Things Journal* (2021).

[10] Fangzhou Cheng, Ajay Raghavan, Deokwoo Jung, Yukinori Sasaki, and Yosuke Tajika. 2019. High-accuracy unsupervised fault detection of industrial robots using current signal analysis. In *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE, 1–8.

[11] Fangzhou Cheng, Jun Wang, Liyan Qu, and Wei Qiao. 2017. Rotor-current-based fault diagnosis for DFIG wind turbine drivetrain gearboxes using frequency analysis and a deep classifier. *IEEE Transactions on Industry Applications* 54, 2 (2017), 1062–1071.

[12] Kerman López de Calle-Etxabe, Meritxell Gómez-Omella, and Eider Garate-Perez. 2021. Divide, Propagate and Conquer: Splitting a Complex Diagnosis Problem for Early Detection of Faults in a Manufacturing Production Line. In *PHM Society European Conference*, Vol. 6(1). 9–9.

[13] Ailin Deng and Bryan Hooi. 2021. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35(5). 4027–4035.

[14] Alexander Geiger, Dongyu Liu, Sarah Alnegheimish, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2020. TadGAN: Time series anomaly detection using generative adversarial networks. In *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 33–43.

[15] André Gensler, Janosch Henze, Bernhard Sick, and Nils Raabe. 2016. Deep Learning for solar power forecasting—An approach using AutoEncoder and LSTM Neural Networks. In *2016 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE, 002858–002865.

[16] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. 2019. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1705–1714.

[17] Mahmudul Hasan, Md Milon Islam, Md Ishrak Islam Zarif, and MMA Hashem. 2019. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet of Things* 7 (2019), 100059.

[18] Ville Hautamaki, Ismo Karkkainen, and Pasi Franti. 2004. Outlier detection using k-nearest neighbour graph. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, Vol. 3. IEEE, 430–433.

[19] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[20] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 387–395.

[21] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. 2017. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215* (2017).

[22] Kwei-Herng Lai, Daochen Zha, Junjie Xu, Yue Zhao, Guanchu Wang, and Xia Hu. 2021. Revisiting Time Series Outlier Detection: Definitions and Benchmarks. (2021).

[23] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. 2019. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. In *International Conference on Artificial Neural Networks*. Springer, 703–716.

[24] Zhihan Li, Youjian Zhao, Jiaqi Han, Ya Su, Rui Jiao, Xidao Wen, and Dan Pei. 2021. Multivariate Time Series Anomaly Detection and Interpretation using Hierarchical Inter-Metric and Temporal Embedding. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3220–3230.

[25] 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.

[26] Junshui Ma and Simon Perkins. 2003. Time-series novelty detection using one-class support vector machines. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, Vol. 3. IEEE, 1741–1745.

[27] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. 2016. LSTM-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148* (2016).

[28] Larry M Manevitz and Malik Yousef. 2001. One-class SVMs for document classification. *Journal of machine Learning research* 2, Dec (2001), 139–154.

[29] Aditya P. Mathur and Nils Ole Tippenhauer. 2016. SWaT: a water treatment testbed for research and training on ICS security. *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)* (2016), 31–36.

[30] Daehyung Park, Yuuna Hoshi, and Charles C Kemp. 2018. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation Letters* 3, 3 (2018), 1544–1551.

[31] Prognostics and Health Management Society. 2021. DATA CHALLENGE. https://phm-europe.org/data-challenge

[32] Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. 2019. Time-series anomaly detection service at microsoft. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 3009–3017.

[33] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In *International conference on machine learning*. PMLR, 4393–4402.

[34] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting* 36, 3 (2020), 1181–1191.

[35] Lifeng Shen, Zhuocong Li, and James Kwok. 2020. Timeseries anomaly detection using temporal hierarchical one-class network. *Advances in Neural Information Processing Systems* 33 (2020), 13016–13026.

[36] Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. 2003. *A novel anomaly detection scheme based on principal component classifier*. Technical Report. MIAMI UNIV CORAL GABLES FL DEPT OF ELECTRICAL AND COMPUTER ENGINEERING.

[37] SA Spiewak, R Duggirala, and K Barnett. 2000. Predictive monitoring and control of the cold extrusion process. *CIRP Annals* 49, 1 (2000), 383–386.

[38] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. 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*. 2828–2837.

[39] RCM Yam, PW Tse, L Li, and P Tu. 2001. Intelligent predictive decision support system for condition-based maintenance. *The International Journal of Advanced Manufacturing Technology* 17, 5 (2001), 383–391.

[40] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V Chawla. 2019. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 1409–1416.

[41] G Peter Zhang. 2003. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* 50 (2003), 159–175.

[42] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. 2018. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*.