Calibrated One-class Classification for Unsupervised Time Series Anomaly Detection

Hongzuo Xu, Yijie Wang, Songlei Jian, Qing Liao, Yongjun Wang and Guansong Pang

Abstract—Time series anomaly detection is instrumental in maintaining system availability in various domains. Current work in this research line mainly focuses on learning data normality deeply and comprehensively by devising advanced neural network structures and new reconstruction/prediction learning objectives. However, their one-class learning process can be misled by latent anomalies in training data (i.e., anomaly contamination) under the unsupervised paradigm. Their learning process also lacks knowledge about the anomalies. Consequently, they often learn a biased, inaccurate normality boundary. To tackle these problems, this paper proposes calibrated one-class classification for anomaly detection, realizing contamination-tolerant, anomaly-informed learning of data normality via uncertainty modeling-based calibration and native anomaly-based calibration. Specifically, our approach adaptively penalizes uncertain predictions to restrain irregular samples in anomaly contamination during optimization, while simultaneously encouraging confident predictions on regular samples to ensure effective normality learning. This largely alleviates the negative impact of anomaly contamination. Our approach also creates native anomaly examples via perturbation to simulate time series abnormal behaviors. Through discriminating these dummy anomalies, our one-class learning is further calibrated to form a more precise normality boundary. Extensive experiments on ten real-world datasets show that our model achieves substantial improvement over sixteen state-of-the-art contenders.

Index Terms—Anomaly Detection, One-class Classification, Time Series, Anomaly Contamination, Native Anomalies.

1 Introduction

Over recent decades, with the burgeoning of informatization, a substantial amount of time series data have been created. As the functioning status of various target systems such as large-scale data centers, cloud servers, and space crafts, these time series data are a source where we can monitor and alarm potential faults, threats, and risks of target systems by identifying their unusual states (i.e., anomalies). Anomaly detection, an important field in data mining and analytics, is to find exceptional data observations that deviate significantly from the majority [1], which plays a critical role in achieving this goal. Due to the cost and difficulty of labeling work in these real-world applications, time series anomaly detection is often formulated as an unsupervised task with unlabeled training data.

Challenges. Without any guidance of supervisory signals, unsupervised time series anomaly detection generally relies on modeling data normality via one-class learning because most training samples are normal. However, this learning process still faces two key challenges: (1) *the presence of anomalies in training sets*, and (2) *the absence of knowledge about the anomalies*. Specifically, as has been done in mainstream methods like [2], [3], [4], [5], the whole training set is often directly fed into learning models by assuming all training samples are normal. However, this assumption does not always hold. The training set might not be pure but contaminated by a small part of anomalies. Anomaly contamination can greatly disturb the learning process, thus leading to severe overfitting problems. Besides, the learning process, without any knowledge about anomalies, may find an inaccurate normality boundary, since it is hard to define the range of normal behaviors in such cases. As shown in Fig. 1 (a), due to these two problems, canonical one-class classification methods typically learn a suboptimal normality model.

Prior Art. Most unsupervised time series anomaly detectors use generative models in one-class learning to restore input data [3], [4], [6], [7] or forecast future data [2], [8], [9]. Data normality is implicitly learned behind the rationale of reconstruction or prediction. The abnormal degrees of observations in time series can be measured according to loss values. To achieve a comprehensive delineation of data normality (e.g., deeper inter-metric correlations, longer-term temporal dependence, and more diverse patterns), these methods design advanced network structures (e.g., using variational Autoencoders [7], [10], graph neural networks [2], [9], and Transformer [3], [5]) and new reconstruction/prediction learning objectives (e.g., adding adversarial training [4], [11], [12], ensemble learning [6], [13], and meta-learning [3]). However, these current methods generally do not have components to deal with the anomaly contamination issue. There are a few attempts to address this problem, e.g., using pseudo-labels via self-training [14], [15], [16] or

- *Hongzuo Xu is with the Intelligent Game and Decision Lab (IGDL), Beijing 100091, PR China. (e-mail: hongzuoxu@126.com)*
- *Yijie Wang is with the National Key Laboratory of Parallel and Distributed Computing, College of Computer, National University of Defense Technology, Hunan 410073, PR China. (e-mail: wangyijie@nudt.edu.cn)*
- *Songlei Jian and Yongjun Wang are with the College of Computer, National University of Defense Technology, Hunan 410073, PR China. (e-mail: {jiansonglei, wangyongjun}@ nudt.edu.cn)*
- *Qing Liao is with the School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Guangdong 518055, PR China. (e-mail: liaoqing@hit.edu.cn)*
- *Guansong Pang is with the School of Computing and Information Systems, Singapore Management University Singapore, 178902, Singapore. (e-mail: gspang@smu.edu.sg)*
- *Yijie Wang is the corresponding author.*

an extra pre-positive one-class classification model [12] to remove plausible anomalies in the training set. Nonetheless, these additional components themselves might be biased by the anomaly contamination, leading to high errors in the pseudo labeling or anomaly removal. They may also wrongly discard some normal samples of the boundary that are informative and vital in learning data normality. On the other hand, these methods do not consider information related to the anomaly class when performing their normality learning process. It is difficult to learn accurate normality without knowing what the abnormalities are.

New Insights. To address these challenges, this paper investigates an intriguing question: *Can we calibrate one-class classification from two facets, i.e., alleviating the negative impact of anomaly contamination and introducing knowledge about anomalies, to learn a contamination-tolerant, anomaly-informed data normality?*

As for the first facet, the essence is to eliminate the contribution of these noisy samples in the learning process. We resort to model uncertainty to tackle this problem. These anomalies are typically accompanied by rare and inconsistent behaviors, and as a result, the one-class learning model tends to make predictions unconfidently. As shown in Fig. 1 (c), we aim to use this type of uncertainty to weaken the contribution of anomaly contamination, thereby calibrating the one-class model w.r.t. the contaminated training data. Particularly, we can design a new learning objective embedded with the uncertainty concept. It adaptively penalizes uncertain predictions, while simultaneously encouraging more confident predictions to ensure effective learning of hard samples. Therefore, this process can discriminate these harmful anomalies in learning data normality, thus masking the notorious anomaly contamination problem during network optimization.

To address the second facet, we are motivated by self-supervised learning that creates supervisory signals from the data itself. Current self-supervised anomaly detection methods design various supervised proxy tasks, e.g., geometric transformation prediction [17], masking prediction [18], and continuity identification [19], but these tasks mainly contribute to learning clearer semantics of the input data rather than introducing information related to anomalies. Since time series anomalies are well characterized and defined as point, contextual, and collective anomalies [20], we aim to utilize these definitions and characterizations to simulate abnormal behaviors by tailored data perturbation operations upon original time series data. This process can offer reliable primitive anomaly examples, or at least data samples with abnormal semantics, to the one-class learning process. As shown in Fig. 1 (d), these *native anomaly examples* ("native" indicates that they are generated based on original data) can further help calibrate the discriminability of the learned normality.

The Proposed Approach. Based on the above motivation and insights, this paper proposes a novel Calibrated One-class classification-based Unsupervised Time series Anomaly detection method (COUTA for short). The approach fulfills contamination-tolerant, anomaly-informed normality learning by two novel normality calibration methods, including *uncertainty modeling-based calibration (UMC)* and *native anomaly-based calibration (NAC)*. In UMC, a novel

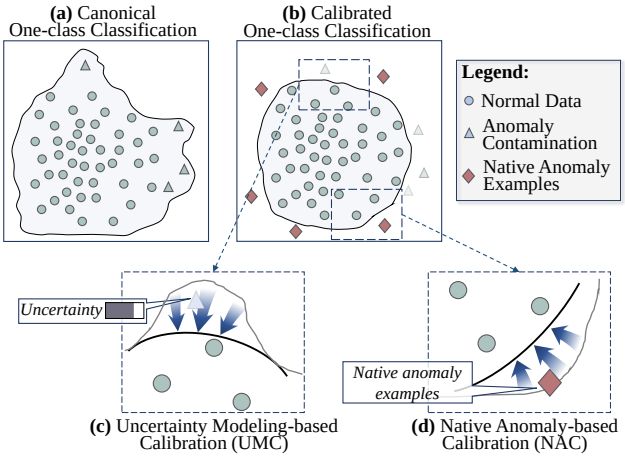
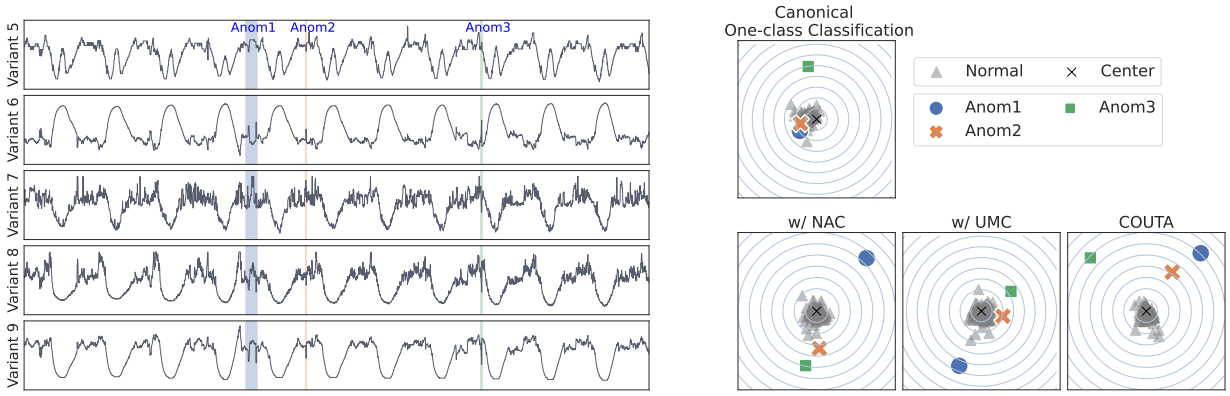


Fig. 1. Demonstration of our key insights. **(a)** Broadly-used canonical one-class classification may learn an inaccurate, biased normality boundary due to the anomaly contamination problem and the absence of knowledge about anomalies. **(b)** By contrast, the two proposed calibration methods, UMC and NAC, respectively address these two issues, and our calibrated one-class classifier can produce a more accurate, clearer normality boundary. **(c)** UMC helps weaken contaminated data during optimization based on model uncertainty scores, while (**d**) NAC helps ground the normality boundary by calibrating the normality with native anomaly examples.

calibrated one-class loss function is proposed, in which a prior (i.e., Gaussian) distribution is imposed on the one-class distances and utilized to capture the prediction uncertainties. It calibrates one-class representations by adaptively penalizing uncertain predictions while at the same time emphasizing confident predictions. This prior is theoretically motivated by the probability density function of Gaussian distribution. In NAC, we propose three simple but effective data perturbation operations to generate native anomaly examples based on original time series sub-sequences. A new supervised training branch is devised to further calibrate the learned normality to be discriminative w.r.t. these anomaly-informed samples. By jointly optimizing these two components, our calibrated one-class classification model COUTA is enforced to be robust to anomaly contamination and discriminative to primeval anomalous behaviors, thus producing a more accurate data normality boundary, as depicted in Fig. 1 (b).

As an example, we use the *Omi-4* server data of the real-world Application Server Dataset (*ASD*) [7] to have a straightforward demonstration of the benefit of each calibration to our model in Fig. 2, where Fig. 2 (a) visualizes the time series data (five representative dimensions out of the original nineteen dimensions are selected) while Fig. 2 (b) displays the new feature spaces learned by four different one-class classification models. It is clear that the canonical one-class classification (i.e., COUTA without calibration) fails to identify two anomalies (Anom1 and Anom2), and the hypersphere formed by normal data is also biased. By contrast, adding either NAC or UMC all effectively calibrates the normality of the data, resulting in better discrimination of real anomalies from normal data. As a result, by adding both calibrations, COUTA can easily differentiate all three anomalies of diverse abnormal behaviors.

The contributions are summarized as follows.



(a) **Time Series Data** (b) **Feature Space**

Fig. 2. **(a)** Time series data with three anomaly segments; **(b)** Learned feature space of canonical (non-calibrated) one-class classification and the proposed methods by using NAC/UMC separately and two calibration methods simultaneously (i.e., COUTA). Normal data is expected to be enclosed in a compact hypersphere, and anomalies can be successfully identified if they are distant from the center.

- We propose the calibrated one-class classification method COUTA, which calibrates one-class learning using prediction uncertainties and native anomalies. These two calibrations result in a contamination-tolerant, anomaly-informed COUTA.
- We propose the uncertainty modeling-based calibration, UMC. It restrains irregular noisy training data, while at the same time encouraging confident predictions on regular samples to ensure an effective normality learning process. This largely alleviates the negative impact of anomaly contamination.
- We propose the native anomaly-based calibration, NAC. It generates different types of anomaly examples and wields them to calibrate one-class representations for learning a more precise abstraction of normality and a clearer normality boundary.

Extensive experiments show that: (1) COUTA substantially outperforms 15 state-of-the-art competing methods on 10 real-world datasets and averagely achieves over 11% improvement; and (2) COUTA is also with several desired properties including generalization capability in identifying different anomaly types, favorable robustness to anomaly contamination, and good scalability w.r.t. both length and dimensionality of time series.

2 Related Work

2.1 Anomaly Detection in Time Series

Time series anomaly detection is an old discipline. There is a long list of traditional methods in the literature using various techniques like decomposition, clustering, distance, and pattern mining. Besides, traditional time series prediction models such as moving average, autoregressive, and their multiple variants are adapted to detect anomalies by comparing the predicted values and the real ones. Reviews of traditional time series anomaly detection can be found in [21], and we also recommend some comprehensive time series anomaly detection benchmark studies on univariate data [22], multivariate data [23], and explainability [24].

Deep learning fuels many deep time series anomaly detection methods. They use generative one-class learning models to restore input data or predict future data as precisely as possible. Prior work generally categorizes these current studies into reconstruction- and prediction-based methods. As the training set is dominated by normal data, reconstruction/prediction errors can indicate abnormal degrees. The core insight of these methods is to implicitly model normal patterns and behaviors via the rationale behind restoring or forecasting. The pioneer in this research line [25] uses the Long-Short Term Memory (LSTM) network in an encoder-decoder structure. In recent years, the data mining community has made tremendous efforts to successfully enhance the performance of this pipeline by devising various advanced network structures and new reconstruction/prediction learning objectives. A number of studies focus on capturing more comprehensive temporal and inter-variate dependencies by using graph neural networks [2], [9], convolutional kernels [6], [26], and variational Autoencoders [7], [10]. Besides, adaptive memory network [27] and hierarchical structure-based multi-resolution learning [26], [28], [29] are used to better handle diverse normal patterns. Some other methods use adversarial training in Autoencoder [4], [11], [12] or generative adversarial network [11], [14], [30] that introduce regularization into the learning process to alleviate the overfitting problem. Additionally, ensemble learning is also explored in [6], [13]. A very recent work [3] employs Transformer [31] to effectively model long-term trends in time series data, and several tools are used as building blocks to further enhance the detection model, including adversarial training to amplify reconstruction errors, self-conditioning for better training stability and generalization, model agnostic meta-learning to handle the circumstance that only limited data are available. It is noteworthy that some studies propose new anomaly scoring strategies to replace reconstruction or prediction errors. A Transformer-based method [5] uses association variance as a novel criterion to measure abnormality, and the literature [32] employs the Bayesian filtering algorithm for anomaly scoring. Various advanced techniques are equipped in this pipeline, achieving state-of-the-art performance. Nevertheless, these methods may still considerably suffer from the presence of anomaly contamination and the absence of knowledge about anomalies. We below review techniques related to solving these two key problems.

2.2 Anomaly Contamination and Label-noise Learning

A few anomaly detection methods consider the anomaly contamination problem. The literature [14], [15], [16] filters possible anomalous samples via self-training. An additional Autoencoder is used in [12] to obtain a clean set of time series data before the training process. A recent work [33] jointly infers binary labels to each sample (normal vs. anomalous) while updating the model parameters. This work applies two coupled losses that are respectively designed for normal and anomalous data. These methods attempt to use the abnormality derived from the original/additional one-class learning component to filter these noises. However, these filtering processes also suffer from the anomaly contamination problem, and these methods may also wrongly filter some hard normal samples which are important to the network training. This problem is also related to label-noise learning or inaccurate supervision because these hidden anomalies are essentially training data with wrong labels. This topic is under the big umbrella of the weakly-supervised paradigm. A survey [34] categorizes the methodology of this research line into three perspectives, i.e., data, learning objective, and optimization policy. The proposed uncertainty modeling-based calibration in the one-class learning objective also contributes a novel solution to this research line.

2.3 Self-supervised Anomaly Detection

Creating supervisory signals from the data itself is an essential idea in self-supervised learning. Inspired by a number of self-supervised methods in image anomaly detection [35], some recent studies are designed for time series data. They assign class labels to different augmentation operations (e.g., adding noise, reversing, scaling, and smoothing) [27], neural transformations [36], contiguous and separate time segments [19], or different time resolutions [28]. However, although these proxy tasks can help to better learn data characteristics, these tasks still fail to provide information related to anomalies. They may neglect that it is also possible to reliably simulate abnormal behaviors in time series via simple data perturbation. We explore this direction in our proposed method, showing that these dummy anomaly examples can greatly benefit the learning process.

2.4 Anomaly Exposure

Providing extra anomaly information is a direct solution to address the absence of knowledge about anomalies. This idea is initially proposed by a work named outlier/anomaly exposure [37]. This study employs data from an auxiliary natural dataset as manually introduced out-of-distribution examples. Our work is fundamentally different from [37]. We create dummy anomaly examples by performing data perturbation on original data instead of taking data samples from a supplementary nature dataset. A concurrent study [38] also works on perturbation learning for anomaly detection in images, which constructs a perturbator and a classifier to perturb data with minimum efforts and correctly classify the normal data and perturbed data into two classes. Besides, DROCC [39] also adaptively generates anomalous points while training via a gradient ascent phase reminiscent of adversarial training. Different from these studies, we propose tailored perturbation methods for time-series data by fully considering the unique characteristics and clear definitions of anomalies in time series.

3 The Proposed Method: COUTA

3.1 Problem Formulation

Let $\mathcal{X} = \langle \boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N \rangle$ be a time series dataset, and $\mathcal{X}$ is an ordered sequence of N observations. Each observation in $\mathcal{X}$ is a vector described by D dimensions (i.e., $\boldsymbol{x}_t \in \mathbb{R}^D, \forall \boldsymbol{x}_t \in \mathcal{X}$). Dataset $\mathcal{X}$ is termed as multivariate time series when $D > 1$, and the dataset is reduced to the univariate setting if $D = 1$. Unsupervised time series anomaly detection f is to measure the abnormal degree of each observation and give anomaly scores without accessing any label information, i.e., $f : \mathcal{X} \mapsto \mathbb{R}$. Higher anomaly scores indicate a higher likelihood to be anomalies.

We consider a local contextual window of each observation to model their temporal dependence. Specifically, a sliding window with length l and stride r is used to transform the training set into a collection of sub-sequences $\mathcal{S} = \{\boldsymbol{s}_1, \boldsymbol{s}_{1+r}, \cdots\}$, where $\boldsymbol{s}_t = \langle \boldsymbol{x}_t, \boldsymbol{x}_{t+1}, \cdots, \boldsymbol{x}_{t+l-1} \rangle$. This is a common practice in most deep time-series anomaly detection methods [3], [6], [7], [10]. During the inference stage, the testing set is also split into sub-sequences using the same window length l, and the sliding stride is set to 1. The anomaly detection model evaluates the abnormal degree of each sub-sequence, and the anomaly score is assigned to the last timestamp of each sub-sequence. We use 0 to pad the beginning $l - 1$ timestamps to obtain the final anomaly score list.

3.2 Overall Framework

The overall framework COUTA is shown in Fig. 3. A temporal modeling network ϕ is used to model time-axis dependency and inter-variate interactions. We opt for Temporal Convolutional Network (TCN) [40] as the temporal modeling network ϕ in COUTA. TCN is more time-efficient than traditional RNN-based structures, and it can better capture local time dependency due to the usage of convolutional kernels. We further use a lightweight projection head ψ to map data into an H-dimensional feature space $\mathcal{F} \subset \mathbb{R}^H$. The multi-layer perceptron network structure is employed in ψ. The whole representation process can be denoted as $\psi(\phi(\cdot))$. We aim to map the training sample $\boldsymbol{s}$ into a compact hypersphere with center $\boldsymbol{c}$ upon the feature space $\mathcal{F}$. We devise the Uncertainty Modeling-based Calibration (UMC) in COUTA. Specifically, a prior probability distribution is imposed on the one-class distance. We set a bypass in ψ by using an additional layer, producing an extra prediction $\psi'(\phi(\boldsymbol{s}))$. Two distance values (i.e., d and $\tilde{d}$) are used to obtain the mean and the variance of the prior distribution. Our one-class learning objective $\mathcal{L}_{\text{UMC}}$ is calibrated to adaptively penalize uncertain predictions and simultaneously encourage confident predictions, thus accomplishing the masking of anomaly contamination in the training set. Besides, we propose Native Anomaly-based Calibration (NAC) to introduce knowledge about time series anomalies to the one-class learning process. Native anomaly example is created based on original time series data via

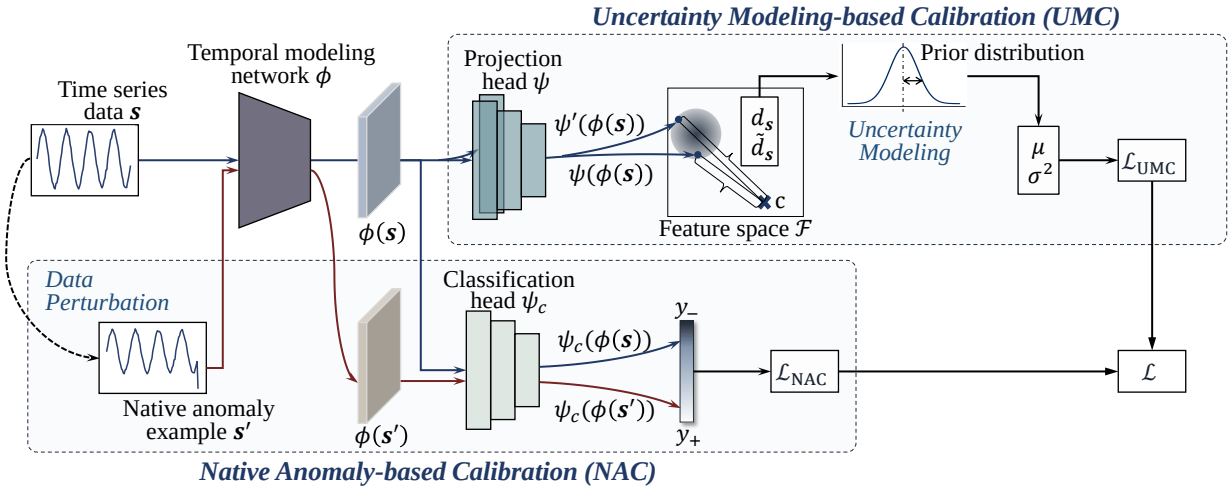


Fig. 3. The framework of COUTA.

tailored data perturbation operations. A new supervised training branch with a classification head ψ_c is added to empower COUTA to discriminate abnormal behaviors in time series via loss function $\mathcal{L}_{\text{NAC}}$. We also employ the multi-layer perceptron structure in ψ_c.

The final loss function is computed by assembling:

$$\mathcal{L} = \mathcal{L}_{\text{UMC}} + \alpha \mathcal{L}_{\text{NAC}}, \tag{1}$$

where α is a parameter to adjust the weight of the native anomaly-based classification branch.

The learnable parameters within the neural network are jointly trained by loss function $\mathcal{L}$. During the inference stage, the testing set is also pre-processed to sub-sequences, and COUTA measures abnormal degrees of input sub-sequences according to their deviation from the learned normality model (i.e., the hypersphere).

3.3 Calibrated One-class Classification

3.3.1 UMC for Contamination-tolerant One-class Learning

COUTA aims at learning a hypersphere with the minimum radius that can well enclose the training data upon the feature space $\mathcal{F}$. Therefore, the data normality can be explicitly defined as this hypersphere, and the distance to the hypersphere center can faithfully indicate the degree of data abnormality. This basic goal is the same as SVDD [41] (a popular technique in one-class classification). The traditional SVDD algorithm relies on the kernel trick. As has been done in a recent extension [42], after mapping original data to a new feature space via neural network ϕ and ψ, the canonical one-class loss function can be defined as

$$\mathcal{L}_{\text{canonical}} = \mathbb{E}_{\boldsymbol{s}\sim\mathcal{S}}\left[\left\|\psi(\phi(\boldsymbol{s})) - \boldsymbol{c}\right\|^2\right], \tag{2}$$

where $\boldsymbol{c} \in \mathbb{R}^H$ is the hypersphere center upon the feature space $\mathcal{F}$.

Anomaly contamination of the training set is essentially noisy data that have very rare abnormal behaviors and inconsistent patterns, and thus the one-class classification model tends to output unconfident predictions on these noisy data. Therefore, to address the anomaly contamination problem, we can give a relatively mild penalty to predictions that are with high model uncertainty, thus masking anomaly contamination in a soft manner. On the other hand, to ensure effective optimization of hard normal samples, the one-class classification model should also be encouraged to output confident predictions. The learning objective in Equation (2) is basically defined according to the distance to the hypersphere center. Therefore, the core idea is to impose a prior (i.e., Gaussian) distribution $\mathcal{N}_{\boldsymbol{s}}(\mu, \sigma^2)$ to the single distance value $d_{\boldsymbol{s}} = \|\psi(\phi(\boldsymbol{s})) - \boldsymbol{c}\|^2$, and the variance σ^2 of the distribution can represent the model uncertainty. Hence, to fulfill uncertainty modeling-based calibration, we need to answer two questions, i.e., *how to design a new learning objective that can handle distance distribution* and *how to extend the single distance value to its prior distribution*.

We first consider the design of our new one-class loss function. Given a prior distribution of the one-class distance, we need to maximize the probability of the distance being zero, instead of simply minimizing a single distance value. Based on the probability density function of the Gaussian distribution, the learning objective of the distance distribution $\mathcal{N}_s(\mu, \sigma^2)$ of sub-sequence $\boldsymbol{s}$ can be defined to maximize the following function:

$$J = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(\frac{\mu}{\sigma})^2}. \tag{3}$$

We can further derive the following function:

$$\log J = -\frac{1}{2\sigma^2}\mu^2 - \frac{1}{2}\log 2\pi\sigma^2. \tag{4}$$

We omit 2π and use ζ to indicate $\log \sigma^2$, and the learning objective of the Gaussian distribution $\mathcal{N}_{\boldsymbol{s}}(\mu, \sigma^2)$ is equivalent to minimize the following loss value:

$$\ell(\boldsymbol{s}) = \frac{1}{2} e^{-\zeta}\mu^2 + \frac{1}{2}\zeta. \tag{5}$$

We then address how to extend the single output of a distance value to a Gaussian distribution. One direct solution is to employ the ensemble method to obtain a group of predictions, thus estimating the mean and the variance of the distribution. However, the GPU memory consumption and the computational effort might be costly when there are

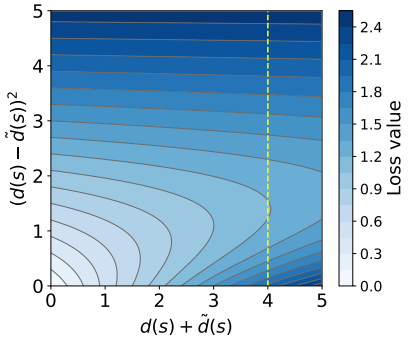


Fig. 4. Loss values in $\mathcal{L}_{\text{UMC}}$ w.r.t. $d_{\boldsymbol{s}} + \tilde{d}_{\boldsymbol{s}}$ and $(d_{\boldsymbol{s}} - \tilde{d}_{\boldsymbol{s}})^2$. As indicated by the yellow line, the penalty of a fixed $d_{\boldsymbol{s}} + \tilde{d}_{\boldsymbol{s}}$ is first adjusted to more mild levels with the increase of the uncertainty $(d_{\boldsymbol{s}} - \tilde{d}_{\boldsymbol{s}})^2$, while the loss function further penalizes heavily when the uncertainty reaches a high value.

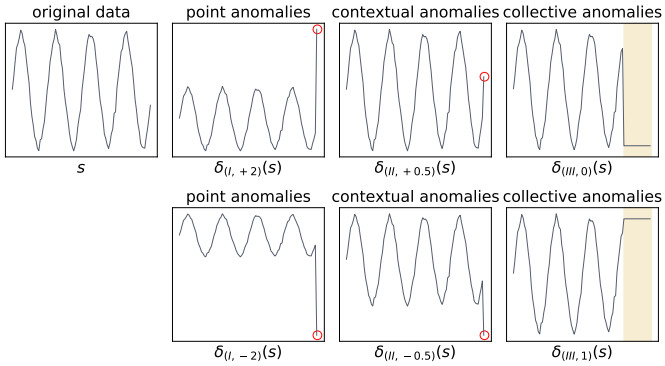


Fig. 5. Native anomaly examples generated from a time series subsequence $\boldsymbol{s}$ by six perturbation functions in Ω.

many ensemble members. The essence of the Gaussian distribution is to find the variance, that is, the ensemble process does not need a heavy computational overhead. Recall that a lightweight projection head ψ is used after the temporal modeling network ϕ, we can set a bypass hidden layer in ψ, yielding an additional projection output (denoted by $\psi'(\phi(\boldsymbol{s}))$). We calculate one-class distance values of $\psi(\phi(\boldsymbol{s}))$ and $\psi'(\phi(\boldsymbol{s}))$ as

$$d_{\boldsymbol{s}} = \|\psi(\phi(\boldsymbol{s})) - \boldsymbol{c}\|^2, \tilde{d}_{\boldsymbol{s}} = \|\psi'(\phi(\boldsymbol{s})) - \boldsymbol{c}\|^2. \quad (6)$$

We further employ $d_{\boldsymbol{s}} + \tilde{d}_{\boldsymbol{s}}$ and $(d_{\boldsymbol{s}} - \tilde{d}_{\boldsymbol{s}})^2$ to delegate μ^2 and ζ, respectively. Therefore, the loss function of one-class classification with uncertainty modeling-based calibration can be further derived as follows:

$$\mathcal{L}_{\text{UMC}} = \mathbb{E}_{\boldsymbol{s}\sim\mathcal{S}}\Big[\frac{1}{2}e^{-(d_{\boldsymbol{s}}-\tilde{d}_{\boldsymbol{s}})^2}(d_{\boldsymbol{s}} + \tilde{d}_{\boldsymbol{s}}) + \frac{1}{2}(d_{\boldsymbol{s}} - \tilde{d}_{\boldsymbol{s}})^2\Big]. \quad (7)$$

Our loss function in Equation (7) can mask anomaly contamination and weaken their contribution by assigning mild penalties. The one-class classification model tends to output inconsistent predictions on these hidden anomalies, i.e., $(d_{\boldsymbol{s}} - \tilde{d}_{\boldsymbol{s}})^2$ is high. Therefore, the loss value $d_{\boldsymbol{s}} + \tilde{d}_{\boldsymbol{s}}$ is adjusted to a lower level by its coefficient $e^{-(d_{\boldsymbol{s}}-\tilde{d}_{\boldsymbol{s}})^2}$. On the other hand, the second term also penalizes high uncertainty, which encourages more confident predictions to ensure the effective optimization of hard samples. Fig. 4 visualizes $\mathcal{L}_{\text{UMC}}$ of Equation (7) by presenting how loss values change w.r.t. $d_{\boldsymbol{s}} + \tilde{d}_{\boldsymbol{s}}$ and $(d_{\boldsymbol{s}} - \tilde{d}_{\boldsymbol{s}})^2$. As expected, this loss function adaptively adjusts the loss value of the data sample with high uncertainty and concurrently impels more confident predictions.

Note that our method is fundamentally different from the existing work based on Gaussian processes [43]. This work transfers the one-class classification problem to Gaussian Process Regression and approximates Gaussian Process classification with Laplace approximation or expectation propagation. Differently, our work imposes a Gaussian distribution to the one-class distance value, thus modeling the uncertainty during the one-class learning process.

3.3.2 NAC for Anomaly-informed One-class Learning

To introduce knowledge about anomalies, we propose to offer dummy anomaly examples to the one-class classification model. We introduce several tailored perturbation operations to generate native anomaly examples based on original time series data.

Data perturbation δ modifies the input time series subsequence $\boldsymbol{s}$ via a specific operation to obtain a new subsequence $\boldsymbol{s}'$, such that $\boldsymbol{s}'$ contains realistic abnormal behaviors of time series. $\boldsymbol{s}$ and $\boldsymbol{s}'$ are with the same shape. According to the definitions and characterizations of three basic time series anomaly types (i.e., point, contextual, and collective anomalies), we define the following data perturbation operations.

- $\delta_{(\text{I},\gamma)}(\boldsymbol{s})$: Data perturbation is performed on the last observation of the input sub-sequence $\boldsymbol{s}$. Data values of a group of randomly selected dimensions of the last observation are replaced with a global extreme value γ. A new sub-sequence with a point anomaly can be created.
- $\delta_{(\text{II},\gamma)}(\boldsymbol{s})$: Data perturbation is also applied to randomly selected dimensions of the last observation of the input sub-sequence $\boldsymbol{s}$. We use an offset γ based on the mean of the previous k values to pad these selected places. This perturbation method produces contextual anomalies. We use $k = 10$ by default.
- $\delta_{(\text{III},\gamma)}(\boldsymbol{s})$: We also randomly choose a group of dimensions, but this perturbation operation acts on a segment of the input sub-sequence $\boldsymbol{s}$. The right side is fixed as the last observation, and the segment length is randomly sampled from the range $[1, w]$. These values are replaced with an outlier value γ. This perturbation process produces collective anomalies. We use $w = 0.5l$ by default.

In practice, each of the above operations is deployed with two γ values to simulate anomalies at two extremes. As time series data is first preprocessed via data normalization to $[0, 1]$, we use $\gamma = +2, -2$ in δ_{I}, $\gamma = +0.5, -0.5$ in δ_{II}, $\gamma = 1, 0$ in δ_{III}. We define a pool Ω containing these six data perturbation functions, i.e.,

$$\Omega = \big\{\delta_{(\text{I},+2)}, \delta_{(\text{I},-2)}, \delta_{(\text{II},+0.5)}, \delta_{(\text{II},-0.5)}, \delta_{(\text{III},0)}, \delta_{(\text{III},1)}\big\}. \quad (8)$$

These six perturbation functions can simulate abnormal behaviors in time series data. Fig. 5 presents a base time series sub-sequence $\boldsymbol{s}$ and corresponding native anomaly examples generated via these six data perturbation operations within Ω.

A new set of dummy anomaly examples $\mathcal{S}'$ with size $\beta|\mathcal{S}|$ is generated, which is denoted as follows.

$$\mathcal{S}' = \{\delta(\boldsymbol{s})|\boldsymbol{s} \sim \mathcal{S}, \delta \sim \Omega\}, \tag{9}$$

where original sub-sequence $\boldsymbol{s}$ and function δ are randomly sampled from the training set $\mathcal{S}$ and the operation pool Ω.

We then set a new supervised training branch to calibrate COUTA such that our one-class classifier can discriminate primitive anomalies in $\mathcal{S}'$. We use an extra lightweight classification head ψ_c following the temporal modeling network ϕ to transfer each data sample to a score. ψ_c also uses the multi-layer perceptron structure. Mean squared error is employed to regress the output of sub-sequences of the original set to y_- and the output of those anomaly examples to y_+. The loss function of this branch is defined as:

$$\begin{aligned}\mathcal{L}_{\text{NAC}} = \mathbb{E}_{\boldsymbol{s}\sim\mathcal{S}\cup\mathcal{S}'}\Big[&\mathbb{1}_{\boldsymbol{s}\in\mathcal{S}}\big(\psi_c(\phi(\boldsymbol{s})) - y_-\big)^2 + \\ &\mathbb{1}_{\boldsymbol{s}\in\mathcal{S}'}\big(\psi'(\phi(\boldsymbol{s})) - y_+\big)^2\Big],\end{aligned} \tag{10}$$

where $y_+ = 1$ and $y_- = -1$ are used in our implementation.

Although these manually defined perturbation operations seem to be old-school compared to deep generation methods, this way is simple enough to generate dummy anomaly examples with abnormal behaviors of time series. Exposing these native anomaly examples to the one-class classification model via this training branch can lead to a more precise abstraction of data normality and a clearer boundary of the normality hypersphere.

It is noteworthy that COUTA also contributes to better detection performance when there is no overlap between these simulated anomalies and the specific anomaly type in target datasets. NAC can be seen as a self-supervised learning process, which can effectively strengthen one-class learning. Self-supervised learning defines various transformations and designs pretext tasks to classify or compare these transformations. Similarly, COUTA employs tailored anomaly-aware transformations for time series data and trains neural networks to discriminate transformed data from original data. By harnessing pretext tasks, self-supervised learning can embed data semantics into representations. Likewise, COUTA also better learns temporal patterns and inter-variant dependency within input time-series data, which poses a positive effect on the one-class learning process. Besides, we define tailored anomaly-aware transformations for time series data, i.e., NAC can achieve anomaly-informed learning that better suits the downstream anomaly detection task.

3.4 Anomaly Scoring

The learned hypersphere upon the feature space $\mathcal{F}$ can explicitly represent the data normality, and data abnormality can be simply defined as the Euclidean distance to the hypersphere center $\boldsymbol{c}$. As each distance value is extended to a Gaussian distribution to express model uncertainty in our calibrated one-class classification model, we employ two distance values to define the abnormal degree. Given the optimized network including the temporal modeling network ϕ^* and the projection heads ψ^* and ψ'^*, the anomaly score of a sub-sequence $\boldsymbol{s}$ is calculated as follows.

$$\begin{aligned}f(\boldsymbol{s}) &= d_{\boldsymbol{s}} + \tilde{d}_{\boldsymbol{s}} \\ &= \|\psi^*(\phi^*(\boldsymbol{s})) - \boldsymbol{c}\| + \|\psi'^*(\phi^*(\boldsymbol{s})) - \boldsymbol{c}\|.\end{aligned} \tag{11}$$

3.5 Discussion

This section further discusses our considerations and implementation details.

Discriminative vs. Generative. COUTA is in a discriminative manner instead of using mainstream reconstruction- or prediction-based generative models. Compared to the autoencoder structure that is composed of an encoder phase and a decoder phase, COUTA does not need to reconstruct the encoded feature back to the original shape, which is more time efficient. Moreover, COUTA learns a compact hypersphere, which is an explicit way to model data normality. That is, the optimization is directly related to anomaly detection rather than implicitly behind the rationale of data reconstruction or forecasting.

The Choice of Hypersphere Center. Arguably, including $\boldsymbol{c}$ as an optimization variable will lead to a trivial solution, i.e., all learnable parameters in network ϕ and ψ are zero [42]. Hence, following [42], we use the initialized network to perform a forward pass on all training data, and $\boldsymbol{c}$ is set as the averaged representation, i.e., $\boldsymbol{c} = \frac{1}{|\mathcal{S}|}\sum_{\boldsymbol{s}\in\mathcal{S}}\psi_0(\phi_0(\boldsymbol{s}))$, where ϕ_0 and ψ_0 are initialized neural networks before gradient optimization. It is an empirically good strategy, which makes the optimization process converge quickly and also avoids the above "hypersphere collapse" problem.

Anomaly Types in Native Anomaly Generation. We define six perturbation functions with fixed parameters in Ω. This component is also a good handle to embed readily accessible prior knowledge into the learning process. Some specific real-world applications may have their own definitions of anomalies. For example, IT operations in data centers focus on collective (pattern) anomalies and often omit point anomalies. These collective anomalies may indicate real severe faults, possible downtime of running services, and unreasonable increase of system overhead, but point anomalies are often noises induced by many possible factors of system instability. Thus, data perturbation can be designed to generate corresponding anomalies of real interests. Note that it is also a limitation of this calibration method. These data perturbation methods are designed based on general anomaly definitions. They may bring negative effects if these generated dummy anomalies are essentially normal in target systems. Nevertheless, we empirically prove that these data perturbation methods are effective in the vast majority of real-world datasets from different domains.

Anomaly Scoring Strategy. The anomaly scoring function does not use the prediction results reported by the classification head ψ_c. This branch is used to assist the optimization of the temporal modeling network ϕ by providing knowledge about anomalies. We simply treat all training data as a normal class in this branch, which means this learning task might also be misled by the anomaly contamination problem. Besides, these dummy anomaly examples might not always be reliable due to the limitation

analyzed above. Simply adding the output of this branch may downgrade the detection performance, and it is also challenging to devise a good ensemble method to integrate the outputs of two branches. Therefore, we leverage the one-class learning results calibrated by both UMC and NAC, i.e., the distance to the hypersphere center, to measure data abnormality in our anomaly scoring function.

4 Experiments

In this section, we first introduce the experimental setup, and then we conduct experiments to answer the following questions.

- **Effectiveness:** How accurate are the anomaly detection results computed by COUTA and current state-of-the-art methods on real-world datasets?
- **Generalization ability:** Can COUTA generalize to different types of time series anomalies?
- **Robustness:** How does the robustness of COUTA w.r.t. various anomaly contamination levels of the training set?
- **Scalability:** Is COUTA more time-efficient compared to existing methods?
- **Sensitivity:** How do hyper-parameters of COUTA influence the detection performance?
- **Ablation study:** Do the proposed calibration methods contribute to sufficiently better detection performance?

4.1 Experimental Setup

4.1.1 Datasets and Preprocessing Methods

Ten publicly available datasets are used in our experiments, including six multivariate datasets and four univariate datasets. Application Server Dataset (*ASD*) and Server Machine Dataset (*SMD*) are data of IT operations, in which each variate denotes the status and resource utilization of servers in a cluster. Secure Water Treatment dataset (*SWaT*) and Water Quality dataset (*WaQ*) are industrial data. The dimensions of *SWaT* are different sensors and actuators, and *WaQ* is to identify undesirable variations in the water according to a group of chemistry and physical metrics. Epilepsy seizure dataset (*Epilepsy*) and Daily and Sports Activities Dataset (*DSADS*) are motion sensor data. *Epilepsy* is to detect epilepsy seizures from three activities including walking, running, and sawing. Following [27], anomalies in *DSADS* are rare and intense activities, while the remaining activities are defined as normal data. *UCR-Fault* and *UCR-Gait* come from a biomechanics lab, recording the gait of someone walking on a force plate. Anomalies in *UCR-Fault* are faults in the left foot sensor, and anomalies in *UCR-Gait* are the gait of the individual who has Huntington's disease. *UCR-ECG-s* and *UCR-ECG-v* are ECG data, respectively containing supraventricular beats and ventricular beats. We report their statistics in Table 1, and Fig. 6 shows representative anomaly segments of each dataset.

These datasets are selected due to the following considerations: (i) These data are from various real-world applications; (ii) They are of different lengths, dimensionality, and anomaly ratios; (iii) Anomalies in these datasets can be categorized into different types; (iv) These datasets are frequently used in the literature. *ASD*, *SMD* and *SWaT* are benchmark datasets in this research line [2], [3], [6], [7], [10], [11]. *WaQ* is included in a benchmark paper [20]. *Epilepsy* and *DSADS* are datasets taken from the time series classification task by treating semantically abnormal class(es) as anomalies, as has been done in many prior anomaly detection studies [27], [36], [44], [45]. *UCR* datasets are from the latest time series anomaly archive [46].

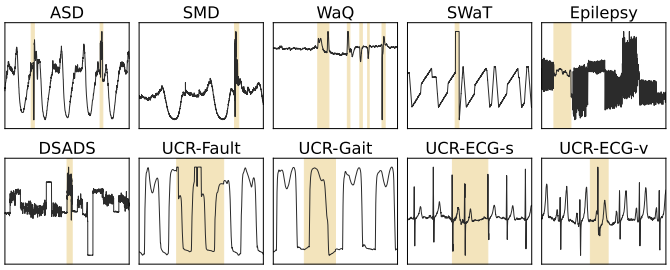


Fig. 6. Anomaly segments in ten real-world datasets.

TABLE 1
Dataset Information. N and D are length and dimensionality, #anom denotes the number of anomalies, and #entity is the number of sub-datasets in each data repository (each sub-dataset is trained and tested independently). We report the maximum and minimum values of entities in *ASD*. Each entity in *DSADS* has the same statistics. The information of each entity in *SMD* and four *UCR* datasets are respectively reported.

Dataset	N	D	#anom	#entity
ASD	11,611 - 12,960	19	55 - 441	12
SMD	57,426/57,391/57,426	38	308/198/303	3
SWAT	925,119	51	54584	1
WaQ	253,607	9	2329	1
DSADS	142,500	45	9000	8
Epilepsy	56,650	3	5768	1
UCR-Fault	64,000/64,000/64,000	1	111/651/801	3
UCR-Gait	65,000/65,000/65,000	1	301/301/601	3
UCR-ECG-s	60,000/57,001	1	371/371	2
UCR-ECG-v	40,000/40,000/80,000	1	101/101/251	3

ASD, *SMD*, *SWaT*, *WaQ*, and *UCR* datasets have predefined training-testing split. In terms of *Epilepsy* and *DSADS*, we use 60% data as the training set and the remaining 40% as the testing set while maintaining the original anomaly proportion. Following [47], the minimum and maximum values per dimension of the training set are obtained to conduct min-max normalization, and the data values in the testing set are clipped to $[\min -4, \max +4]$ to prevent excessively large values skewing anomaly scoring process. Deep anomaly detectors generally use a sliding window to divide time series datasets into small sub-sequences. A sliding window with a fixed size of 100 is used for *ASD*, *SMD*, *SWaT*, *WaQ*, and four *UCR* datasets. We take 1 as the sliding stride for *ASD*, *SMD* and respectively use 100 and 5 for large-scale datasets *SWaT* and *WaQ*. As for the remaining two datasets, *Epilepsy* and *DSADS*, the original data format is collections of divided time sequences, and thus sliding window is not required.

4.1.2 Competing Methods

COUTA is compared with sixteen anomaly detection methods including ARMA [48], OCSVM [49], GOAD [50], ECOD [51], DAGMM [52], LSTM-ED [25], LSTM-Pr [8], Tcn-ED [47], DSVDD [42], MSCRED [26], Omni [10], USAD [4],

GDN [2], NeuTraL [36], TranAD [3], and AnomTran [5]. These competing methods include both traditional and deep approaches. Also, these competing methods employ different learning strategies (prediction, reconstruction, and discriminative one-class/self-supervised learning) and various network structures (MLP, LSTM, GRU, TCN, Transformer, convolutional net, and graph neural network). The above competitor list can well represent the state-of-the-art performance of the research line of time series anomaly detection.

4.1.3 Evaluation Metrics and Computing Infrastructure

Many anomalies in time series data are consecutive, and they can be viewed as multiple anomaly segments. In many practical applications, an anomaly segment can be properly handled if detection models can trigger an alert at any timestamp within this segment. Therefore, the vast majority of previous studies [2], [3], [4], [5], [7], [10], [11] employ *point-adjust* strategy in their experiment protocols. Specifically, the scores of observations in each anomaly segment are adjusted to the highest value within this segment, which simulates the above assumption (a single alert is sufficient to take action). Other points outside anomaly segments are treated as usual. For the sake of consistency with the current literature, we also employ this adjustment strategy before computing evaluation metrics.

Precision P and recall R of the anomaly class can directly indicate the costs and benefits of finding anomalies in real-world applications, which can intuitively reflect model performance. F_1 score is the harmonic mean of precision and recall, which takes both precision and recall into account. These detection models output continuous anomaly scores, but there is often no specific way to determine a decision threshold when calculating precision and recall. Therefore, following prior work in this research line [3], [4], [6], [7], [10], [11], we use the best F_1 score and the Area Under the Precision-Recall Curve (AUC-PR), considering simplicity and fairness. These two metrics can avoid possible bias brought by fixed thresholds or threshold calculation methods. The best F_1 score represents the optimal case, while AUC-PR is in an average case that is less optimal. Specifically, we enumerate all possible thresholds (i.e., scores of each timestamp) and compute corresponding precision and recall scores. The best F_1 can then be obtained, and AUC-PR is computed by employing the average precision score. In the following experiments, F_1, P, and R denote the best F_1 and its corresponding precision and recall score. The above performance evaluation metrics range from 0 to 1, and a higher value indicates better performance.

All the experiments are executed at a workstation with an Intel(R) Xeon(R) Silver 4210R CPU @ 2.4GHz, an NVIDIA TITAN RTX GPU, and 64 GB RAM.

4.1.4 Parameter Settings and Implementations

In COUTA, the temporal modeling network ϕ is with one hidden layer, and the kernel size uses 2. The projection head ψ and the classification head ψ_c are two-layer multi-layer perceptron networks with LeakyReLU activation. The hidden layer of ϕ and ψ has 16 neural units by default, and the dimension of the feature space $\mathcal{F}$ is also 16. We respectively use 32 and 64 in complicated datasets *ASD* and *DSADS* to enhance the representation power of the neural network. Adam optimizer is employed, where the learning rate is set to 10^{-4}. The weight factor α of the supervised classification branch in the loss function uses 0.1 by default. The factor β that controls the size of generated anomaly examples is set to 0.2. As for the competing methods, we use the default or recommended hyper-parameter settings in their respective original papers.

These anomaly detectors are implemented in Python. We use the implementations of OCSVM and ECOD from `pyod`, a python library of anomaly detection approaches. The source code of GOAD, DSVDD, USAD, GDN, NeuTraL, and TranAD are released by their original authors. In terms of LSTM-ED, Tcn-ED, MSCRED, and Omni, we use the implementations from an evaluation study [47]. The source code of COUTA is available at https://github.com/xuhongzuo/couta.

4.2 Effectiveness in Real-World Datasets

This experiment evaluates the effectiveness of COUTA. We perform COUTA and its competing methods on ten real-world time series datasets. These models are trained on training sets, and we report their detection performance on testing sets. Ground-truth labels in testing sets are strictly unknown in the training stage.

Table 2 and Table 3 respectively illustrate the F_1 score and AUC-PR of anomaly detectors on six multivariate datasets and four univariate datasets. ARMA cannot obtain results in three days on SMD, SWaT, and DSADS. MSCRED runs out of memory on the large-scale dataset *SWaT* due to the high computational cost of the deep convolutional network structure used in MSCRED. It is noteworthy that OCSVM, GOAD, ECOD, and DAGMM are not originally designed for time series data. MSCRED and GDN learn interactions between different varieties, i.e., they cannot perform single-dimensional data. Thus, we report COUTA and the remaining nine competitors in Table 3. COUTA obtains the best F_1 performance on all datasets. In terms of AUC-PR, COUTA outperforms all of its state-of-the-art competing methods on nine datasets with a 0.015 disparity to the best performer on the remaining dataset. Averagely, COUTA achieves 11% - 49% F_1 improvement and 11% - 68 % AUC-PR enhancement over various competitors.

According to the comparison results, COUTA successfully achieves state-of-the-art performance by addressing two key limitations in the current one-class learning pipeline. The superiority of COUTA can be attributed to the synergy of our two novel one-class calibration components, which achieves contamination-tolerant, anomaly-informed learning of data normality. The canonical one-class learning process used in these competing methods suffers from the anomaly contamination problem, and they may also learn an inaccurate range of normal behaviors due to the lack of guidance about the anomaly class. It is noteworthy that, on dataset *SWaT*, a pure training set with only normal observations is ensured (water treatment attack is only launched in the testing set), and thus competing methods can obtain relatively good performance. AnomTran, TranAD, and GDN achieve better performance compared to other competitors. AnomTran and TranAD employ the advanced Transformer structure and several training tricks, and GDN models inter-variate correlations with the help of the powerful capability

TABLE 2
F_1 score and AUC-PR ± standard deviation of COUTA and its competitors on six multivariate datasets.

Methods	F_1						AUC-PR					
	ASD	SMD	SWaT	WaQ	DSADS	Epilepsy	ASD	SMD	SWaT	WaQ	DSADS	Epilepsy
ARMA	$0.474_{\pm 0.000}$	N/A	N/A	$0.647_{\pm 0.000}$	N/A	$0.826_{\pm 0.000}$	$0.395_{\pm 0.000}$	N/A	N/A	$0.633_{\pm 0.000}$	N/A	$0.801_{\pm 0.000}$
OCSVM	$0.625_{\pm 0.000}$	$0.761_{\pm 0.000}$	$0.839_{\pm 0.000}$	$0.732_{\pm 0.000}$	$0.807_{\pm 0.000}$	$0.781_{\pm 0.000}$	$0.540_{\pm 0.000}$	$0.664_{\pm 0.000}$	$0.804_{\pm 0.000}$	$0.666_{\pm 0.000}$	$0.753_{\pm 0.000}$	$0.708_{\pm 0.000}$
GOAD	$0.827_{\pm 0.025}$	$0.975_{\pm 0.019}$	$0.831_{\pm 0.003}$	$0.739_{\pm 0.052}$	$0.723_{\pm 0.014}$	$0.554_{\pm 0.147}$	$0.834_{\pm 0.023}$	$0.981_{\pm 0.013}$	$0.799_{\pm 0.005}$	$0.685_{\pm 0.034}$	$0.676_{\pm 0.015}$	$0.530_{\pm 0.128}$
ECOD	$0.589_{\pm 0.000}$	$0.755_{\pm 0.000}$	$0.849_{\pm 0.000}$	$0.676_{\pm 0.000}$	$0.923_{\pm 0.000}$	$0.785_{\pm 0.000}$	$0.527_{\pm 0.000}$	$0.724_{\pm 0.000}$	$0.899_{\pm 0.000}$	$0.716_{\pm 0.000}$	$0.941_{\pm 0.000}$	$0.763_{\pm 0.000}$
DAGMM	$0.743_{\pm 0.105}$	$0.652_{\pm 0.168}$	$0.832_{\pm 0.010}$	$0.592_{\pm 0.092}$	$0.775_{\pm 0.067}$	$0.793_{\pm 0.024}$	$0.708_{\pm 0.117}$	$0.581_{\pm 0.221}$	$0.806_{\pm 0.011}$	$0.539_{\pm 0.101}$	$0.764_{\pm 0.091}$	$0.774_{\pm 0.040}$
LSTM-Pr	$0.593_{\pm 0.011}$	$0.860_{\pm 0.044}$	$0.834_{\pm 0.055}$	$0.532_{\pm 0.035}$	$0.856_{\pm 0.011}$	$0.661_{\pm 0.006}$	$0.513_{\pm 0.013}$	$0.858_{\pm 0.051}$	$0.823_{\pm 0.101}$	$0.463_{\pm 0.049}$	$0.934_{\pm 0.010}$	$0.513_{\pm 0.011}$
LSTM-ED	$0.807_{\pm 0.013}$	$0.960_{\pm 0.000}$	$0.847_{\pm 0.007}$	$0.759_{\pm 0.006}$	$0.865_{\pm 0.011}$	$0.802_{\pm 0.012}$	$0.767_{\pm 0.016}$	$0.955_{\pm 0.009}$	$0.848_{\pm 0.005}$	$0.714_{\pm 0.018}$	$0.902_{\pm 0.005}$	$0.784_{\pm 0.012}$
Tcn-ED	$0.853_{\pm 0.015}$	$0.848_{\pm 0.050}$	$0.843_{\pm 0.011}$	$0.707_{\pm 0.082}$	$0.850_{\pm 0.021}$	$0.758_{\pm 0.011}$	$0.862_{\pm 0.019}$	$0.881_{\pm 0.036}$	$0.846_{\pm 0.003}$	$0.658_{\pm 0.090}$	$0.868_{\pm 0.023}$	$0.763_{\pm 0.008}$
DSVDD	$0.691_{\pm 0.014}$	$0.682_{\pm 0.012}$	$0.829_{\pm 0.002}$	$0.519_{\pm 0.038}$	$0.751_{\pm 0.057}$	$0.686_{\pm 0.040}$	$0.671_{\pm 0.017}$	$0.621_{\pm 0.033}$	$0.811_{\pm 0.003}$	$0.410_{\pm 0.043}$	$0.690_{\pm 0.078}$	$0.555_{\pm 0.069}$
MSCRED	$0.766_{\pm 0.036}$	$0.628_{\pm 0.031}$	N/A	$0.717_{\pm 0.007}$	$0.657_{\pm 0.029}$	$0.640_{\pm 0.017}$	$0.756_{\pm 0.050}$	$0.536_{\pm 0.049}$	N/A	$0.644_{\pm 0.016}$	$0.659_{\pm 0.062}$	$0.604_{\pm 0.023}$
Omni	$0.810_{\pm 0.044}$	$0.954_{\pm 0.006}$	$0.845_{\pm 0.012}$	$0.738_{\pm 0.018}$	$0.867_{\pm 0.018}$	$0.811_{\pm 0.027}$	$0.789_{\pm 0.063}$	$0.928_{\pm 0.011}$	$0.841_{\pm 0.008}$	$0.714_{\pm 0.023}$	$0.914_{\pm 0.015}$	$0.780_{\pm 0.054}$
USAD	$0.595_{\pm 0.033}$	$0.744_{\pm 0.006}$	$0.835_{\pm 0.000}$	$0.666_{\pm 0.049}$	$0.733_{\pm 0.041}$	$0.663_{\pm 0.009}$	$0.510_{\pm 0.035}$	$0.658_{\pm 0.006}$	$0.808_{\pm 0.000}$	$0.611_{\pm 0.044}$	$0.713_{\pm 0.065}$	$0.541_{\pm 0.038}$
GDN	$0.801_{\pm 0.034}$	$0.939_{\pm 0.015}$	$0.846_{\pm 0.025}$	$0.640_{\pm 0.052}$	$0.795_{\pm 0.022}$	$0.783_{\pm 0.010}$	$0.779_{\pm 0.042}$	$0.959_{\pm 0.016}$	$0.885_{\pm 0.036}$	$0.659_{\pm 0.043}$	$0.728_{\pm 0.024}$	$0.789_{\pm 0.010}$
NeuTraL	$0.627_{\pm 0.047}$	$0.770_{\pm 0.050}$	$0.862_{\pm 0.023}$	$0.681_{\pm 0.085}$	$0.534_{\pm 0.035}$	$0.739_{\pm 0.075}$	$0.592_{\pm 0.045}$	$0.746_{\pm 0.067}$	$0.850_{\pm 0.026}$	$0.616_{\pm 0.111}$	$0.490_{\pm 0.045}$	$0.729_{\pm 0.111}$
TranAD	$0.899_{\pm 0.020}$	$0.761_{\pm 0.001}$	$0.831_{\pm 0.009}$	$0.755_{\pm 0.010}$	$0.730_{\pm 0.114}$	$0.776_{\pm 0.008}$	$0.915_{\pm 0.018}$	$0.662_{\pm 0.003}$	$0.810_{\pm 0.007}$	$\mathbf{0.729}_{\pm 0.017}$	$0.690_{\pm 0.158}$	$0.734_{\pm 0.012}$
AnomTran	$0.679_{\pm 0.134}$	$0.657_{\pm 0.097}$	$0.845_{\pm 0.006}$	$0.704_{\pm 0.025}$	$0.836_{\pm 0.067}$	$0.817_{\pm 0.012}$	$0.627_{\pm 0.166}$	$0.608_{\pm 0.139}$	$0.822_{\pm 0.010}$	$0.639_{\pm 0.019}$	$0.868_{\pm 0.066}$	$0.814_{\pm 0.029}$
COUTA	$\mathbf{0.942}_{\pm 0.031}$	$\mathbf{0.980}_{\pm 0.018}$	$\mathbf{0.886}_{\pm 0.022}$	$\mathbf{0.781}_{\pm 0.013}$	$\mathbf{0.926}_{\pm 0.029}$	$\mathbf{0.830}_{\pm 0.053}$	$\mathbf{0.955}_{\pm 0.030}$	$\mathbf{0.984}_{\pm 0.015}$	$\mathbf{0.900}_{\pm 0.017}$	$0.714_{\pm 0.006}$	$\mathbf{0.942}_{\pm 0.020}$	$\mathbf{0.823}_{\pm 0.086}$

TABLE 3
F_1 score and AUC-PR on four univariate datasets.

	Methods	UCR-Fault	UCR-Gait	UCR-ECG-s	UCR-ECG-v
F_1	ARMA	$\mathbf{1.000}_{\pm 0.000}$	$\mathbf{0.992}_{\pm 0.000}$	$0.906_{\pm 0.000}$	$0.824_{\pm 0.000}$
	LSTM-Pr	$\mathbf{1.000}_{\pm 0.000}$	$0.865_{\pm 0.052}$	$0.917_{\pm 0.014}$	$0.919_{\pm 0.099}$
	LSTM-ED	$0.685_{\pm 0.103}$	$0.815_{\pm 0.036}$	$0.760_{\pm 0.024}$	$0.723_{\pm 0.062}$
	Tcn-ED	$0.994_{\pm 0.010}$	$0.912_{\pm 0.161}$	$0.867_{\pm 0.059}$	$0.679_{\pm 0.228}$
	DSVDD	$0.846_{\pm 0.028}$	$0.869_{\pm 0.011}$	$0.850_{\pm 0.057}$	$0.783_{\pm 0.074}$
	Omni	$0.316_{\pm 0.094}$	$0.426_{\pm 0.138}$	$0.759_{\pm 0.013}$	$0.828_{\pm 0.068}$
	USAD	$0.519_{\pm 0.072}$	$0.567_{\pm 0.103}$	$0.359_{\pm 0.014}$	$0.510_{\pm 0.092}$
	NeuTraL	$1.000_{\pm 0.000}$	$0.804_{\pm 0.032}$	$0.711_{\pm 0.028}$	$0.790_{\pm 0.071}$
	TranAD	$0.877_{\pm 0.117}$	$0.916_{\pm 0.040}$	$0.779_{\pm 0.029}$	$0.814_{\pm 0.001}$
	AnomTran	$0.786_{\pm 0.212}$	$0.781_{\pm 0.181}$	$0.863_{\pm 0.094}$	$0.827_{\pm 0.187}$
	COUTA	$\mathbf{1.000}_{\pm 0.000}$	$0.976_{\pm 0.034}$	$\mathbf{0.924}_{\pm 0.038}$	$\mathbf{0.968}_{\pm 0.014}$
AUC-PR	ARMA	$\mathbf{1.000}_{\pm 0.000}$	$\mathbf{0.984}_{\pm 0.000}$	$0.835_{\pm 0.000}$	$0.752_{\pm 0.000}$
	LSTM-Pr	$\mathbf{1.000}_{\pm 0.000}$	$0.777_{\pm 0.072}$	$0.855_{\pm 0.021}$	$0.899_{\pm 0.123}$
	LSTM-ED	$0.559_{\pm 0.114}$	$0.698_{\pm 0.046}$	$0.646_{\pm 0.035}$	$0.627_{\pm 0.077}$
	Tcn-ED	$0.989_{\pm 0.018}$	$0.886_{\pm 0.197}$	$0.780_{\pm 0.084}$	$0.566_{\pm 0.289}$
	DSVDD	$0.791_{\pm 0.028}$	$0.784_{\pm 0.015}$	$0.763_{\pm 0.074}$	$0.702_{\pm 0.081}$
	Omni	$0.216_{\pm 0.078}$	$0.313_{\pm 0.127}$	$0.636_{\pm 0.019}$	$0.769_{\pm 0.081}$
	USAD	$0.460_{\pm 0.068}$	$0.463_{\pm 0.094}$	$0.266_{\pm 0.015}$	$0.389_{\pm 0.077}$
	NeuTraL	$1.000_{\pm 0.000}$	$0.717_{\pm 0.037}$	$0.622_{\pm 0.030}$	$0.701_{\pm 0.099}$
	TranAD	$0.801_{\pm 0.172}$	$0.849_{\pm 0.068}$	$0.689_{\pm 0.030}$	$0.758_{\pm 0.001}$
	AnomTran	$0.757_{\pm 0.249}$	$0.703_{\pm 0.211}$	$0.779_{\pm 0.141}$	$0.758_{\pm 0.224}$
	COUTA	$\mathbf{1.000}_{\pm 0.000}$	$0.956_{\pm 0.060}$	$\mathbf{0.871}_{\pm 0.060}$	$\mathbf{0.941}_{\pm 0.025}$

of graph neural networks in exploring high-order interactions among graph nodes. However, TranAD may suffer from the overfitting problem because its learning process contains several complicated components, and thus it fails to produce effective detection results on simple datasets like *SMD* and *SWaT*. In contrast, detectors with plain encoder-decoder structures (e.g., LSTM-ED) obtain better performance.

4.3 Generalization Ability to Different Types of Time Series Anomalies

We investigate whether COUTA can handle different anomaly types in time series data. Following a fine-grained taxonomy of time series anomalies in [20], three synthetic datasets are created. These datasets contain 1000 observations, which are described in two dimensions. The first 400 points are used for training, and the remaining data are treated as testing sets. We demonstrate the testing data of three cases in Fig. 7, where (a), (b), and (c) contain point-wise anomalies, pattern-wise anomalies, and pattern-wise anomalies with different lengths, respectively. Point-wise anomalies include both global and contextual forms, and

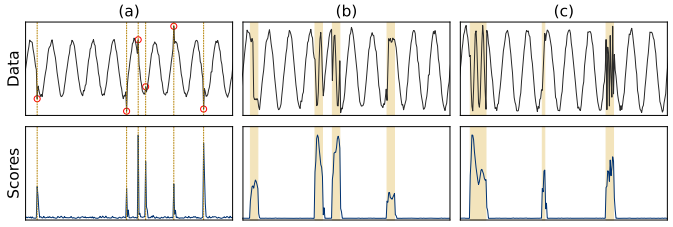


Fig. 7. Performance of COUTA in handling (a) point-wise (global and contextual) anomalies, (b) pattern-wise (seasonal and shapelet) anomalies, and (c) anomalies with different lengths.

pattern-wise anomalies are caused by basic shapelet and seasonality changes.

The bottom panel of Fig. 7 shows the anomaly scores produced by our method COUTA. COUTA successfully identifies all of these anomaly cases with distinguishably higher scores on true anomalies and consistently lower scores on normal moments. We use three pre-defined fixed operations in creating native anomaly examples. These native anomaly examples are used to calibrate the normality boundary. COUTA can still generalize to different anomaly types even if some anomaly types may not strictly correspond to the generated native anomalies. It is mainly because COUTA is essentially a one-class classification model, and by definition, it can alarm all the observations that deviate from the learned normality according to one-class distances. The data normality is modeled based on the majority of the training data instead of these created native anomalies. Therefore, the generalization ability of our method COUTA is not limited by the pre-defined operations in NAC, i.e., COUTA can also identify novel anomalies that are not covered in the NAC process.

4.4 Robustness w.r.t. Anomaly Contamination

This experiment is to quantitatively measure the interference of anomaly contamination to time series anomaly detectors, that is, we test the robustness of each anomaly detector w.r.t. different anomaly contamination ratios in the training set. Due to the continuity of time series data, we cannot directly remove or inject anomalies like other robustness testing experiments in prior work on tabular data [45],

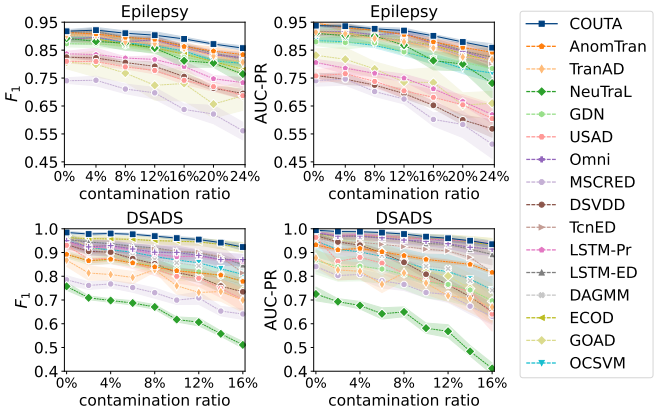


Fig. 8. Robustness w.r.t. anomaly contamination.

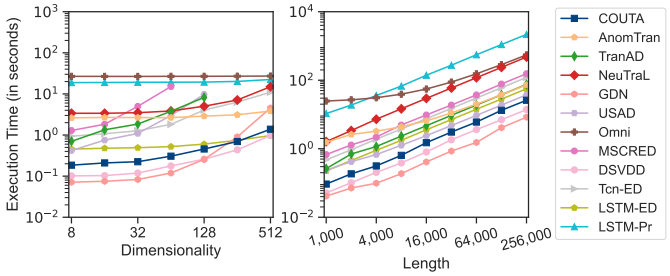


Fig. 9. Scalability test results w.r.t. dimensionality and length of time series. TranAD, USAD, and MSCRED run out of memory when handling time series data with high dimensionality.

[53]. It is hard to precisely adjust ratios of abnormal timestamps in the training set. Therefore, we employ *Epilepsy* and *DSADS* in this experiment because these datasets are collections of divided small sequences with sequence-level labels. We treat small sequences as data objects and generate a group of datasets with different anomaly contamination levels of the training set. We first randomly remove anomaly sequences in the training set to meet the requirements of specific contamination ratios. The removed anomalies are then added to the testing set. The original sequence-level anomaly contamination rate in *Epilepsy* and *DSADS* are 24% and 16%, respectively. A wide range of contamination levels is used for each dataset by starting from a pure training set and taking 4% as the increasing step.

Fig. 8 presents the F_1 score and the AUC-PR performance of COUTA and its competing methods on datasets with different anomaly contamination ratios. The performance of all anomaly detectors downgrades with the increase of anomaly contamination. COUTA shows better robustness compared to its contenders, especially on datasets with a large contamination rate. It owes to the novel one-class classification loss function, which successfully masks these noisy data via uncertainty modeling-based adaptive penalty. By contrast, these competing methods simply view these hidden anomalies as normal data, and the learned normality model may mistakenly overfit these noises. This experiment validates the contribution of the proposed uncertainty modeling-based calibration method. This experiment also shows superior applicability of COUTA in some real-world applications that are with complicated noisy circumstances.

4.5 Scalability Test

This experiment investigates the scalability of COUTA compared to its competing methods. Time efficiency w.r.t. both time series length T and dimensionality D are recorded. As for the scalability test w.r.t. dimensionality, a group of seven time-series datasets with a fixed length (i.e., 2,000) and various dimensions (i.e., from 8 to 512 with 2 as the magnification factor) is generated. We synthesize another group of nine datasets containing different lengths (i.e., range from 1,000 to 256,000) and a fixed dimension (i.e., 8) for the scale-up test w.r.t. length. As these deep anomaly detectors are deployed with different numbers of training epochs, we report the execution time of one training epoch by taking 128 as a unified size of training mini-batches.

Fig. 9 presents the execution time of COUTA and its ten competing state-of-the-art methods on time series datasets with various sizes. Note that this experiment excludes three general anomaly detectors (i.e., OCSVM, ECOD, and GOAD) that are not originally designed for time series data. COUTA has good scalability compared to most of these existing methods, which shows its potential capability of being applied in practical scenarios where time series data are with large volumes and high dimensions.

4.6 Sensitivity Test

This experiment shows the impact of different parameter settings on detection performance. First, we only use a single kind of perturbation (δ_{I}, δ_{II}, or δ_{III}) to solely generate point, contextual, or collective anomalies in NAC. Besides, we investigate several key hyper-parameters of COUTA including α, β, H, and l. α is the weight of loss $\mathcal{L}_{\text{NAC}}$ in Equation (1), β is the ratio that controls the size of generated anomaly examples, H is the dimensionality of the feature space $\mathcal{F}$, and l is the sliding window length. Each parameter is sampled from a wide range.

Fig. 10 shows the F_1 performance of COUTA by taking different parameter settings, and AUC-PR performance also shows the same trend, which is omitted here. COUTA shows better performance when a full perturbation operation pool Ω is employed in NAC (especially on the *ASD* dataset). A single type of generated anomaly example may provide fragmentary knowledge about the anomaly class, thus leading to less optimal performance. In terms of α, β, and l, these parameters do not largely influence the performance, and COUTA performs stably with different hyper-parameters. Some parameter selection methods can be employed. There might be some unreliable generated anomaly examples that fall into the normal distribution, and thus we use $\alpha = 0.1$, $\beta = 0.2$ by default. In terms of l, we employ a frequently-used sliding window length (i.e., $l = 100$). COUTA shows fluctuation trends w.r.t. the dimensionality of the feature space H. It might be because these time series datasets are with various numbers of dimensions. Generally, a larger H might be preferable when handling high-dimensional time series datasets (e.g., *DSADS* with 47 dimensions) because the representation capability can be ensured, while low-

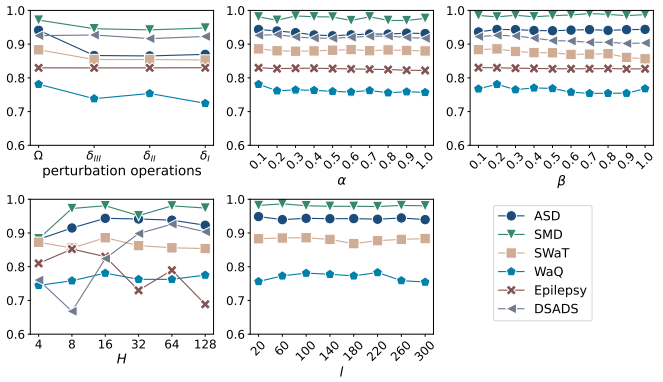


Fig. 10. Sensitivity test results (F_1 performance w.r.t. different parameter settings). *Epilepsy* and *DSADS* have default lengths of sub-sequences, and thus the detection performance w.r.t. l on these two datasets is omitted.

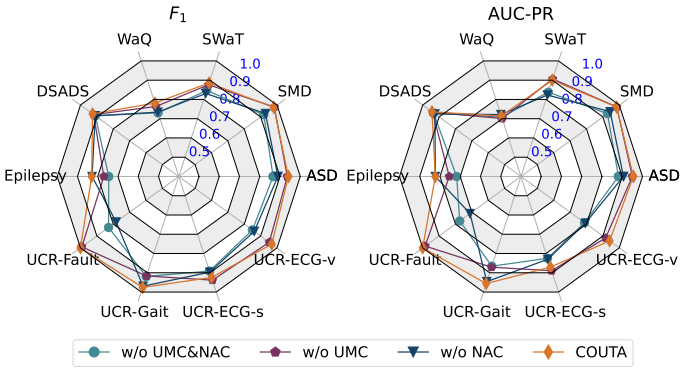


Fig. 11. Ablation study results.

dimensional datasets can be well processed by using smaller H (e.g., *Epilepsy* with 5 dimensions).

4.7 Ablation Study

This experiment further validates the contribution of two key components in COUTA. Three ablated variants are used. Two calibration methods, i.e., uncertainty modeling-based calibration and native anomaly-based calibration, are respectively removed from COUTA in two ablated variants, **w/o** UMC and **w/o** NAC. These two components are simultaneously excluded in **w/o** UMC&NAC. The remaining parts of these ablated versions remain the same as COUTA. We report the F_1 and AUC-PR performance of standard COUTA and its three ablated versions in Fig. 11. Based on the comparison of COUTA and its variants, the superiority of COUTA can verify the significant contribution of two calibration methods on one-class classification. COUTA outperforms **w/o** UMC, **w/o**, and **w/o** UMC&NAC by 8%, 2%, and 7%, respectively. Particularly, UMC conduces to 11% improvement in F_1 score and 14% gain in AUC-PR on the *Epilepsy* dataset, where the training set is severely contaminated by unknown anomalies, and NAC can bring approximate 8 % enhancement in both F_1 and AUC-PR on the *ASD* dataset.

5 Conclusions

This paper introduces COUTA, an unsupervised time series anomaly detection method based on calibrated one-class classification. We address two key challenges in the current one-class learning pipeline, i.e., the presence of anomaly contamination and the absence of knowledge about anomalies. COUTA achieves this goal through two novel calibration methods – uncertainty modeling-based calibration (UMC) and native anomaly-based calibration (NAC). In UMC, we obtain model uncertainty by imposing a prior distribution to the one-class distance value, and a theoretically motivated novel learning objective is devised to restrain noisy data that are with high uncertainty, while simultaneously encouraging confident predictions to ensure effective learning of hard normal samples. In NAC, we design tailored data perturbation operations to produce native anomaly examples based on original time series data, which provides one-class classification with valuable knowledge about primeval anomalous behaviors. These calibration methods enable COUTA to learn data normality in a noise-tolerant, anomaly-informed manner. Extensive experiments show that COUTA achieves state-of-the-art performance in time series anomaly detection by substantially outperforming sixteen competitors. We also validate several desired properties of COUTA, including outstanding generalization ability to different anomaly types, superior robustness to anomaly contamination, and good scalability.

Similar to many one-class classification methods, our approach also assumes that normal data share similarities and belong to one prototype. Our approach is adept at handling time series data that are seasonal and have many repetitive patterns, whereas our approach may fail when there are concept drifts happening in time series.

In the future, we plan to investigate new strategies to estimate the mean and variance of the prior one-class distance distribution. Also, in producing native anomalies, tailored perturbation methods can be extended to a heuristic generation process.

Acknowledgments

The work of Hongzuo Xu, Yijie Wang, Songlei Jian, Qing Liao, and Yongjun Wang were supported in part by the National Key R&D Program of China under Grant 2022ZD0115302, in part by the National Natural Science Foundation of China under Grant 62002371, 62076079, U19A2067, 61379052, and 61472439, in part by the Science Foundation of Ministry of Education of China under Grant 2018A02002, in part by the Natural Science Foundation for Distinguished Young Scholars of Hunan Province under Grant 14JJ1026, in part by the Foundation of National University of Defense Technology under Grant ZK21-17. The work of Guansong Pang was supported in part by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 under Grant 21SISSMU031.

We thank Prof. Eamonn Keogh and his students for creating the UCR time series anomaly archive that has been used in our experiment. We also thank the referees for their comments, which helped improve this paper considerably.

REFERENCES

[1] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, 2021.

[2] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 5, 2021, pp. 4027–4035.

[3] S. Tuli, G. Casale, and N. R. Jennings, "TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data," *Proc. VLDB Endowment*, vol. 15, no. 6, pp. 1201–1214, 2022.

[4] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "Usad: unsupervised anomaly detection on multivariate time series," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2020, pp. 3395–3404.

[5] J. Xu, H. Wu, J. Wang, and M. Long, "Anomaly transformer: Time series anomaly detection with association discrepancy," in *Proc. Int. Conf. Learn. Represent.*, 2022.

[6] D. Campos, T. Kieu, C. Guo, F. Huang, K. Zheng, B. Yang, and C. S. Jensen, "Unsupervised time series outlier detection with diversity-driven convolutional ensembles," *Proc. VLDB Endowment*, vol. 15, no. 3, pp. 611–623, 2021.

[7] Z. Li, Y. Zhao, J. Han, Y. Su, R. Jiao, X. Wen, and D. Pei, "Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2021, pp. 3220–3230.

[8] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 387–395.

[9] H. Zhao, Y. Wang, J. Duan, C. Huang, D. Cao, Y. Tong, B. Xu, J. Bai, J. Tong, and Q. Zhang, "Multivariate time-series anomaly detection via graph attention network," in *Proc. 20th IEEE Int. Conf. Data Mining*. IEEE, 2020, pp. 841–850.

[10] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 2828–2837.

[11] S. Liu, B. Zhou, Q. Ding, B. Hooi, Z. bo Zhang, H. Shen, and X. Cheng, "Time series anomaly detection with adversarial reconstruction networks," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 4293–4306, 2023.

[12] T. Kieu, B. Yang, C. Guo, C. S. Jensen, Y. Zhao, F. Huang, and K. Zheng, "Robust and explainable autoencoders for unsupervised time series outlier detection—extended version," *arXiv preprint arXiv:2204.03341*, 2022.

[13] T. Kieu, B. Yang, C. Guo, and C. S. Jensen, "Outlier detection for time series with recurrent autoencoder ensembles," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 2725–2732.

[14] B. Du, X. Sun, J. Ye, K. Cheng, J. Wang, and L. Sun, "Gan-based anomaly detection for multivariate time series using polluted training set," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12 208–12 219, 2023.

[15] G. Pang, L. Cao, L. Chen, and H. Liu, "Learning representations of ultrahigh-dimensional data for random distance-based outlier detection," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 2041–2050.

[16] G. Pang, C. Yan, C. Shen, A. v. d. Hengel, and X. Bai, "Self-trained deep ordinal regression for end-to-end video anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 12 173–12 182.

[17] I. Golan and R. El-Yaniv, "Deep anomaly detection using geometric transformations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9781–9791.

[18] C.-L. Li, K. Sohn, J. Yoon, and T. Pfister, "Cutpaste: Self-supervised learning for anomaly detection and localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 9664–9674.

[19] S. Deldari, D. V. Smith, H. Xue, and F. D. Salim, "Time series change point detection with self-supervised contrastive predictive coding," in *Proc. Web Conf.*, 2021, pp. 3124–3135.

[20] K.-H. Lai, D. Zha, J. Xu, Y. Zhao, G. Wang, and X. Hu, "Revisiting time series outlier detection: Definitions and benchmarks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021.

[21] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2250–2267, 2013.

[22] J. Paparrizos, Y. Kang, P. Boniol, R. S. Tsay, T. Palpanas, and M. J. Franklin, "Tsb-uad: an end-to-end benchmark suite for univariate time-series anomaly detection," *Proc. VLDB Endowment*, vol. 15, no. 8, pp. 1697–1711, 2022.

[23] S. Schmidl, P. Wenig, and T. Papenbrock, "Anomaly detection in time series: A comprehensive evaluation," *Proc. VLDB Endowment*, vol. 15, no. 9, pp. 1779–1797, 2022.

[24] V. Jacob, F. Song, A. Stiegler, B. Rad, Y. Diao, and N. Tatbul, "Exathlon: a benchmark for explainable anomaly detection over time series," *Proc. VLDB Endowment*, vol. 14, no. 11, pp. 2613–2626, 2021.

[25] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Lstm-based encoder-decoder for multi-sensor anomaly detection," *arXiv preprint arXiv:1607.00148*, 2016.

[26] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, and N. V. Chawla, "A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 01, 2019, pp. 1409–1416.

[27] Y. Zhang, J. Wang, Y. Chen, H. Yu, and T. Qin, "Adaptive memory networks with self-supervised learning for unsupervised anomaly detection," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12 068–12 080, 2023.

[28] D. Huang, L. Shen, Z. Yu, Z. Zheng, M. Huang, and Q. Ma, "Efficient time series anomaly detection by multiresolution self-supervised discriminative network," *Neurocomputing*, vol. 491, pp. 261–272, 2022.

[29] L. Shen, Z. Li, and J. Kwok, "Timeseries anomaly detection using temporal hierarchical one-class network," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 13 016–13 026, 2020.

[30] Y. Zhao, X. Chen, L. Deng, T. Kieu, C. Guo, B. Yang, K. Zheng, and C. S. Jensen, "Outlier detection for streaming task assignment in crowdsourcing," in *Proc. Web Conf.*, 2022, pp. 1933–1943.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[32] C. Feng and P. Tian, "Time series anomaly detection for cyber-physical systems via neural system identification and bayesian filtering," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2021, pp. 2858–2867.

[33] C. Qiu, A. Li, M. Kloft, M. Rudolph, and S. Mandt, "Latent outlier exposure for anomaly detection with contaminated data," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 18 153–18 167.

[34] B. Han, Q. Yao, T. Liu, G. Niu, I. W. Tsang, J. T. Kwok, and M. Sugiyama, "A survey of label-noise representation learning: Past, present and future," *arXiv preprint arXiv:2011.04406*, 2020.

[35] H. Chai, W. Su, S. Tang, Y. Ding, B. Fang, and Q. Liao, "Improving anomaly detection with a self-supervised task based on generative adversarial network," in *Proc. Int. Conf. Acoust., Speech, Signal Process.* IEEE, 2022, pp. 3563–3567.

[36] C. Qiu, T. Pfrommer, M. Kloft, S. Mandt, and M. Rudolph, "Neural transformation learning for deep anomaly detection beyond images," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8703–8714.

[37] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[38] J. Cai and J. Fan, "Perturbation learning based anomaly detection," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 14 317–14 330.

[39] S. Goyal, A. Raghunathan, M. Jain, H. V. Simhadri, and P. Jain, "Drocc: Deep robust one-class classification," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 3711–3721.

[40] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

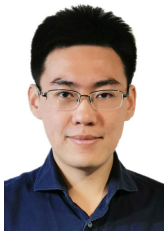
[41] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[42] L. Ruff, R. A. Vandermeulen, N. Görnitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep One-Class Classification," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4393–4402.

[43] M. Kemmler, E. Rodner, E.-S. Wacker, and J. Denzler, "One-class classification with gaussian processes," *Pattern recognition*, vol. 46, no. 12, pp. 3507–3518, 2013.

[44] H. Xu, Y. Wang, J. Wei, S. Jian, Y. Li, and N. Liu, "Fascinating supervisory signals and where to find them: deep anomaly detection with scale learning," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 38 655–38 673.

[45] H. Xu, Y. Wang, G. Pang, S. Jian, N. Liu, and Y. Wang, "Rosas: Deep semi-supervised anomaly detection with contamination-resilient continuous supervision," *Inf. Process. & Manag.*, vol. 60, no. 5, p. 103459, 2023.
[46] R. Wu and E. Keogh, "Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 3, pp. 2421–2429, 2023.
[47] A. Garg, W. Zhang, J. Samaran, R. Savitha, and C.-S. Foo, "An evaluation of anomaly detection and diagnosis in multivariate time series," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2508–2517, 2022.
[48] P. J. Brockwell and R. A. Davis, *Time series: theory and methods*. Springer Science & Business Media, 1991.
[49] L. M. Manevitz and M. Yousef, "One-class svms for document classification," *J. Mach. Learn. Research*, vol. 2, no. Dec, pp. 139–154, 2001.
[50] L. Bergman and Y. Hoshen, "Classification-based anomaly detection for general data," in *Proc. Int. Conf. Learn. Represent.*, 2020.
[51] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, and G. H. Chen, "Ecod: Unsupervised outlier detection using empirical cumulative distribution functions," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12 181–12 193, 2023.
[52] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *Proc. Int. Conf. Learn. Represent.*, 2018.
[53] H. Xu, G. Pang, Y. Wang, and Y. Wang, "Deep isolation forest for anomaly detection," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12 591–12 604, 2023.



Hongzuo Xu received the Ph.D degree in Computer Science and Technology from the National University of Defense Technology in 2023. His research interests include anomaly detection and its applications, with first-authored publications in ICML, WWW, AAAI, ICDM, CIKM, and IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING.

Yijie Wang received the PhD degree in computer science and technology from the National University of Defense Technology in 1998. She was awarded the prize of National Excellent Doctoral Dissertation by Ministry of Education of PR China. She is currently a Full Professor with the National Key Laboratory of Parallel and Distributed Computing, National University of Defense Technology. Her research interests include big data analysis, artificial intelligence and parallel and distributed processing.

Songlei Jian received the B.Sc. degree and Ph.D. degree in computer science from the College of Computer, National University of Defense Technology, Changsha, China, in 2013 and 2019, respectively. She is currently an Associate Research Fellow with the College of Computer, National University of Defense Technology. Her research interests include representation learning, graph learning, multimodal learning and anomaly detection.

Qing Liao received the Ph.D degree in computer science and engineering from the Department of Computer Science and Engineering of the Hong Kong University of Science and Technology, in 2016 supervised by professor Qian Zhang. She is currently a professor with the School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), China. Her research interests include artificial intelligence and data mining.

Yongjun Wang received the Ph.D. degree in computer architecture from the National University of Defense Technology, China, in 1998. He is currently a Full Professor with the College of Computer, National University of Defense Technology, Changsha, China. His research interests include network security and system security.

Guansong Pang is a tenure-track Assistant Professor of Computer Science in the School of Computing and Information Systems at Singapore Management University (SMU), Singapore. Before joining SMU, he was a Research Fellow with the Australian Institute for Machine Learning (AIML) at The University of Adelaide, Australia. He received a PhD degree from University of Technology Sydney in 2019. His research investigates novel data mining and machine learning techniques and their applications.