An Evaluation of Anomaly Detection and Diagnosis in Multivariate Time Series

Astha Garg, Wenyu Zhang, Jules Samaran, Ramasamy Savitha, *Senior Member, IEEE*, and Chuan-Sheng Foo

***Abstract*— Several techniques for multivariate time series anomaly detection have been proposed recently, but a systematic comparison on a common set of datasets and metrics is lacking. This article presents a systematic and comprehensive evaluation of unsupervised and semisupervised deep-learning-based methods for anomaly detection and diagnosis on multivariate time series data from cyberphysical systems. Unlike previous works, we vary the *model* and post-processing of model errors, i.e., the *scoring functions* independently of each other, through a grid of ten models and four scoring functions, comparing these variants to state-of-the-art methods. In time-series anomaly detection, detecting anomalous events is more important than detecting individual anomalous time points. Through experiments, we find that the existing evaluation metrics either do not take events into account or cannot distinguish between a good detector and trivial detectors, such as a random or an all-positive detector. We propose a new metric to overcome these drawbacks, namely, the composite F-score (Fc_1), for evaluating time-series anomaly detection. Our study highlights that dynamic scoring functions work much better than static ones for multivariate time series anomaly detection, and the choice of scoring functions often matters more than the choice of the underlying model. We also find that a simple, channel-wise model—the univariate fully connected auto-encoder, with the dynamic Gaussian scoring function emerges as a winning candidate for both anomaly detection and diagnosis, beating state-of-the-art algorithms.**

***Index Terms*— Anomaly detection, anomaly diagnosis, deep learning, evaluation, metrics, multivariate time series (MVTS).**

I. Introduction

MODERN cyberphysical systems (CPS), such as those encountered in manufacturing, aircraft, and servers, involve sophisticated equipment that records multivariate time-series (MVTS) data from 10s, 100s, or even thousands of sensors. The MVTS need to be continuously monitored to ensure smooth operation and prevent expensive failures. We focus on two key monitoring tasks on MVTS in this work. First, identifying time points during operation where sensors indicate deviation from normal behavior, which we refer to as the streaming time-series *anomaly detection* task. Second, pointing out the specific channel(s)[1] that deviate from normal behavior, which would aid the operator in verifying the anomaly, finding its root cause, and taking corrective action; we refer to this as the *anomaly diagnosis* task. In addition, while the data under healthy operation from sensor monitoring is abundant, there is typically a lack of labeled data for anomalous operation. We thus focus on the *semisupervised* setting, where the training data consists only of data from healthy operation [1], [2], and the *unsupervised* anomaly detection setting, where the training data is mostly healthy but may contain a small number of unknown anomalies [1].

Amongst recent works on semisupervised and unsupervised MVTS anomaly detection, deep learning-based methods have featured prominently [3]–[14], and several of these are state of the art. In this work, we aim to characterize the key design choices behind these deep learning methods and their effect on performance for anomaly detection and diagnosis tasks, to better understand the reasons for their success. Specifically, we observe that a majority of deep MVTS anomaly detection methods [3]–[9] work by learning feature representations of normality [2], under the assumption that a model trained to reconstruct or forecast patterns in healthy data would have a high error on anomalous test data [15], [16]. For this class of methods, the channel-wise model errors must be transformed and combined across channels using a *scoring function* (defined formally in Section V) to obtain a single anomaly score per time point. The score must then be thresholded for anomaly detection. Anomaly diagnosis may be carried out by ranking the channel-wise anomaly scores (or errors) for the duration of the anomaly, before aggregation, and returning the top-ranked channels for anomaly diagnosis. Our observations raise several interesting questions.

1) What is an appropriate scoring function to use for anomaly detection and diagnosis?
2) How important is the choice of scoring functions compared to the choice of model for good performance?
3) How well do existing algorithms perform on anomaly detection and diagnosis on data from real CPS systems?

A. Contributions

To address the questions identified above, we carry out the most comprehensive evaluation of deep semisupervised

Manuscript received October 20, 2020; revised April 16, 2021 and July 10, 2021; accepted August 12, 2021. Date of publication August 31, 2021; date of current version June 2, 2022. This research was supported by the Agency for Science, Technology, and Research (A*STAR) under its Industry Alignment Fund Pre-Positioning Programme (Health & Biomedical Sciences; Grant Number H19/01/a0/023) and Advanced Manufacturing and Engineering Programmatic Funds (Grant Number A20H6b0151). *(Corresponding author: Astha Garg.)*

Astha Garg was with the Institute of Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR), Singapore 138632. She is now with Chord X Pte. Ltd., Singapore 239920 (e-mail: astha.iitb@gmail.com).

Wenyu Zhang, Ramasamy Savitha, and Chuan-Sheng Foo are with I2R, A*STAR, Singapore 138632 (e-mail: Zhang_Wenyu@i2r.a-star.edu.sg; ramasamysa@i2r.a-star.edu.sg; foo_chuan_sheng@i2r.a-star.edu.sg).

Jules Samaran is with Mines Paristech, PSL Research University, 75006 Paris, France.

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TNNLS.2021.3105827.

Digital Object Identifier 10.1109/TNNLS.2021.3105827

[1] A channel refers to the time-series corresponding to a particular variable.

algorithms for MVTS anomaly detection and diagnosis to date, using real-world, publicly available CPS datasets. We show that deep anomaly detection methods that work by learning feature representations of normality can be put into a modular framework consisting of three parts—a model, a scoring function, and a thresholding function. Using this framework, we cross ten distinct models against four scoring functions to investigate the effect of independent choices of the model and the scoring function. We compare these combinations to other recently proposed end-to-end algorithms. In total, we evaluate 45 and 29 unique end-to-end algorithms on seven and four datasets for MVTS anomaly detection and diagnosis respectively. The code will be released at https://github.com/astha-chem/mvts-ano-eval. We make several interesting discoveries.

1) We find that *the choice of an appropriate scoring function can: a) boost the anomaly detection performance of existing methods and b) might matter more than the choice of the underlying model*. In particular, dynamic scoring functions, i.e., scoring functions that adapt to variations in the test set, outperform static scoring functions overall. To the best of our knowledge, dynamic scoring functions have not been investigated for deep anomaly detection for MVTS before.
2) Surprisingly, we find that the *univariate fully connected autoencoder (UAE)—a simple model, when used with dynamic scoring outperforms all other algorithms overall on both anomaly detection and diagnosis.* UAE consists of independent channel-wise fully connected auto-encoder (AE) models. This is a straightforward approach but has not been comprehensively evaluated before for MVTS anomaly detection.
3) In order to identify an appropriate metric for our evaluation, we compared existing F-score metrics for time series anomaly detection in terms of their robustness and ability to reward the detection of anomalous events. Significantly, we find that a popular metric, point-adjusted F_1 score [8], [10], [17] gives a close to perfect score of 0.96 to an anomaly detector that predicts anomalies randomly on one of the datasets. Thus, we find that the existing evaluation metrics either do not take events into account (F_1 score), or are not robust (point-adjusted F_1 score), and *we propose a new simple, yet robust metric for evaluating anomalous event detection—the composite F_1 score*, Fc_1.

II. Related Work

Recent surveys on general anomaly detection [16], deep-learning based anomaly detection [1], [2], and unsupervised time-series anomaly detection [18] review techniques relevant to unsupervised and semisupervised MVTS anomaly detection. Time-series anomaly detection techniques may be categorized based on their detection technique, namely, shallow model-based [19]–[22], deep-learning model based [3]–[6], [8]–[11], [14], [17], [23], pattern-based [24], [25], distance-based [26], [27], and nonparametric [28]. Of these, deep-learning-based techniques have received significant attention for MVTS anomaly detection owing to: 1) their ability to scale to high dimensions and model complex patterns in various domains, compared to straightforward statistical approaches such as out of limit approaches [5], [29], [30]; 2) fast inference and applicability to streaming time-series typical in CPS unlike many distance-based and pattern-based techniques that are not applicable to streaming time-series, as they require both training and test data during inference [24]–[27]; and 3) the ability to localize anomalous time points within sequences, unlike techniques [24], [25], [31] that work at the coarser level to detect anomalous subsequences. Anomaly diagnosis has been approached primarily from a supervised classification perspective [32]. In the unsupervised context, four studies [9]–[12] mention that ranking of scores or errors can be used to diagnose the cause of anomalies but only [10] shows experimental results on an open dataset.

A. Choice of Algorithms for Evaluation

We evaluate semisupervised and unsupervised deep anomaly detection techniques for MVTS, applicable to our problem setup of streaming time-series, localization of anomalies within sequences, and no anomalies in the training set. In terms of the categorization proposed by Pang *et al.* [2] for deep anomaly detection we include a range of techniques, summarized in Fig. S1 in the supplemental information (SI). We include: 1) algorithms that work by *generic normality feature learning*, using AEs—LSTM-ED [7], LSTM-VAE [33], MSCRED [9]; using Generative Adversarial Networks (GAN)—BeatGAN [4]; and using predictability modeling—NASA LSTM [5]; 2) technique that *learns anomaly-measure dependent features*—DAGMM [34]; and 3) *end-to-end anomaly scoring techniques*—OCAN [13] and OmniAnomaly [10]. We include representative shallow techniques—principal component analysis (PCA) [35] and one-class support vector machine (OC-SVM) [36]. In addition, we compare published results for OmniAnomaly, USAD [8] and MAD-GAN [3] with ours in Section S5 in the SI. We did not evaluate some recently proposed algorithms as their source code is proprietary and is not straightforward to implement [11], [14].

III. Problem Setup

A. Anomaly Detection

We are given a training MVTS, $\mathbf{X}^{\text{train}} \in \mathbb{R}^{n_1 \times m}$ with n_1 regularly sampled time points and m channels. $\mathbf{X}^{\text{train}}$ is known or assumed to contain no anomalies. The task is to predict whether an anomaly occurred at each time point t in the test time-series $\mathbf{X}^{\text{test}} \in \mathbb{R}^{n_2 \times m}$ with n_2 time points and $1 \leq t \leq n_2$. When making a prediction for time t, we assume that the test time-series has only been observed until time t to simulate a streaming scenario [19]; we also assume that $\mathbf{X}^{\text{train}}$ is not available at test time.

B. Anomaly Diagnosis

We consider the anomaly diagnosis task independently; it is also referred to as anomaly interpretation [10]. Given $\mathbf{X}^{\text{train}}$ and $\mathbf{X}^{\text{test}}$ as above, as well as the start and end times of each anomalous event, we want to predict the specific channel(s) that deviate from normal behavior. In this article, we refer to

TABLE I
SUMMARY OF DATASET CHARACTERISTICS USED IN THIS ARTICLE. AVERAGING IS DONE OVER ENTITIES

Name	Domain	Entities	Channels, m	Average train length	Average test length	Average % anomalies	Average num events	Event time (mins)	Time step (s)	l_w	l_s
SWaT [37]	Water treatment	1	51	473400	414569	4.65%	35	1.7-28.1	1	100	10
WADI [38]	Water distribution	1	123	1209601	172801	5.76%	14	1.5-29	1	30	10
DMDS [39]	Sugar manufacturing	1	32	507600	217802	1.48%	17	0.2-10.3	1	100	10
SKAB [40]	Water circulation	1	8	9401	35600	36.70%	34	2.4-9.8	1	100	1
MSL [5]	Spacecraft	27	55	2160	2731	12.02%	1.33	11-1141	60	100	1
SMAP [5]	Spacecraft	55	25	2556	8071	12.40%	1.26	31-4218	60	100	1
SMD [10]	Server monitoring	28	38	25300	25301	4.21%	11.68	2-3160	60	100	1

these as *causes*. Note that we do not imply these to be the root causes, but rather that these are the causes due to which the algorithm flagged an anomaly.

IV. DATASETS

We use seven publicly available MVTS datasets from real-world CPS (Table I), characterized by a regular sampling rate, periodicity in several channels, and strong correlations across time and channels (e.g., SI Fig. S2). The training sets are known or assumed to be anomaly-free. A few channels in these datasets stay constant in the training set but they may help to detect anomalies in the test set and are not discarded. To model each time-series, we break it into overlapping windows of length l_w and a step size, l_s such that $1 \leq l_s \leq l_w$ (Table I). We consider three *multientity* datasets (MSL, SMAP, SMD), where each entity is a different physical unit of the same type, having the same dimensionality. Similar to [5] and [10], we train a separate model for each entity in the multientity datasets. The other four datasets are *single-entity* datasets.

The anomalies were induced knowingly by physically compromising the operation in a steady-state system, for datasets other than SMAP and MSL. The ground truth of the root cause is known for SWaT, WADI, DMDS, and SMD datasets. For SMAP and MSL, the anomalies are expert-labeled manually based on past reports of actual spacecraft operation [5]. Unlike the other datasets, each entity in MSL and SMAP consists of only 1 sensor, while all the other channels are one-hot-encoded commands given to that entity. For MSL and SMAP, we use all channels as input to the models, but use the model error of only the sensor channel for anomaly detection. The datasets are summarized in Table I and additional details including train-test splits are provided in SI Section S2.

Broadly, there are two types of anomalies in MVTS datasets—*temporal anomalies*, where one or more channels deviate from their normal behavior when compared to their respective history (e.g., malfunctioning of a monitored part in the system), and *cross-channel anomalies*, where each channel individually looks normal with respect to its history, but its relationship to other channels is abnormal. Based on author descriptions of anomalies for SWaT [37], WADI [38], DMDS [39], MSL [5] and SMAP [5] datasets, and our own visualizations (e.g., Fig S3), the datasets we test primarily contain *temporal anomalies*. While we expect temporal anomalies to be more common in steady-state CPS than strictly cross-channel anomalies, datasets showing multiple operational states such as power plant operation in [22] may be more likely to have cross-channel anomalies; we did not find open datasets where cross-channel anomalies are known to be present.

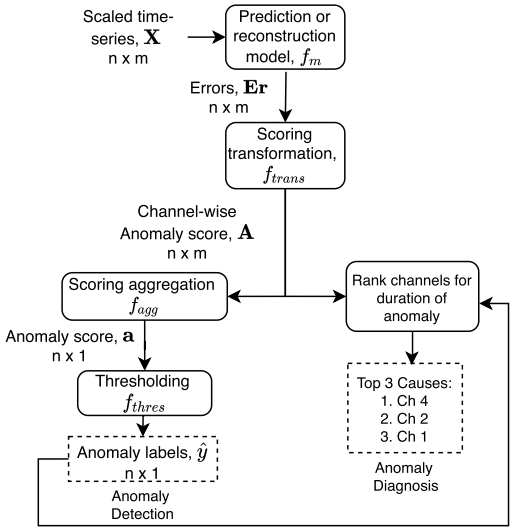

Fig. 1. Framework for anomaly detection and diagnosis.

V. MODULAR FRAMEWORK

A majority of deep anomaly detection approaches for MVTS are based on training a model to reconstruct or predict healthy time-series [3]–[10]. We generalize these model-based approaches into a modular framework for anomaly detection and diagnosis consisting of three parts, or modules—a reconstruction or prediction model, a scoring function, and a thresholding function (Fig. 1).

The first module is a *reconstruction or prediction model*, f_m, that takes as input a subsequence, and outputs a lossy reconstruction of the input or a prediction of the next (or several) time point(s). f_m is used to calculate an error, $\mathbf{Er}_t^i$, where the superscript i denotes channels and the subscript t denotes the tth time point. f_m is not limited to neural network models and could be, for example, PCA [35], ARIMA [41], or HMM [19] models.

The second module, *scoring function* combines a multichannel score such as errors $\mathbf{Er}$ (or latent representation as in DAGMM [34]) into a single anomaly score $\mathbf{a}_t$ per time point. The scoring function usually has two distinct but related parts—a transformation function f_{trans} and an aggregation function f_{agg}. First, f_{trans} transforms the errors into the channel-wise anomaly score, $\mathbf{A}_t^i$, which can be used for anomaly diagnosis to identify which channel has the highest

TABLE II
ALGORITHMS EVALUATED IN THIS WORK

Model + Scoring function	Threshold
Models crossed with scoring functions	
Models: Raw Signal, PCA [35], UAE, FC AE, LSTM AE [7], TCN [42] AE, LSTM VAE [6], BeatGAN [4], MSCRED [9], NASA LSTM [5]. Scoring: Error, Gauss-S, Gauss-D, Gauss-D-K	best-F-score, top-k, tail p
Models with pre-defined scoring	
DAGMM [34], OmniAnomaly [10], OCAN [13]	best-F-score, top-k, tail p
NASA LSTM NPT [5]	Non-parametric threshold
OC-SVM [36]	Thresholding at 0.5

anomaly score for the duration of the anomaly. Then, f_{agg} aggregates the scores across channels resulting in the anomaly score, $\mathbf{a}_t$ for each time point.

The final module, the *thresholding function*, f_{thres} thresholds $\mathbf{a}_t$ to obtain a binary label $\hat{\mathbf{y}}_t$ to classify each point as anomalous or healthy. Formally, for $\mathbf{X} = \mathbf{X}_{\text{test}}$,

$$\mathbf{Er}(\mathbf{X}) = \mathbf{X} - f_m(\mathbf{X}); \quad \mathbf{A}(\mathbf{X}) = f_{\text{trans}}(\mathbf{Er}(\mathbf{X})) \tag{1a}$$

$$\mathbf{a}(\mathbf{X}) = f_{\text{agg}}(\mathbf{A}(\mathbf{X})); \quad \hat{\mathbf{y}} = f_{\text{thres}}(\mathbf{a}(\mathbf{X})). \tag{1b}$$

VI. ALGORITHMS

First, we discuss the models for which we vary scoring functions in Section VI-A, followed by models with pre-defined scoring functions in Section VI-B. We also relate these models to the categorization proposed by Pang *et al.* [2], shown in SI Fig. S1. We discuss scoring functions and thresholding functions separately in Sections VI-C and VI-D, respectively. All the algorithms evaluated are summarized in Table II.

A. Models Where Scoring Functions Are Varied

1) Shallow Models: We test *Raw Signal* as a trivial "model" that reconstructs any signal to 0, so that the error is the same as the normalized signals. We also test *PCA* for lossy reconstruction [35] by retaining only a subset of principal components (PC) that explain 90% of the variance.

2) Generic Normality Feature Learning Models: An AE [43] is an unsupervised deep neural network that is trained to reconstruct the input through a compressed latent representation, using an encoder and a decoder. The AE is the basis of many deep-learning-based models for MVTS anomaly detection [7]–[9], thus we include various architectures for this. In the UAE model, we train a separate AE for each channel. Fully connected AE (FC AE) takes data concatenated across channels as input and hence can capture relationships between channels. *Long short term memory AE (LSTM AE)* is based on the LSTM encoder–decoder model by Malhotra *et al.* [7]. While Malhotra *et al.* [7] used only the first PC of the MVTS as input to LSTM-ED, we instead set 90% explained variance for PCA to minimize information loss. *Temporal convolutional network AE (TCN AE)* is based on the TCN model proposed by Bai *et al.* [42], where we stack TCN residual blocks for the encoder, and we replace the convolutions in TCN residual blocks with transpose convolutions, for the decoder. *LSTM variational auto-encoder (LSTM VAE)* models the data generating process from the latent space to the observed space and is trained using variational techniques [6], [44]. Lastly, the recently proposed *MSCRED* [9] learns to reconstruct signature matrices, i.e. matrices representing cross correlation relationships between channels constructed by pairwise inner-product of the channels. Due to the high dimensionality of WADI, we compress WADI to 90% explained variance by PCA before applying MSCRED.

In addition to AEs, we include other generic feature normality learning techniques. *BeatGAN* [4] uses a generative adversarial network (GAN) framework where reconstructions produced by the generator are regularized by the discriminator instead of fixed reconstruction loss functions. *NASA LSTM* is a two-layer LSTM model that uses predictability modeling, i.e., forecasting for anomaly detection [5]. Hundman *et al.* [5] also proposed a scoring function and threshold for this model which we test separately as *NASA LSTM nonparametric thresholding (NPT)*, consisting of the exponentially weighted moving average (EWMA) of the root-mean-square of sensor prediction errors as the scoring function, and an NPT with pruning.

B. Models With Predefined Scoring Functions

1) Anomaly Measure-Dependent Feature Learning Models: We test *deep auto-encoding Gaussian mixture model (DAGMM)* [34], an algorithm for unsupervised multivariate anomaly detection (that can also be used in the semisupervised setting) where an AE is trained end-to-end with the scoring function—a Gaussian mixture model (GMM), fit over the concatenation of the hidden space and summary metrics from the reconstruction, that returns the anomaly score $\mathbf{a}$ directly. We also test *OC-SVM* [36], a classic shallow technique for unsupervised anomaly detection via one-class classification. OC-SVM learns the hyperplane encompassing normal samples and returns a binary classification label based on the side of the separating hyperplane that the sample is on. To reduce computational complexity of OC-SVM, we compress the samples by PCA retaining 90% explained variance.

2) End-to-End Anomaly Scoring Algorithms: One-class adversarial nets, OCAN [13] is an end-to-end one-class classification method. Here a generator is trained to produce examples *complimentary* to healthy patterns, which is used to train a discriminator for anomaly detection via the GAN framework. *OmniAnomaly* [10] is a prior-driven stochastic model for MVTS anomaly detection that directly returns the posterior reconstruction probability of the MVTS input. The log of the probability serves as the channel-wise score, which is summed across channels to get the anomaly score.

C. Scoring Functions

We use four scoring functions for models in Section VI-A.

1) Normalized Errors: Errors, $\mathbf{Er}_t^i$ can be used directly as the anomaly score [45]. In order to account for differences in the training error across channels, we subtract the channel-wise mean training reconstruction error from the test errors, before taking the root-mean-square across channels.

2) Gauss-S: Similar to [7], we fit a Gaussian distribution to the training errors and design a score based on the fit distribution. We note that directly using the probability distribution

function (pdf) as in $-\log$ pdf would give points at both tails of the distribution high scores. In particular, this means that even points with very low reconstruction errors will be classified as being anomalous. To avoid this, we instead develop a score based on the cumulative distribution function (cdf): $-\log(1-\text{cdf})$, that increases monotonically with reconstruction error as f_{trans}. To obtain the final anomaly score $\mathbf{a}_t$, we simply add the channel-wise scores [10], assuming independence, as suggested by Ahmad *et al.* [19]. Formally, with $\hat{\mu}^i, \hat{\sigma}^i$ the empirical mean and standard deviation, respectively, of the channel-wise errors, and Φ the cdf of $N(0,1)$

$$\mathbf{A}_t^i = -\log\left(1-\Phi\left(\frac{\mathbf{Er}_t^i-\hat{\mu}^i}{\hat{\sigma}^i}\right)\right); \quad \mathbf{a}_t = \sum_{i=1}^{m}\mathbf{A}_t^i. \tag{2}$$

3) Gauss-D: In the dynamic gaussian scoring function, we replace the static mean and variance of Gauss-S in (2) with dynamic mean and variance, $\hat{\mu}_t^i$ and $\hat{\sigma}_t^i$, in order to adapt better to long-term changes occurring during the testing phase [19]

$$\hat{\mu}_t^i = \frac{1}{W}\sum_{j=0}^{W-1}\mathbf{Er}_{t-j}^i; \quad \left(\hat{\sigma}_t^i\right)^2 = \frac{1}{W-1}\sum_{j=0}^{W-1}\left(\mathbf{Er}_{t-j}^i-\hat{\mu}_t^i\right)^2 \tag{3}$$

where W is the window size. We prepend the last $W-1$ values from the training set to the test set in order to compute $\hat{\mu}_t^i$ and $\hat{\sigma}_t^i$ for $t < W$, the initial part of test.

4) Gauss-D-K: This scoring function refers to Gaussian kernel convolution applied on top of Gauss-D [19]. The smoothing of the score before aggregation can potentially amplify the total anomaly score even when multiple channels respond to an anomaly at slightly different times, unlike the previous three scoring functions. $\mathbf{A}_t^i$ for the Gauss-D-K scoring function is

$$G(u;\sigma_k) = e^{-\frac{1}{2}\left(\frac{u}{\sigma_k}\right)^2}; \quad \mathbf{A}_t^i = G * \mathbf{A}_{t,\text{Gauss-}D}^i \tag{4}$$

where G is a Gaussian filter with kernel sigma σ_k, $*$ the convolution operator and $\mathbf{A}_{t,\text{Gauss-}D}^i$ the channel-wise anomaly scores from Gauss-D. $\mathbf{a}_t$ is the same as that in (2).

D. Thresholding Functions

A validation set with anomalies could be used to set a static threshold [7], [9] but we do not assume the availability of such a validation set in our study. We use the following three thresholding functions to evaluate algorithms:

1) Best-F-Score: This is the static threshold that results in the maximum value of the desired metric (F_1, point-adjusted F_1 or Fc_1) for a given $\mathbf{a}_t$. We use the best-F-score threshold to get an upper limit on the anomaly detection performance of an algorithm for a static threshold, similar to [3], [8], and [10], and also for comparison of various metrics in Section VII-B. However, this requires the use of labels in the test set, so it is not applicable in practice.

2) Top-k [34]: The top-k threshold is the threshold that results in exactly k time points being labeled as anomalous, where k is the actual number of anomalies in the test set.[2] We use this threshold to compare across algorithms since k is constant across algorithms. However, since the top-k threshold requires anomaly scores for the full test set before thresholding, it is only applicable to a nonstreaming scenario.

[2]We set a separate top-k threshold for each entity.

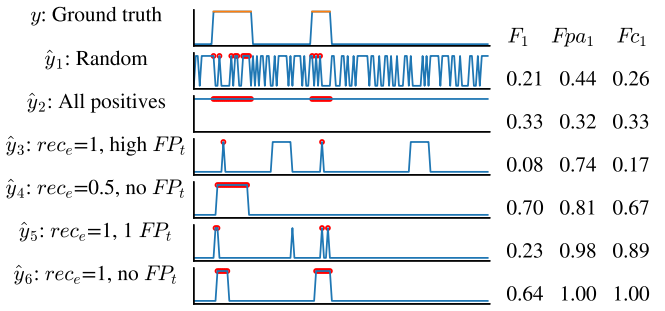

Fig. 2. Synthetic ground-truth anomaly labels (topmost, y), with anomalies denoted as 1 (orange), followed by six sets of predicted labels, $\hat{y}_1$ through $\hat{y}_6$. The y-axis range is 0–1, and circles (in red) indicate TP time points on each plot. The scores from each metric are on the right side.

3) Tail-p: When the anomaly scores correspond to probabilities, the tail-p threshold labels scores $\mathbf{a}_t < \epsilon$ as anomalous, where ϵ is a small tail probability [19]. This is applicable to Gauss-S, Gauss-D, and Gauss-D-K scoring functions in a streaming scenario. Since our scoring function for MVTS is a sum of m negative log probabilities, we set the threshold as $\text{th}_{\text{tail-}p} = -m\log_{10}(\epsilon)$ (except SMAP and MSL, where $\text{th}_{\text{tail-}p} = -\log_{10}(\epsilon)$ based only on the single sensor channel error). In practice, ϵ may be tuned during the initial part of testing, therefore, here we test five different values of $-\log_{10}(\epsilon) \in \{1,2,3,4,5\}$ for each algorithm and dataset pair, and report the highest score obtained. For the multientity datasets, we pick a single best value of $-\log_{10}(\epsilon)$ to be applied to all entities.

VII. Evaluation Metrics

A. Composite F-Score

Anomaly detection in practice needs a threshold to get $\hat{y}$. Some authors [6], [23] evaluate anomaly detection for all possible thresholds using the area under the precision–recall curve (AU-PRC) or the receiver operating characteristic (AU-ROC), but it is more important in practical applications to have a high F-score for a chosen threshold [17]. The point-wise F-score (F_1) [3], [6], [8], [9] is the simplest, but in practice, operators care more about detecting events, i.e., a continuous set of anomalous time points, rather than individual time points [17]. The event-wise F-score, proposed by Hundman *et al.* [5] takes events into account, but since it counts only one false-positive (FP) for a continuous set of time points, it rewards a detector that labels the entire test-set as a single anomalous event. The point-adjusted F-score (Fpa_1) proposed by Xu *et al.* [17], and used by Audibert *et al.* [8] and Su *et al.* [10] also accounts for events, but it might give a high score even when a large number of events are not detected [8]. In the Fpa_1 score, if any time point within an event is a true positive (TP), all the time points within that event are counted as TPs, and then a point-wise F-score is calculated.

An ideal detector would be one that detects at least one time point in each event and has no FPs. Early detection is also desirable [19] but is not the subject of our study and has not been considered in any of the works that we compare

TABLE III
METRICS WITH THE BEST-F-SCORE THRESHOLD, COMPARING RAD AGAINST UAE (WITH GAUSS-D SCORING). BOLD VALUES INDICATE CASES WHERE A METRIC RANKS RAD HIGHER THAN UAE

Metric	DMDS		MSL		SKAB		SMAP		SMD		SWaT		WADI	
	RAD	UAE	RAD	UAE	RAD	UAE	RAD	UAE	RAD	UAE	RAD	UAE	RAD	UAE
Point-wise F_1	0.0293	0.5311	0.2115	0.4514	0.5369	0.5375	0.2055	0.3898	0.0819	0.4351	0.0888	0.4534	0.1090	0.3537
Point-adjusted F_1	0.6371	0.7234	0.8512	0.9204	**0.9858**	0.9695	0.7418	0.8961	0.7585	0.9723	**0.9170**	0.8685	**0.9613**	0.9574
Fc_1	0.0345	0.6719	0.3242	0.7132	0.5444	0.5612	0.2895	0.8088	0.1067	0.8325	0.1215	0.6953	0.1317	0.5303

against. Thus, a good metric for MVTS anomaly detection should 1. reward high precision and high event-wise recall, 2. give a useful indication of how far we are from the ideal case. Following directly from the first criterion, we propose a new metric for time-series anomaly detection—the composite F_1 score, Fc_1. Fc_1 is the harmonic mean of the time-wise precision, Pr_t and event-wise recall, Rec_e

$$\text{Pr}_t = \frac{\text{TP}_t}{\text{TP}_t + \text{FP}_t} \quad \text{and} \quad \text{Rec}_e = \frac{\text{TP}_e}{\text{TP}_e + \text{FN}_e} \tag{5}$$

where TP_t and FP_t are the number of TP and FP time points, respectively, while TP_e and FN_e are the number of TP and false negative (FN) events, respectively. TP_e is the number of true events for which there is at least one TP time point. Remaining true events are counted under FN_e.

B. Comparison of Metrics for Anomaly Detection

To gain intuition on what each metric rewards in a labeling scheme, we compare the three F-scores—F_1, Fpa_1 and Fc_1 for predicted labels on synthetic test cases shown in Fig. 2. All three metrics give low scores to the spurious labeling schemes $\hat{y}_1$ and $\hat{y}_2$, and all metrics suggest that $\hat{y}_4$ is better than $\hat{y}_1$-$\hat{y}_3$. However, the Fpa_1 score for $\hat{y}_3$ is 0.74, which suggests good performance in spite of high FPs. $\hat{y}_5$ is highly desirable due to high rec_e and low FPs but the F_1 score is low. $\hat{y}_6$ is an ideal detector but F_1 ranks it lower than $\hat{y}_4$. Each metric ranks these labeling schemes differently, but Fc_1 is the only one that gives low scores (e.g., < 0.6) to the undesirable predictions $\hat{y}_1$–$\hat{y}_3$ and higher scores to the desirable predictions $\hat{y}_4$-$\hat{y}_6$, while correctly suggesting that $\hat{y}_6$ is ideal.

Furthermore, in order to establish baseline values of various F-score metrics on each dataset, we evaluate the performance of a trivial detector—the random anomaly detector (RAD) with the best-F-score threshold (Table III). RAD simply assigns a random real score in $[0, 1]$ for each time point. The Fpa_1 score suggests good performance ($\text{Fpa}_1 > 0.6$) for RAD on all datasets, while the F_1 and Fc_1 behave as expected, showing low scores. Table III also shows results for UAE with Gauss-D scoring, which is the best overall algorithm for anomaly detection in our study. Surprisingly, the Fpa_1 score suggests that RAD is *better* than UAE on SKAB, SWaT and WADI. RAD's Fpa_1 score of 0.9613 on WADI is attained at a threshold that labels only 0.17% time points as anomalous, so that the number of FPs is kept low, but the number of TPs is exaggerated as the noisy scoring function can label at least 1 time point in most events. Thus, Fig. 2 and Table III show that the F_1 score may be too pessimistic, and the Fpa_1 may be too optimistic in some cases.

C. Metrics for Anomaly Diagnosis

For anomaly diagnosis, we assume that the ground-truth of anomalous time points is known but the causes are unknown. We use HitRate@150 [10] and the Root Cause-top-3 (RC-top-3) metrics to evaluate anomaly diagnosis. The HitRate@150 [10] metric gives the average fraction of overlap between the true causes and the top $1.5c$ identified causes, where c is the number of true causes. The RC-top-3 (or RC-top-k) is a new metric we propose, which gives the fraction of events for which at least one of the true causes was identified to be in the top 3 (or top k) causes identified by the algorithm. HitRate@150 rewards identifying *all* of the true causes while RC-top-3 rewards identifying at least one of the causes.

VIII. EXPERIMENTAL SETUP

A. Dataset Preprocessing

We carry out channel-wise min–max normalization for each channel squashing it in the range of [0, 1] during train and clipping it to [min−4, max+4], i.e., the range $[-4, 5]$ during the test to prevent excessively large values from a particular channel skewing the overall scores. The window length (l_w) and step size (l_s) are shown in Table I. We choose $l_w = 100$, similar to [10] in the absence of specific knowledge of the time-scales in the datasets, but choose a smaller $l_w = 30$ for WADI due to its higher dimensionality. We choose $l_s = 1$ for SKAB, MSL, SMAP and SMD datasets as they are shorter and $l_s = 10$ for the remaining, longer datasets to speed up training.

B. Hyperparameters and Implementation

When available, we adapt preexisting implementations (Table S1 in SI). We tune the hyperparameters of LSTM-VAE, TCN AE, and FC AE models using random search over the search space shown in the SI, Table S2 choosing the configuration with the minimum reconstruction error on the validation set, as done by Malhotra *et al.* [7]. We set the hyperparameters empirically for the remaining algorithms, summarized in SI Table S3. An extensive hyperparameter tuning for each model and dataset is beyond the scope of this work. Runtimes of the models on 3 datasets are provided for reference in SI Table S4. We choose W for Gauss-D and Gauss-D-K scoring functions to be comparable to training set size, $W = 100\,000$ for SWaT, WADI and DAMADICS, $W = 2000$ for MSL and SMAP, $W = 25\,000$ for SMD and $W = 100$ for SKAB. We set σ_k for Gauss-D-K scoring empirically to 120 for SWaT and WADI, 5 for DAMADICS, 10 for SMAP and MSL, and 1 for SMD and SKAB.

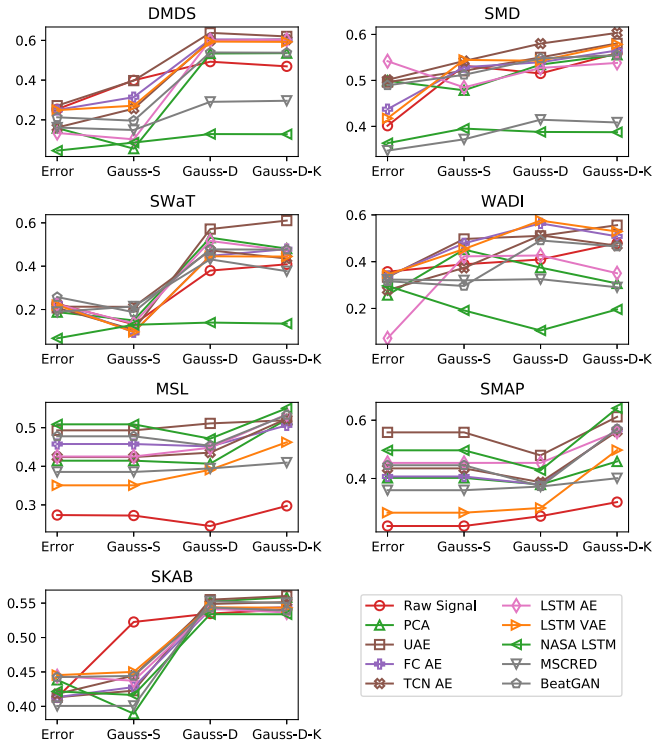

Fig. 3. Effect of scoring functions on the Fc_1 performance with top-k threshold for various models and datasets. Results in this plot are also provided as a supplemental csv file.

C. Evaluation

We use Fc_1 score with the top-k threshold as the primary metric in our anomaly detection evaluation. For completeness, we report additional results for Fc_1 score (SI Sections S8.2 and S8.3) with best-F-score and tail-p thresholds, respectively, the F_1 score with the best-F-score threshold (SI Section S8.4), AU-ROC score (SI Section S8.5), and AU-PRC score (SI Section S8.6). For anomaly diagnosis, we report the RC-top-3 scores (main manuscript) and HitRate@150 (SI Section S8.7). For multientity datasets, anomaly detection and diagnosis metrics are averaged over entities. We run each experiment 5 times with different seeds and report the average. We use the recommendations by Demšar [46] to compare statistical significance at significance level $\alpha = 0.05$. We conduct an overall comparison using the Friedman test [47], under the null hypothesis that all methods perform the same. If we can reject the null hypothesis, we conduct post-hoc tests using Hochberg's step-up procedure [48], comparing the best-performing method against all others. The details of the tests are described in SI Section S7.

IX. Results

A. Anomaly Detection

Fig. 3 shows dataset-wise effect of the choice of model (colors and symbols), and scoring function (X-axis) on the Fc_1 performance for the top-k threshold. We see that both the model and scoring function can make a significant difference to the Fc_1 score.

1) Effect of Scoring Function: For DMDS, SMD, SWaT, and SKAB, the choice of scoring function, in general, makes a bigger difference than the choice of model. For example, the Fc_1 score for the UAE model on SWaT ranges from ∼0.1–0.6 for different scoring functions, while the performance of all models (except NASA LSTM) for the Gauss-D-K scoring function is in the range ∼0.4–0.6. The ranks of the scoring functions, averaged across datasets, and models shown in Fig. 3 are Gauss-D-K—1.5, Gauss-D—2.0, Gauss-S—3.2 and Error—3.3. Statistical tests indicate that the differences between the performance of the dynamic scoring functions—Gauss-D-K, Gauss-D versus the static ones—Gauss-S, Errors are statistically significant (see SI Section S7.1).

TABLE IV

Average Fc_1 Score, With the Gauss-D Scoring Function (Except Those Denoted With *) and Top-k Threshold. See SI Table S6 for Standard Deviation

	DMDS	MSL	SKAB	SMAP	SMD	SWaT	WADI	Mean	Rank
Raw Signal	0.4927	0.2453	0.5349	0.2707	0.5151	0.3796	0.4094	0.4068	9.3
PCA	0.5339	0.4067	0.5524	0.3793	0.5344	0.5314	0.3747	0.4733	5.6
UAE	**0.6378**	**0.5111**	**0.5550**	**0.4793**	0.5501	**0.5713**	0.5105	**0.5450**	**1.6**
FC AE	0.6047	0.4514	0.5408	0.3788	0.5395	0.4478	0.5639	0.5038	4.7
LSTM AE	0.5999	0.4481	0.5418	0.4536	0.5271	0.5163	0.4265	0.5019	4.7
TCN AE	0.5989	0.4354	0.5488	0.3873	**0.5800**	0.4732	0.5126	0.5052	3.9
LSTM VAE	0.5939	0.3910	0.5439	0.2988	0.5427	0.4456	**0.5758**	0.4845	6.0
BeatGAN	0.5391	0.4531	0.5437	0.3732	0.5479	0.4777	0.4908	0.4894	5.0
MSCRED	0.2906	0.3944	0.5526	0.3724	0.4145	0.4315	0.3253	0.3973	8.1
NASA LSTM	0.1284	0.4715	0.5339	0.4280	0.3879	0.1398	0.1058	0.3136	8.9
DAGMM*	0.0000	0.1360	0.0000	0.1681	0.0187	0.0000	0.0256	0.0498	12.9
OmniAnomaly*	0.1425	0.4120	0.4561	0.3767	0.5002	0.1466	0.2443	0.3255	9.4
OCAN*	0.2532	0.3009	0.4369	0.2787	0.4614	0.1547	0.0000	0.2694	11.0

The dynamic scoring functions outperform the static scoring functions as they adapt to the changing normal during the test, but they can also adapt to anomalies in the test set. This does not affect the overall score much when the score is aggregated across channels and when the Gauss-D window length, W is large compared to anomalous event lengths. However, for SMAP and MSL, no aggregation of scores takes place, and for SMAP the average event length (1001 min) is comparable to W (2000 min), which may be why Gauss-D performs worse than Gauss-S on SMAP.

Gauss-D-K performs better overall than Gauss-D by amplifying anomaly scores from multiple channels responding to the same anomaly, even if they respond at slightly different times. However, Gauss-D-K performance is sensitive to the value of σ_k which is not straightforward to set, and we set it empirically here in a nonrigorous search to approximately attain good test performance. In contrast, the window size, W for Gauss-D was set simply as a size comparable to the training set size. For this reason, we choose the Gauss-D scoring function to compare the effect of the choice of the model on anomaly detection.

2) Effect of Model: Table IV summarizes the Fc_1 score for models using either Gauss-D scoring or predefined scoring functions (starred), with the top-k threshold. We see that UAE is the best performing model for five out of seven datasets, and is leading overall by mean Fc_1 and average rank. We find that the difference between the performance of UAE against Raw Signal and several recently proposed algorithms—DAGMM [34], OCAN [13], OmniAnomaly [10], NASA LSTM [5] and MSCRED [9] is statistically significant, while the remaining comparisons are statistically insignificant (see SI Section S7.2), perhaps due to the large number of algorithms tested with just 7 datasets. UAE is also the top performing in Tables S7, S10, S11, S12, S13, S14, and S15 in

the SI where we compare models under other scoring functions (Gauss-D-K) or other metrics—point-wise F_1, AU-ROC and AU-PRC.

The superior performance of UAE could be attributed to its channel-wise models that learn and retain information about each channel before aggregating scores across channels, enabling it to effectively detect temporal anomalies which are prevalent in these datasets. Another question is why do other sophisticated techniques that model both temporal and multivariate correlations perform worse than UAE? The reason may be that in a dataset that has large temporal and inter-channel correlations (typical of the datasets we use), the two effects can get confounded, resulting in spurious correlations (e.g., SI Fig. S5). The space of possible anomalies is also much larger when both cross-channel and temporal anomalies are considered by the algorithm, which might result in more noisy anomaly detection behavior.

The second best performing model by average rank is FC AE. Interestingly, FC AE and TCN AE perform better than RNN based methods—LSTM AE, LSTM VAE, and OmniAnomaly, even though RNN based methods are deemed to be better suited to time-series tasks. This may be because the fixed length time-series can easily be modeled by FC and TCN units as well. The poor performance of OmniAnomaly [10], DAGMM [34] and OCAN [13] could be attributed partly to the use of static scoring functions. The forecasting model NASA LSTM performs poorly for all scoring functions in Fig. 3 for multiple datasets, suggesting that AE-based techniques might be better than forecasting techniques for semisupervised MVTS anomaly detection.

Fig. 4 shows the prec_t and rec_e values for the algorithms presented in Table IV with the top-k threshold. While the best algorithms are able to attain a high event-wise recall, time-wise precision for every dataset and algorithm is below 0.5. This suggests that there might exist better thresholds that tag fewer anomalies, which could improve prec_t without lowering rec_e too much; Table S8 in SI shows Fc_1 score with the Gauss-D scoring function and the best-F-score (best-Fc_1 in this case) threshold, and these scores are generally higher than the top-k threshold scores shown in Table IV.

Table S9 in the SI shows Fc_1 scores for all datasets and algorithms with the Gauss-D scoring function and tail-p threshold, or a different streaming threshold as noted in Table S9. The tail-p threshold results show similar trends to the top-k threshold results of Table IV, with UAE leading the pack based on overall mean and average rank. The precision–recall results in the SI Fig. S8 show that with the streaming threshold, different techniques pick different tradeoffs between prec_t and rec_e, and higher values of prec_t are attained than those shown in Fig. 4.

B. Anomaly Diagnosis Results

Fig. 5 shows the effect of scoring functions and models on the independent anomaly diagnosis performance using the RC-top-3 metric. We see that for DMDS and SMD, scores of ∼0.95 are achieved by the best algorithms, suggesting that ranking channels by anomaly score is an effective strategy for independent anomaly diagnosis.

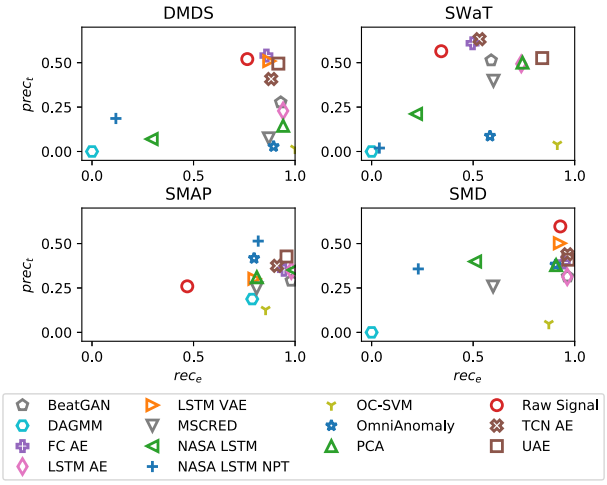

Fig. 4. Plots of prec_t versus rec_e for algorithms with the top-k threshold and scoring functions as in Table IV. See additional datasets in SI Fig. S6.

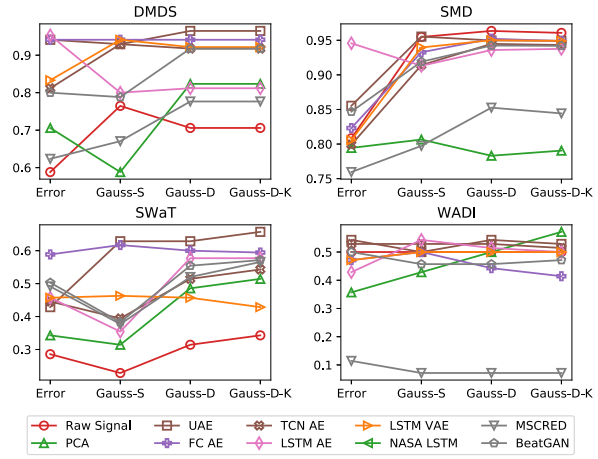

Fig. 5. Effect of scoring functions on the RC-top-3 performance for various algorithms and datasets.

TABLE V

RC-TOP-3 METRIC USING THE GAUSS-D SCORING FUNCTION (EXCEPT STARRED). SEE SI TABLE S16 FOR STANDARD DEVIATION

Algo	DMDS	SMD	SWaT	WADI	Overall mean	Avg Rank
Raw Signal	0.7059	**0.9635**	0.3143	0.5000	0.6209	6.5
PCA	0.8235	0.7831	0.4857	0.5000	0.6481	7.2
UAE	**0.9647**	0.9498	**0.6286**	**0.5428**	**0.7715**	**1.8**
FC AE	0.9412	0.9522	0.6000	0.4428	0.7341	3.8
LSTM AE	0.8117	0.9360	0.5771	0.5143	0.7098	5.2
TCN AE	0.9177	0.9448	0.5143	0.5286	0.7263	4.5
LSTM VAE	0.9177	0.9501	0.4571	0.5000	0.7062	4.8
BeatGAN	0.9176	0.9424	0.5543	0.4571	0.7178	5.8
MSCRED	0.7765	0.8526	0.5200	0.0714	0.5551	8.2
OmniAnomaly*	0.9177	0.9272	0.3772	0.4857	0.6770	6.25

The RC-top-3 scores for SWaT and WADI are lower because in these datasets, for some events, channels other than the original causes can also be affected by an anomalous event, and can get diagnosed as the causes (e.g., SI Fig. S3b). The HitRate@150 scores in SI Fig. S10 are lower than RC-top-3 results since HitRate@150 evaluates the ability to identify all of the root causes, rather than at least one. However, the trends across datasets, models, and scoring functions are similar to that seen in Fig. 5. Among scoring functions, Gauss-D has the

highest average rank (2.0) but we find the differences between scoring functions to be statistically insignificant.

C. Effect of Model

Table V shows the RC-top-3 results for various models with the Gauss-D scoring function (except OmniAnomaly). Again, UAE is the top performer with an average rank of 1.5. Similar results are seen in SI Table S17 showing the HitRate@150 performance. Thus, even though we evaluate anomaly detection and diagnosis independently, the results from both evaluations are consistent and confirm the intuition that an algorithm that is good at anomaly detection, i.e., ranking the anomaly score correctly across time points, is also good at anomaly diagnosis, i.e., ranking the scores correctly across channels, before aggregation. However, the overall comparison across models on anomaly diagnosis is not statistically significant, likely due to a large number of comparisons on just four datasets.

X. Conclusion and Future Work

We conducted a comprehensive evaluation of deep-learning-based algorithms on MVTS anomaly detection and independent anomaly diagnosis, by training 11 deep learning models on seven MVTS datasets (114 entities) and five repeats, resulting in 6270 deep learning models and 4273 end to end experiments. We showed through experiments that existing evaluation metrics in use for event-wise time-series anomaly detection are not adequate and can be misleading. To remedy this, we proposed a new metric for event-wise time-series anomaly detection, the composite F-score, Fc_1, which is the harmonic mean of event-wise recall and point-wise precision. Unlike previous studies, we studied the effect of models and scoring functions independently to gain a deeper understanding of what makes a good time-series anomaly detection algorithm. We found that the choice of the scoring function can have a large impact on anomaly detection performance, and dynamic scoring functions Gauss-D and Gauss-D-K work better than the static scoring function, Gauss-S. Surprisingly and significantly, we found that the top performing model in our evaluation for anomaly detection and the diagnosis was the UAE model with the Gauss-D scoring function.

While the good performance of UAE could be partly due to the prevalence of temporal anomalies, our study makes it clear that recently proposed deep algorithms [4], [5], [9], [10], [13] fail to effectively detect even these simple [49] anomalies in MVTS datasets. The UAE model would be a good starting point for anomaly detection in a steady-state CPS dataset, but might not perform as well on a system with multiple operating states. Possible routes to further improve upon the performance of UAE may be through the use of improved channel-wise models and through ensembling the channel-wise reconstructions with reconstructions from a complementary model that accounts for only cross-channel effects, similar to the idea used by Zhang *et al.* [22]. Our study also highlights the need for more challenging CPS datasets that display multiple operating conditions and where both temporal and cross-channel anomalies are observed.

Overall, our work provides important insights into the design and evaluation of methods for anomaly detection and diagnosis, and serves as a useful guide for future method development.

References

[1] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019, *arXiv:1901.03407*. [Online]. Available: http://arxiv.org/abs/1901.03407

[2] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection," *ACM Comput. Surveys*, vol. 54, no. 2, pp. 1–38, Apr. 2021, doi: 10.1145/3439950.

[3] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks," in *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, 2019, pp. 703–716.

[4] B. Zhou, S. Liu, B. Hooi, X. Cheng, and J. Ye, "BeatGAN: Anomalous rhythm detection using adversarially generated time series," in *Proc. 28th Int. Joint Conf. Artif. Intell.* Menlo Park, CA, USA: AAAI Press, 2019, pp. 4433–4439.

[5] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 387–395.

[6] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 1544–1551, Jul. 2018.

[7] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based encoder-decoder for multi-sensor anomaly detection," 2016, *arXiv:1607.00148*. [Online]. Available: http://arxiv.org/abs/1607.00148

[8] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "USAD: Unsupervised anomaly detection on multivariate time series," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2020, pp. 3395–3404.

[9] C. Zhang *et al.*, "A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 1409–1416.

[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. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 2828–2837.

[11] S. Tariq *et al.*, "Detecting anomalies in space using multivariate convolutional LSTM with mixtures of probabilistic PCA," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 2123–2133.

[12] A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, and S. Robinson, "Deep learning for unsupervised insider threat detection in structured cybersecurity data streams," 2017, *arXiv:1710.00811*. [Online]. Available: https://arxiv.org/abs/1710.00811

[13] P. Zheng, S. Yuan, X. Wu, J. Li, and A. Lu, "One-class adversarial nets for fraud detection," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, pp. 1286–1293, Jul. 2019.

[14] H. Ren *et al.*, "Time-series anomaly detection service at microsoft," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 3009–3017.

[15] S. Hawkins, H. He, G. Williams, and R. Baxter, "Outlier detection using replicator neural networks," in *Proc. Int. Conf. Data Warehousing Knowl. Discovery*. Berlin, Germany: Springer, 2002, pp. 170–180.

[16] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, Jul. 2009.

[17] H. Xu *et al.*, "Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications," in *Proc. World Wide Web Conf.*, 2018, pp. 187–196.

[18] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, "A review on outlier/anomaly detection in time series data," 2020, *arXiv:2002.04236*. [Online]. Available: http://arxiv.org/abs/2002.04236

[19] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, Nov. 2017.

[20] J. Hochenbaum, O. S. Vallis, and A. Kejariwal, "Automatic anomaly detection in the cloud via statistical learning," 2017, *arXiv:1704.07706*. [Online]. Available: http://arxiv.org/abs/1704.07706

[21] N. Laptev, S. Amizadeh, and I. Flint, "Generic and scalable framework for automated time-series anomaly detection," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 1939–1947.
[22] Y. Zhang, Z. Y. Dong, W. Kong, and K. Meng, "A composite anomaly detection system for data-driven power plant condition monitoring," *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4390–4402, Jul. 2020.
[23] T. Kieu, B. Yang, C. Guo, and C. S. Jensen, "Outlier detection for time series with recurrent autoencoder ensembles," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 2725–2732.
[24] L. Feremans, V. Vercruyssen, B. Cule, W. Meert, and B. Goethals, "Pattern-based anomaly detection in mixed-type time series," in *Machine Learning and Knowledge Discovery in Databases*. Cham, Switzerland: Springer, 2020, pp. 240–256.
[25] C.-C.-M. Yeh *et al.*, "Matrix profile I: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 1317–1322.
[26] E. Keogh, J. Lin, and A. Fu, "HOT SAX: Efficiently finding the most unusual time series subsequence," in *Proc. 5th IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2005, p. 8.
[27] X. Wang, J. Lin, N. Patel, and M. Braun, "Exact variable-length anomaly detection algorithm for univariate and multivariate time series," *Data Mining Knowl. Discovery*, vol. 32, no. 6, pp. 1806–1844, 2018.
[28] A. Siffer, P.-A. Fouque, A. Termier, and C. Largouet, "Anomaly detection in streams with extreme value theory," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*. New York, NY, USA: Association for Computing Machinery, Aug. 2017, pp. 1067–1075, doi: 10.1145/3097983.3098144.
[29] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part I: Quantitative model-based methods," *Comput. Chem. Eng.*, vol. 27, no. 3, pp. 293–311, 2003.
[30] S. J. Qin, "Survey on data-driven industrial process monitoring and diagnosis," *Annu. Rev. Control*, vol. 36, no. 2, pp. 220–234, Dec. 2012.
[31] Y. Zhao, Z. Nasrullah, and Z. Li, "PyOD: A Python toolbox for scalable outlier detection," *J. Mach. Learn. Res.*, vol. 20, no. 96, pp. 1–7, Jan. 2019. [Online]. Available: http://jmlr.org/papers/v20/19-011.html
[32] G. M. de Almeida and S. W. Park, "Fault detection and diagnosis in the DAMADICS benchmark actuator system—A hidden Markov model approach," *IFAC Proc. Volumes*, vol. 41, no. 2, pp. 12419–12424, 2008.
[33] R.-Q. Chen, G.-H. Shi, W.-L. Zhao, and C.-H. Liang, "A joint model for IT operation series prediction and anomaly detection," *Neurocomputing*, vol. 448, pp. 130–139, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231221004483, doi: 10.1016/j.neucom.2021.03.062.
[34] B. Zong *et al.*, "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," in *Proc. ICLR*, 2018, pp. 1–19.
[35] S. Li and J. Wen, "A model-based fault detection and diagnostic methodology based on PCA method and wavelet transform," *Energy Buildings*, vol. 68, pp. 63–71, Jan. 2014.
[36] L. Manevitz and M. Yousef, "One-class SVMs for document classification," *J. Mach. Learn. Res.*, vol. 2, pp. 139–154, Dec. 2002.
[37] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in *Proc. Int. Conf. Crit. Inf. Infrastruct. Secur.* Cham, Switzerland: Springer, 2016, pp. 88–99.
[38] C. M. Ahmed, V. R. Palleti, and A. P. Mathur, "WADI: A water distribution testbed for research in the design of secure cyber physical systems," in *Proc. 3rd Int. Workshop Cyber-Phys. Syst. Smart Water Netw.*, 2017, pp. 25–28.
[39] (2020). *Damadics Benchmark Website*. [Online]. Available: http://diag.mchtr.pw.edu.pl/damadics/
[40] I. D. Katser and V. O. Kozitsin. (2020). *Skoltech Anomaly Benchmark (SKAB)*. [Online]. Available: https://www.kaggle.com/dsv/1693952
[41] Q. Yu, L. Jibin, and L. Jiang, "An improved ARIMA-based traffic anomaly detection algorithm for wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 2016, pp. 1–9, Jan. 2016.
[42] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*. [Online]. Available: http://arxiv.org/abs/1803.01271
[43] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning Internal Representations by Error Propagation*. Cambridge, MA, USA: MIT Press, 1986, pp. 318–362.
[44] C. Doersch, "Tutorial on variational autoencoders," 2016, *arXiv:1606.05908*. [Online]. Available: http://arxiv.org/abs/1606.05908
[45] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, "DeepAnT: A deep learning approach for unsupervised anomaly detection in time series," *IEEE Access*, vol. 7, pp. 1991–2005, 2019.
[46] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.
[47] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Amer. Stat. Assoc.*, vol. 32, no. 200, pp. 675–701, 1937.
[48] Y. Hochberg, "A sharper Bonferroni procedure for multiple tests of significance," *Biometrika*, vol. 75, no. 4, pp. 800–802, Dec. 1988.
[49] R. Wu and E. J. Keogh, "Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress," 2020, *arXiv:2009.13807*. [Online]. Available: https://arxiv.org/abs/2009.13807

Astha Garg received the Ph.D. degree in chemical engineering from Pennsylvania State University, State College, PA, USA, in 2017.

Her experience spans semiconductor equipment manufacturing (Applied Materials, USA), data-driven chemicals and materials research (Citrine Informatics, USA) and marine equipment monitoring (ChordX Pte. Ltd., current), as well as research on time-series (Institute for Infocomm Research (I2R), A*STAR, Singapore). Her research focuses on applications of machine learning to Industry 4.0, and data-driven experimental design.

Wenyu Zhang received the Ph.D. degree in statistics from Cornell University, Ithaca, NY, USA, in 2020.

She is currently a Scientist at I2R (Institute for Infocomm Research) working in areas of machine learning and time series analysis.

Jules Samaran is pursuing an integrated undergraduate and master's degree at Mines Paristech and PSL University, Paris, France, as well as Ecole Normale Superieure de Saclay, Paris.

His research interests lie at the intersection of statistics and machine learning.

Ramasamy Savitha (Senior Member, IEEE) received the Ph.D. degree from Nanyang Technological University, Singapore, in 2011.

Currently, she is a Research Group Leader at the Institute for Infocomm Research, A*STAR, Singapore. She has published about 100 papers in various international conferences and journals, along with a research monograph published by Springer-Verlag, Germany. Her research interests are in developing robust AI, with special focus on lifelong learning and time series data analysis, and has led translation of these robust models for predictive analytics in real-world applications. Her contributions to data analysis and AI have been recognized in the inaugural 100 SG Women in Technology list.

Chuan-Sheng Foo received the B.S., M.S., and Ph.D. degrees from Stanford University, Stanford, CA, USA, in 2008, 2012, and 2017, respectively.

He currently leads a research group at the Institute for Infocomm Research, A*STAR, Singapore, which focuses on developing data-efficient deep learning algorithms that can learn from less labeled data.