Mr.TAD: Time Series Anomaly Detection via Multiresolution Ensemble Learning

Junhyeok Kang, Patara Trirat, Youngeun Nam, Taeyoon Kim
Korea Advanced Institute of Science and Technology
Daejeon, Republic of Korea
{junhyeok.kang, patara.t, youngeun.nam, tykimseoul}@kaist.ac.kr

Abstract

Time series anomaly detection is the task of finding abnormal patterns in sequential data, which have complex underlying temporal dynamics. Multi-resolution learning is the crucial factor for capturing such inherent characteristics of time series. Recent studies have tackled the challenges of multi-resolution learning with a single joint representation from different resolution levels. In this paper, we propose Mr.TAD*, a sequence-to-sequence autoencoder for time series anomaly detection.* Mr.TAD *not only learns the inter-resolution features but also additionally learns intra-resolution features, which are unique characteristics of time series for each resolution. We adopt ensemble learning with skip connections across different resolutions. Through extensive experiments on various real-world time series benchmarks, we show that* Mr.TAD *can surpass several baselines proving the effectiveness of the multi-resolution ensemble learning.*

1. Introduction

Anomaly detection in time series is the task of finding points or sequences that are anomalous or deviated from normal patterns in sequential or stream data. It is a crucial task as it can be utilized in many real-world applications (e.g., electrocardiogram, fraud detection, and smart factory monitoring) [5]. Building an effective time series anomaly detector is challenging because time series exhibit complex patterns formed by a combination of trend, seasonality, and noise. Classical approaches can be categorized as clustering[12], density-based[14], distance-based[3], or isolation-based[13]. These algorithms find anomalies based on their heuristics. However, these classical methods does not provide satisfactory performances. Thanks to the emergence of deep learning, numerous novel approaches in time series anomaly detection [21, 6, 22, 10, 19] mostly adopted deep learning methods to learn latent representations from time-series data while addressing the above challenge.

In addition, multi-resolution learning in time series has recently brought great attention for capturing complex temporal features of time series in multiple scales [18, 17]. However, discovering multi-resolution features from a real-world time series is still non-trivial because of its inherently complex characteristics. To understand the complexity of multi-resolution features, we propose *Mr.TAD*, a new variant of the sequence-to-sequence ensemble autoencoder. First, *Mr.TAD* models both *intra-resolution* features which represent intrinsic characteristics of each resolution and *inter-resolution* features which can guide the common information shared between different resolutions. Each of the encoder and decoder blocks consists of recurrent skip connection networks (RSCNs) [10], that adopt additional auxiliary connections among the RNN units. Reconstructed time series that have high reconstruction errors above a predefined threshold are classified as anomaly points.

Our main contributions are summarized as follows.

- We propose a novel framework, *Mr.TAD*, that captures both *intra-* and *inter-resolution* latent representations.
- We assemble models based on a recurrent autoencoder with skip connections across different resolutions.
- We conduct extensive experiments on popular benchmarks with a set of ablations studies and compare our *Mr.TAD* with both machine learning-based algorithms and deep learning-based methods.

2. Related Work

2.1. Deep Learning Approaches

Time series anomaly detection arises from various domains, but the labels are often hard to come by. Also, because time series data are mostly composed of normal patterns, unsupervised learning can readily be adopted. A recent study called DONUT [21] proposed a VAE-based model with three new techniques: modified Evidence Lower Bound (ELBO), missing data injection in training, and a Markov Chain Monte Carlo (MCMC) imputation in detection in order to achieve better performance. BeatGAN [25] is a reconstructive approach based on a Generative Adversarial Network (GAN) for detecting anomalous time series. It used time-series warping for data augmentation to improve accuracy and demonstrated their effectiveness in fast inference. Another method named MSCRED [24] proposed a novel architecture of attention-based ConvLSTM to take temporal dependency into account. OmniAnomaly [19] and USAD [1] also adopted reconstruction-based model architectures. However, these latest works do not consider the temporal resolution to varying degrees.

2.2. Learning with Multi-Resolution

There have been a few approaches that extracted temporal features at multiple time scales. Temporal Hierarchical One-Class (THOC) network fuses and processes multi-resolution features through a hierarchical network with dilated recurrent neural networks (RNN) [17]. Lately, recurrent autoencoder with multi-resolution ensemble decoding (RAMED) uses decoders of different decoding lengths and proposes a coarse-to-fine fusion mechanism for computing multi-resolution outputs[18]. Both methods adopted a multi-resolution feature extraction technique and a fusion mechanism to capture the temporal dynamics across resolutions of varying scales.

3. The *Mr.TAD* Framework

In this section, we describe our proposed algorithm *Mr.TAD* which captures both *intra-* and *inter*-resolution features. First, we introduce the overall architecture of *Mr.TAD* and explain the details of the encoding and decoding parts.

3.1. Problem Statement

A time series $\mathcal{T} = \langle s_1, s_2, \cdots, s_T \rangle$ is a series of data points indexed chronologically. Each data point $s_t = \{x_t^1, x_t^2 \cdots x_t^d\}$ represents d features of an entity at a specific timestamp t_i. The time series anomaly detection model aims to compute an anomaly score for each data point in a given sequence. A series s at time t denoted by s_t is classified as an anomaly if the anomaly score exceeds a predefined threshold.

3.2. Architecture

Figure 1 depicts the overall architecture of our *Mr.TAD*, which is based on the ensemble sequence-to-sequence autoencoder of different resolutions to learn multi-resolution features. A latent vector $\mathbf{h}^{(E*)}$ which represents the common information between different resolutions is shared by the encoders. Thus, $\mathbf{h}^{(E*)}$ helps the model to learn *inter-resolution* features from various resolutions. For each resolution k, there exists a hidden state $\mathbf{h}^{(E_k)}$. It addresses our claim that each resolution has its own unique characteristics and helps the model to learn the *intra-resolution* features.

3.3. Multi-resolution with Skip RNN Autoencoders

We adopt the idea of recurrent skip connection networks (RSCNs) [22] to learn the latent representations from encoders and decoders of each resolution. As shown in Figure 2, we make sparse skip connections to learn the lower resolution features and dense skip connections for the higher resolution features. Because RSCNs use sparseness weight vector w_t that decides which connections should be connected at each time step t, we make the skip connection according to the resolution level.

3.3.1 Encoder

We employed a recurrent neural network (RNN) to encode time series data. Given $\mathcal{T}$ with time series length T, the hidden state $\mathbf{h}$ of the encoder at time t is computed as

$$\mathbf{h}_t^{(E)} = f^{(E)}([s_t; \mathbf{h}_{t-1}^{(E)}]), \tag{1}$$

where $\mathbf{h}_{t-1}^{(E)}$ is the previous state and $f^{(E)}$ is a nonlinear activation function.

In the encoder, the time series $\mathcal{T}$ is fed into each of the RNN units (here, GRU is selected as it performs well in preliminary experiments) from different resolutions. The the hidden state $\mathbf{h}_t$ is computed as Equation 2:

$$\mathbf{h}_t^{(E)} = f^{(E)}\left(\left[s_t; \frac{w_1 \mathbf{h}_{t-1}^{(E)} + w_2 \mathbf{h}_{t-L}^{(E)}}{|w_1| + |w_2|}\right]\right), \tag{2}$$

where w_1 and w_2 denote the random samples from $\{(1,0),(0,1),(1,1)\}$ at each time step. We use L as the skip length formulated by 2^{k-1} for k^{th} resolution. Furthermore, the k^{th} resolution is represented as Equation 3, which is the concatenated result of *intra-* and *inter*-resolution features. The hidden state of the k^{th} resolution is defined as

$$\mathbf{h}^k = f_{MLP}(\mathit{concat}[\mathbf{h}^{(E_k)}, \mathbf{h}^{(E*)}]), \tag{3}$$

where $\mathbf{h}^{(E_k)}$ denotes the hidden state of *intra-resolution* and $\mathbf{h}^{(E*)}$ denotes the hidden state of *inter-resolution* calculated as

$$\mathbf{h}^{(E*)} = f_{MLP}(\mathit{concat}[\mathbf{h}^{(E_1)}, \mathbf{h}^{(E_2)} \cdots, \mathbf{h}^{(E_k)}]). \tag{4}$$

The *inter-resolution* is responsible for capturing the global features of multiple resolutions through the shared latent vectors.

3.3.2 Decoder

The compressed representation of the encoders $\mathbf{h}_T^{(E)}$ is fed into the decoder using a GRU. The outputs (i.e., reconstructed time series) of the decoder is generally processed in a reverse chronological order in time series anomaly detection [10, 22] since it is proved to be more efficient.

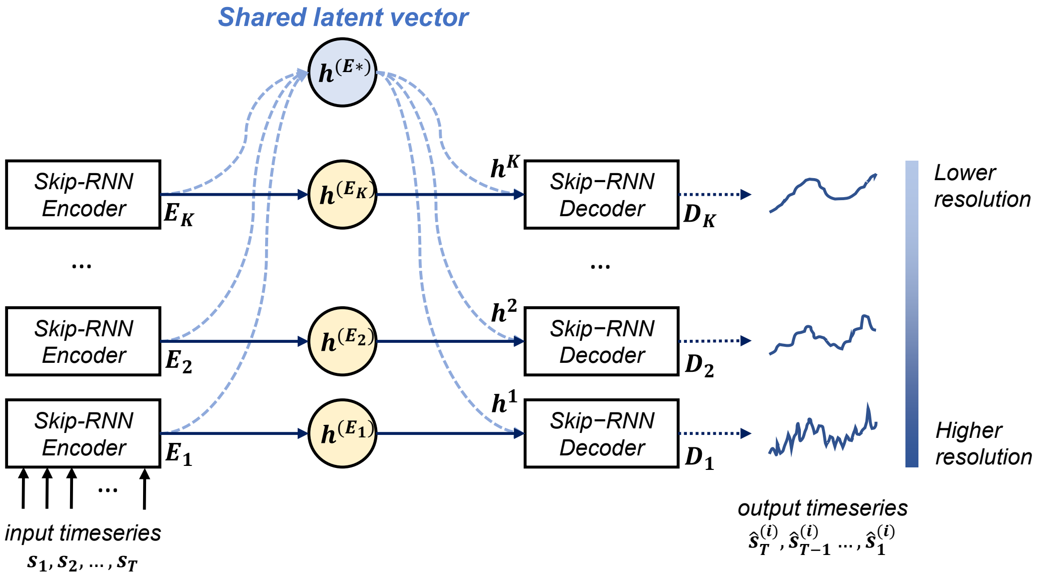
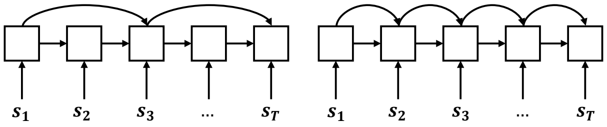


Figure 1. Overall architecture of *Mr.TAD*.



(a) Lower resolution (b) Higher resolution

Figure 2. Skip RNN Encoder and Decoder in *Mr.TAD*. As the resolution goes down, it connects more sparse skip connections.

Therefore, the reconstructed decoder output is expressed as $\hat{\mathcal{T}}^{(i)} = \langle \hat{s}_T^{(i)}, \hat{s}_{T-1}^{(i)}, \cdots, \hat{s}_1^{(i)} \rangle$ for input $\mathcal{T}$, where $1 \le i \le K$. Based on the previous hidden state of the decoder $\mathbf{h}_{t+1}^{(D)}$ and the previous reconstructed vector $\hat{s}_{t+1}$, the decoder output is calculated using Equation 5.

$$\hat{s}_t = \mathbf{W}\mathbf{h}_t^{(D)} + \mathbf{b}, \tag{5}$$

where $\mathbf{W}, \mathbf{b}$ are learnable parameters. Also, the current hidden state in decoder is calculated using Equation 6,

$$\mathbf{h}_t^{(D)} = GRU([\hat{s}_t; \mathbf{h}_{t+1}^{(D)}]). \tag{6}$$

3.4. Loss function

To optimize the multi-resolution ensemble learning framework, we integrate errors from all resolutions and summarize them. Formally, we have

$$\begin{aligned} \mathcal{J} &= \sum_{i=1}^{K} \mathcal{J}_i + \lambda \left\| \mathbf{h}^{(E)} \right\|_2 \\ &= \sum_{i=1}^{K} \sum_{t=1}^{T} \left\| s_t - \hat{s}_t^{(D_i)} \right\|_2^2 + \lambda \left\| \mathbf{h}^{(E)} \right\|_2, \end{aligned} \tag{7}$$

where λ is the weight of the L2 regularization and $\mathcal{J}_i = \sum_{t=1}^{T} \| s_t - \hat{s}_t^{(D_i)} \|_2^2$ is the reconstruction error for the i^{th} resolution.

3.5. Anomaly Scoring

When using the ensemble framework, we calculate the anomaly score of each vector in time series using the concept of autoencoder ensembles for non-sequential data[4]. Let $\mathcal{T} = \langle s_1, s_2, \cdots, s_T \rangle$ be a time-ordered sequence of vectors. For each vector $s_t = \{x_t^1, x_t^2 \cdots x_t^d\}$, the anomaly score $AS(s_i)$ is calculated. The higher the $AS(s_i)$ the more likely s_i is an anomaly.

We obtain K reconstructed time series $\hat{\mathcal{T}}^{(i)} = \langle \hat{s}_T^{(i)}, \hat{s}_{T-1}^{(i)}, \cdots, \hat{s}_1^{(i)} \rangle$. For each vector s_t in the original time series $\mathcal{T}$, we compute K reconstruction errors $\{\|s_t - \hat{s}_t^{(1)}\|_2^2, \|s_t - \hat{s}_t^{(2)}\|_2^2, \cdots, \|s_t - \hat{s}_t^{(K)}\|_2^2\}$. Finally, we use the median of the K reconstruction errors as the anomaly score $AS(s_k) = (\|s_t - \hat{s}_t^{(1)}\|_2^2, \|s_t - \hat{s}_t^{(2)}\|_2^2, \cdots, \|s_t - \hat{s}_t^{(K)}\|_2^2)$. To prevent the autoencoder from overfitting to the original time series, the median is used instead of the mean [10].

4. Experiments

In this section, we demonstrate the performance of *Mr.TAD* evaluated by common time-series benchmarks and compare *Mr.TAD* against existing anomaly detection algorithms. The source code is available at https://github.com/itouchz/Mr.TAD.

4.1. Datasets

We adopted *six* publicly available datasets, including sub-datasets. Table 1 summarizes the characteristics of

Table 1. Dataset Information

Datasets		Dim.	# Training	# Testing	% Anomaly
Yahoo S5	A1	1	55,790	23,883	4.39
	A2		72,400	69,700	0.67
	A3		108,192	46,368	0.65
	A4		96,432	41,328	0.58
NASA	SMAP	25	140,825	444,035	12.85
	MSL	55	58,317	73,729	10.48
SMD		38	708,405	708,420	4.16
ECG	A	2	1,833	1,841	14.61
	B		2,439	1,287	12.35
	C		10,863	3,348	4.45
	D		2,610	1,121	11.51
	E		2,011	1,447	9.61
	F		2,943	2,255	8.38
	G		34,735	9,882	2.01
	H		2,373	2,721	9.52
	I		3,152	1,756	21.58
Power Demand		1	1,513	1,596	13.22
2D Gesture		2	8,451	2,800	26.04

them. For datasets that do not have a predefined train-test split, we select 70% of the dataset for training and 30% for testing. Further, 30% of the training set is used for validation.

Yahoo S5 [1]. This dataset consists of real and synthetic time series with labeled anomaly points. The dataset tests the detection of various anomaly types, including outliers and change-points. The synthetic data contains time series with varying trend, noise, and seasonality while the real data represents the metrics of various Yahoo services.

Power Demand [9]. The Power Demand dataset contains one year of power consumption records measured by a Dutch research facility in 1997.

ECG [9]. The ECG dataset contains anomalous beats from electrocardiograms. It has nine sub-datasets.

2D Gesture [9]. The 2D Gesture dataset contains time series of X-Y coordinates of an actor's right hand. The data is extracted from a video in which the actor grabs a gun from his hip-mounted holster, moves it to the target, and returns it to the holster. The anomalous region is in the area where the actor fails to return his gun to the holster.

NASA Dataset [8]. The NASA dataset consists of Soil Moisture Active Passive (SMAP) satellite and Mars Science Laboratory (MSL) rover datasets. SMAP and MSL are two real-world public datasets, labeled by experts in NASA. They contain data of 55/27 entities each monitored by $m = 25/55$ metrics, respectively.

Server Machine Dataset (SMD) [19]. SMD is a new 5-week-long dataset from a large Internet company collected and made publicly available. It contains data from 28 server machines each monitored by $m = 38$ metrics.

4.2. Comparison Baselines

We compare *Mr.TAD* against the following competitive baselines, including *3* traditional methods and *3* deep learning-based models with their variants.

Traditional Methods. (1) Local Outlier Factor (LOF) [2] is a popular density-based outlier detection method. (2) Isolation Forest (IF) [13] is an unsupervised learning algorithm catching up with randomized clustering forest. (3) One-Class Support Vector Machines (SVM) [16] is also an unsupervised learning algorithm based on a kernel-based method for outlier detection.

Deep Learning-based Methods. (1) Auto-Encoder (AE) [15] is a generative unsupervised deep learning algorithm used for reconstructing high dimensional input data using a neural network with a bottleneck latent layer which contains the compressed representation of the input data. (2) Variational Auto-Encoder (VAE) [23] is a modified reconstruction model to prevent the model from reconstructing abnormal samples well. By adding a constraint network in the latent space, the model is forced to generate new latent variables similar to that of training samples. (3) Generative Adversarial Network (GAN) [7] is composed of the generator, which compresses and decompresses the input to generate a time series, and the discriminator, which tries to distinguish whether the time series is normal or not. (4) Sparsely-connected RNNs with Shared Framework (S-RNNS SF) [10] is the ensemble network with sparsely connected RNNs and randomly removed connections. Moreover, the shared framework is employed to enable interactions among the autoencoders. (5) Dilated LSTM AE is our baseline for the manual structure of *Mr.TAD*. During preprocessing for time-series, a multi-resolution skip connection is implemented.

Variations of the AE-based models are CNN, LSTM, GRU, and Bi-directional GRU, while for the others, CNN, LSTM, and GRU are applied.

4.3. Evaluation Metrics

Following the common practices adopted by recent studies [21, 19, 17, 18], a point-adjust approach, we adopt Precision (P), Recall (R), best F1 score (F1), and *AUC* as our evaluation metrics.

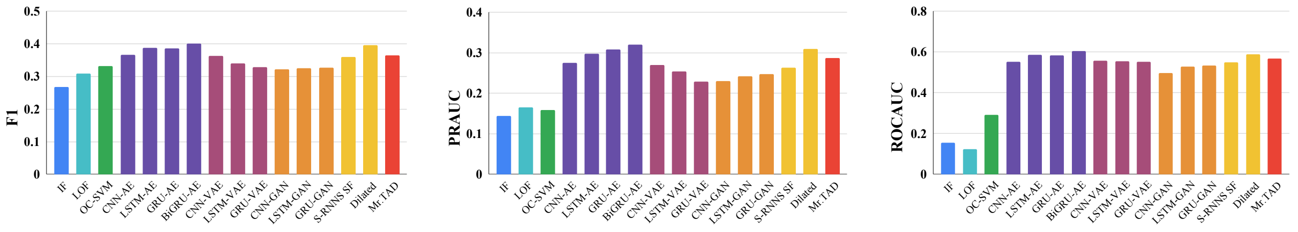
$$P = \frac{TP}{TP + FP},\ R = \frac{TP}{TP + FN},\ F1 = 2 \cdot \frac{P \cdot R}{P + R},$$

where TP is True Positives, FP is False Positives, and FN is False Negatives. Also, we evaluate the area under the precision-recall curve (PRAUC), which represents the average of precision scores calculated for each recall threshold and the area under the ROC curve (ROCAUC), which represents the trade-off between the true positive rate and the false positive rate. Finally, all metrics are computed with

[1] https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70

Table 2. Overall Performance Comparison

	Yahoo S5			NASA			SMD			ECG			Power Demand			2D Gesture		
	F1	**PRAUC**	**ROCAUC**	**F1**	**PRAUC**	**ROCAUC**	**F1**	**PRAUC**	**ROCAUC**	**F1**	**PRAUC**	**ROCAUC**	**F1**	**PRAUC**	**ROCAUC**	**F1**	**PRAUC**	**ROCAUC**
IF	0.3226	0.0913	0.2494	0.0440	0.0048	0.0017	0.3199	0.2468	0.1788	0.2269	0.1003	0.0855	0.1826	0.0323	0.0566	0.5091	0.3883	0.3604
LOF	0.5224	0.2364	0.1934	0.2256	0.0427	0.0350	0.3354	0.2640	0.3674	0.2819	0.1897	0.0718	0.0323	0.0008	0.0006	0.4533	0.2553	0.0785
OC-SVM	0.4667	0.1243	0.4889	0.1857	0.0473	0.0578	0.3348	0.1621	0.5072	0.2545	0.1488	0.2435	0.2451	0.0720	0.1301	0.5072	0.3946	0.3199
CNN-AE	0.6146	0.3557	0.7839	0.2259	0.3218	0.5047	0.4206	0.4185	0.7069	0.2483	0.1481	0.5260	0.2691	0.0996	0.3295	0.4132	0.3075	0.4596
LSTM-AE	0.6306	0.3695	0.7838	0.2123	0.3052	0.4916	0.4483	0.4674	0.7277	0.3011	0.2129	0.5636	0.2632	0.1046	0.3477	0.4699	0.3260	0.5948
GRU-AE	0.6509	0.3870	0.8103	0.2123	0.3141	0.4929	0.4469	0.4780	0.7215	0.2821	0.2065	0.5873	0.2816	0.1012	0.3259	0.4435	0.3640	0.5666
BiGRU-AE	0.7680	0.4867	0.8735	0.2123	0.3135	0.4953	0.4430	0.4817	0.7110	0.2762	0.1948	0.5768	0.2643	0.1205	0.3998	0.4401	0.3259	0.5665
CNN-VAE	0.5351	0.3063	0.7292	0.2131	0.3145	0.4567	0.4379	0.4445	0.7376	0.2688	0.1756	0.5482	0.2679	0.0953	0.2934	0.4548	0.2815	0.5773
LSTM-VAE	0.4383	0.2332	0.6749	0.2123	0.3146	0.4571	0.4420	0.4388	0.7508	0.2598	0.1725	0.5539	0.2643	0.1059	0.3646	0.4262	0.2573	0.5188
GRU-VAE	0.3822	0.2074	0.6147	0.2123	0.2003	0.4757	0.4426	0.4377	0.7517	0.2509	0.1710	0.5568	0.2643	0.1152	0.4110	0.4167	0.2446	0.4980
CNN-GAN	0.3906	0.1791	0.4934	0.2123	0.3127	0.4596	0.4078	0.3991	0.7005	0.2444	0.1583	0.5314	0.2632	0.0989	0.3181	0.4132	0.2318	0.4680
LSTM-GAN	0.2001	0.0819	0.2514	0.2123	0.1501	0.4731	0.4297	0.4188	0.7047	0.2153	0.1351	0.4983	0.4524	0.3270	0.6852	0.4444	0.3424	0.5506
GRU-GAN	0.2318	0.0957	0.2751	0.2123	0.2347	0.4434	0.4147	0.4020	0.6942	0.2735	0.1819	0.6097	0.4167	0.3270	0.6852	0.4132	0.2402	0.4904
S-RNNS SF	0.4712	0.2508	0.6407	0.2123	0.3157	0.4635	0.4492	0.4472	0.7477	0.2802	0.1831	0.5527	0.2749	0.0960	0.3189	0.4640	0.2836	0.5646
Dilated LSTM AE	0.6542	0.3933	0.8031	0.2123	0.3145	0.4943	0.4579	0.4589	0.7435	0.2901	0.2055	0.5420	0.2802	0.0990	0.3051	0.4770	0.3831	0.6374
Mr.TAD	0.5000	0.2679	0.6635	0.2123	0.3144	0.4900	0.4553	0.4632	0.7411	0.2710	0.1842	0.5572	0.2755	0.0968	0.3152	0.4759	0.3952	0.6329



(a) F1

(b) PRAUC

(c) ROCAUC

Figure 3. Average performance comparison in three metrics.

1000 thresholds generated uniformly from 0 to the maximum anomaly score over all time steps in the test set [22].

4.4. Implementation Details

As in the recent study [18], we use three encoders and three decoders. Each encoder and decoder is a single-layer GRU with 64 units. Due to the time limitation, we do not perform any hyperparameter tuning. All hyperparameter settings were taken from relevant prior studies [17, 18, 10], except for the sliding window size and the stride length. In this work, we set the sliding window with the following lengths: 16 for *Power Demand*, 8 for *Yahoo S5*, *NASA*, *SMD*, and 4 for *ECG*, *2D Gesture*. The stride length is half of the sliding window size. We need to set the very short sequence length due to the limited time and resources.

We implement all algorithms in Python 3.7. The machine learning methods (i.e., LOF, IF, and OC-SVM) are implemented using scikit-learn 0.24.2. We implement deep learning-based methods including ours using TensorFlow 2.4. The Adam optimizer [11] is used with an initial learning rate of 0.001. Early stopping and learning rate decay are also adopted to avoid overfitting. We set λ to 0.005. *Mr.TAD* and all baselines are trained on the same platform[2].

Table 3. Effect of Intra- and Inter-Resolution

	F1	**PRAUC**	**ROCAUC**
w/o Intra-resolution	0.3244	0.2220	0.5425
w/o Inter-resolution	0.3158	0.2122	0.5307
Mr.TAD	0.3372	0.2396	0.5744

Table 4. Effect of Skip Length Strategy

	F1	**PRAUC**	**ROCAUC**
linear skip connection	0.3356	0.2367	0.5720
random skip connection	0.3340	0.2348	0.5681
Mr.TAD	0.3372	0.2396	0.5744

Table 5. Effect of Weighting Strategy

	F1	**PRAUC**	**ROCAUC**
w/o weights	0.3343	0.2343	0.5648
w/ uniform weight	0.3380	0.2327	0.5661
Mr.TAD	0.3372	0.2396	0.5744

Table 6. Effect of Regularization Techniques

	F1	**PRAUC**	**ROCAUC**
w/o regularization	0.3360	0.2395	0.5762
w/ L1	0.3369	0.2391	0.5709
Mr.TAD	0.3372	0.2396	0.5744

[2] Ubuntu 18.04 LTS with an NVIDIA GeForce RTX 2080 Ti GPU

4.5. Experimental Results

4.5.1 Performance Comparison

We report the performance of all models grouped by each dataset in Table 2 on all metrics and visualize the global averaged performance grouped by each metric in Figure 3. In general, all methods give similar *F1* scores, even the traditional models. It is possible because any model can achieve a reasonably good result, given 1000 thresholds. However, the deep learning-based methods perform better when measured with *PRAUC* and *ROCAUC* in all cases with a significant margin. It indicates that the deep learning-based models have lower false positive rates and higher true negative rates, which is highly desired for anomaly detection. Besides, it is observable that AE-based methods outperform both VAEs and GANs, especially with bidirectional learning. Still, according to Table 2, 7, 8, and 9, some VAE-based and GAN-based models can achieve higher than AEs. It is probably because specific methods may favor particular patterns over others. Therefore, developing a model that can overcome this issue can be a promising research direction. Concerning our *Mr.TAD*, it does not outperform all baselines in all datasets yet obtains reasonable results. We suspect that it results from the short sequence length that degrades the performance of our method, similar to the S-RNN SF. However, our proposed method will achieve better results with the longer sequence length, i.e., window size. Additionally, for Dilated LSTM AE, this model still attains adequate performance because the skip connection is made in the input level; thus, there is no accumulated loss between the hidden state of each timestamp.

4.5.2 Ablation Study

To examine the contributions of each design choice, we perform ablation studies on the following aspects.

Effect of intra- and inter-resolution learning. In this experiment, we use the identical architectures as the full model but only remove one latent vector at a time. As presented in Table 3, it is evident that using both latent representation learning results in better performance. Obviously, when removing inter-resolution (i.e., joint representation), the performance decreases significantly. These results are also comparable to the previous study [10].

Effect of skip length strategy. In this experiment, we study the difference between each skip length strategy by simply changing the 2^{k-1} to k for a linear case or a random value between 2 and T. The results, presented in Table 4, indicate minor differences between each strategy. Thus, we believe that the model can still perform multi-resolution learning, given the same number of autoencoders.

Effect of weighting strategy. We study the effect of the weights w_1 and w_2 in this experiment. For the first case, we simply remove all the weights so that the input of the next hidden state is the addition of the previous hidden state and the L-th hidden state before. For the second case, we set the $w_1 = w_2 = 1$. Table 5 presents the results from this experiment. We observe that the model performs better with weight control, either by uniformly set or randomly selecting.

Effect of regularization. In [10], the authors suggest that L1 regularization has the effect of making the shared hidden state $\mathbf{h}_T^{(E)}$ sparse, resulting in more robust decoders. However, we suspect whether it is the case in our method. Accordingly, we first remove the regularizer. Then, we try with L1 as suggested. However, we see that using L2 regularization achieves better performance on average. Therefore, we decide to include the L2 regularizer with *Mr.TAD* instead of the L1.

5. Conclusion

In this paper, we introduce a sequence-to-sequence-based network called *Mr.TAD* for time series anomaly detection. *Mr.TAD* learns *inter-resolution* features representing the internal characteristic within each resolution, after which the learned representations are shared for all resolution levels. Moreover, it learns *intra-resolution* features representing the characteristics of each resolution through skip connections across different resolutions. Through extensive experiments on commonly used benchmarks, we show that *Mr.TAD* outperforms classical algorithms and is comparable to deep learning-based methods.

References

[1] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3395–3404, 2020. 2

[2] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000. 4

[3] Wanpracha Art Chaovalitwongse, Ya-Ju Fan, and Rajesh C Sachdeo. On the time series k-nearest neighbor classification of abnormal brain activity. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 37(6):1005–1016, 2007. 1

[4] Jinghui Chen, Saket Sathe, Charu Aggarwal, and Deepak Turaga. Outlier detection with autoencoder ensembles. In *Proceedings of the 2017 SIAM international conference on data mining*, pages 90–98. SIAM, 2017. 3

[5] Chih-Chun Chia and Zeeshan Syed. Scalable noise mining in long-term electrocardiographic time-series to predict death following heart attacks. In *Proceedings of the 20th ACM*

SIGKDD international conference on Knowledge discovery and data mining, pages 125–134, 2014. 1

[6] Alexander Geiger, D. Liu, Sarah Alnegheimish, Alfredo Cuesta-Infante, and K. Veeramachaneni. Tadgan: Time series anomaly detection using generative adversarial networks. *2020 IEEE International Conference on Big Data (Big Data)*, pages 33–43, 2020. 1

[7] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014. 4

[8] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 387–395, 2018. 4

[9] Eamonn Keogh, Jessica Lin, and Ada Fu. Hot sax: Efficiently finding the most unusual time series subsequence. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 8–pp. Ieee, 2005. 4

[10] Tung Kieu, Bin Yang, Chenjuan Guo, and Christian S Jensen. Outlier detection for time series with recurrent autoencoder ensembles. In *IJCAI*, pages 2725–2732, 2019. 1, 2, 3, 4, 5, 6

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[12] Istvan Kiss, Béla Genge, Piroska Haller, and Gheorghe Sebestyén. Data clustering-based anomaly detection in industrial control systems. In *2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 275–281. IEEE, 2014. 1

[13] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008. 1, 4

[14] Junshui Ma and Simon Perkins. Time-series novelty detection using one-class support vector machines. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, volume 3, pages 1741–1745. IEEE, 2003. 1

[15] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*, 2016. 4

[16] Larry M Manevitz and Malik Yousef. One-class svms for document classification. *Journal of machine Learning research*, 2(Dec):139–154, 2001. 4

[17] Lifeng Shen, Zhuocong Li, and James Kwok. Timeseries anomaly detection using temporal hierarchical one-class network. *Advances in Neural Information Processing Systems*, 33:13016–13026, 2020. 1, 2, 4, 5

[18] Lifeng Shen, Zhongzhong Yu, Qianli Ma, and James T Kwok. Time series anomaly detection with multiresolution ensemble decoding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9567–9575, 2021. 1, 2, 4, 5

[19] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2828–2837, 2019. 1, 2, 4

[20] Renjie Wu and Eamonn J Keogh. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *arXiv preprint arXiv:2009.13807*, 2020. 8

[21] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications. In *Proceedings of the 2018 World Wide Web Conference*, pages 187–196, 2018. 1, 4

[22] Yong-Ho Yoo, Ue-Hwan Kim, and Jong-Hwan Kim. Recurrent reconstructive network for sequential anomaly detection. *IEEE Transactions on Cybernetics*, 51:1704–1715, 2021. 1, 2, 5

[23] Chunkai Zhang and Yingyang Chen. Time series anomaly detection with variational autoencoders. *arXiv preprint arXiv:1907.01702*, 2019. 4

[24] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V Chawla. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1409–1416, 2019. 2

[25] Bin Zhou, Shenghua Liu, Bryan Hooi, Xueqi Cheng, and Jing Ye. Beatgan: Anomalous rhythm detection using adversarially generated time series. In *IJCAI*, pages 4433–4439, 2019. 1

Appendix

A. Additional Experimental Results

Due to the space limit, the additional experimental results for all datasets in *F1*, *PRAUC*, and *ROCAUC* are presented in Table 7, 8, and 9, respectively.

B. KDD Cup Competition Results

We illustrate our submissions' results from the competition in Figure 4. The submissions were evaluated by computing the correct percentile from 250 files (i.e., datasets). The max score one can obtain is 100%. As presented, the highest score we got is **58.4**. With this score, among 1967 submissions, we were ranked 58th out of 544 teams and 661 competitors.

B.1. Datasets

KDD Cup 2021 UCR Dataset[3]. The KDDCup21 datasets are created for KDDCup21 and designed to miti-

[3] https://compete.hexagon-ml.com/practice/competition/39/#data

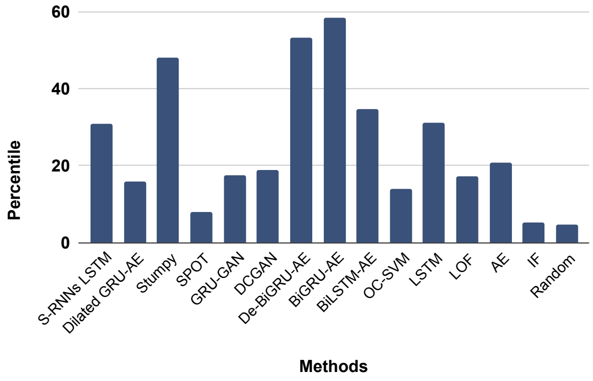


Figure 4. KDD Cup 2021 Results.

gate previous benchmark problems. The 250 datasets reflect more than 20 years of work surveying the time series anomaly detection literature. All 250 datasets are composed of univariate time-series and unknown domains.

B.2. Challenges

Through the KDD Cup 2021, the competition organizer proposed the limitation of previous existing benchmark datasets used in time-series anomaly detection task [20]. Existing datasets have four flaws: triviality, unrealistic anomaly density, mislabeled ground truth, and run-to-failure bias. (1) Triviality: A time-series anomaly detection problem is trivial if it can be solved with a single line of standard library MATLAB code. The overall 86.1% number is competitive with most papers that have examined this dataset. (2) Unrealistic anomaly density: More than half the test data exemplars consist of a contiguous region marked as anomalies. There are many regions marked as anomalies or the annotated anomalies are very close to each other. (3) Mislabeled: All of the benchmark datasets appear to have mislabeled data, both false positives, and false negatives. For example, if two data points have similar values, but one point is labeled as an anomaly and another is labeled as normal. (4) Run-to-failure bias: Many of the anomalies appear towards the end of the test datasets. Many real-world systems are run-to-failure, so in many cases, there is no data to the right of the last anomaly. A naive algorithm that simply labels the last point as an anomaly has an excellent chance of being correct.

Table 7. Performance Comparison in F1 scores.

Datasets	**IF**	**LOF**	**OC-SVM**	**CNN-AE**	**LSTM-AE**	**GRU-AE**	**BiGRU-AE**	**CNN-VAE**	**LSTM-VAE**	**GRU-VAE**	**CNN-GAN**	**LSTM-GAN**	**GRU-GAN**	**S-RNNS SF**	**Dilated LSTM AE**	**Mr.TAD**
Yahoo A1	0.7261	0.6397	0.7292	0.8166	0.8123	0.8084	0.8230	0.7394	0.6477	0.5706	0.6432	0.4233	0.4880	0.6967	0.8256	0.7078
Yahoo A2	0.0631	0.7697	0.4334	0.8730	0.8862	0.9136	0.9394	0.7540	0.6804	0.6259	0.6191	0.0589	0.1498	0.6990	0.9442	0.7259
Yahoo A3	0.2745	0.3939	0.3976	0.4072	0.4213	0.4617	0.6779	0.3321	0.2193	0.1713	0.1451	0.1656	0.1661	0.2776	0.4418	0.3244
Yahoo A4	0.2267	0.2862	0.3066	0.3617	0.4026	0.4199	0.6316	0.3150	0.2059	0.1612	0.1548	0.1524	0.1232	0.2113	0.4052	0.2418
NASA-SMAP	0.0685	0.2769	0.1324	0.2568	0.2296	0.2296	0.2296	0.2312	0.2296	0.2296	0.2296	0.2296	0.2296	0.2296	0.2296	0.2296
NASA-MSL	0.0195	0.1742	0.2391	0.1950	0.1950	0.1950	0.1950	0.1950	0.1950	0.1950	0.1950	0.1950	0.1950	0.1950	0.1950	0.1950
SMD	0.3199	0.3354	0.3348	0.4206	0.4483	0.4469	0.4430	0.4379	0.4420	0.4426	0.4078	0.4297	0.4147	0.4492	0.4579	0.4553
ECG 1	0.4056	0.4181	0.4471	0.2785	0.4466	0.4470	0.4667	0.4399	0.4375	0.4017	0.2947	0.3136	0.4211	0.5419	0.5291	0.4975
ECG 2	0.4307	0.6293	0.4053	0.2941	0.3874	0.3210	0.3550	0.3874	0.3433	0.3308	0.3067	0.2241	0.4167	0.3548	0.3824	0.3309
ECG 3	0.2103	0.2156	0.1976	0.0953	0.2796	0.2796	0.2857	0.1987	0.1732	0.1572	0.1988	0.1209	0.1173	0.1800	0.2143	0.1633
ECG 4	0.2410	0.1146	0.2135	0.2533	0.2715	0.2600	0.2645	0.2486	0.2698	0.2545	0.2214	0.2407	0.2461	0.2698	0.2331	0.2612
ECG 5	0.3152	0.5811	0.4771	0.4106	0.4878	0.4481	0.3556	0.3963	0.3636	0.3636	0.3614	0.1885	0.3679	0.4020	0.4859	0.3822
ECG 6	0.0932	0.1333	0.0996	0.2781	0.1571	0.1571	0.1605	0.1571	0.1601	0.1610	0.2043	0.1876	0.2030	0.1571	0.1571	0.1646
ECG 7	0.0650	0.1301	0.0643	0.0612	0.1460	0.0909	0.0643	0.0548	0.0563	0.0547	0.0587	0.0774	0.0465	0.0580	0.0744	0.1054
ECG 8	0.1112	0.0353	0.1264	0.2054	0.1758	0.1758	0.1758	0.1758	0.1763	0.1760	0.1954	0.2220	0.1874	0.1880	0.1758	0.1758
ECG 9	0.1704	0.2792	0.2595	0.3580	0.3580	0.3591	0.3580	0.3605	0.3580	0.3585	0.3583	0.3633	0.4552	0.3705	0.3583	0.3580
Power Demand	0.1826	0.0323	0.2451	0.2691	0.2632	0.2816	0.2643	0.2679	0.2643	0.2643	0.2632	0.4524	0.4167	0.2749	0.2802	0.2755
2D Gesture	0.5091	0.4533	0.5072	0.4132	0.4699	0.4435	0.4401	0.4548	0.4262	0.4167	0.4132	0.4444	0.4132	0.4640	0.4770	0.4759

Table 8. Performance comparison on PRAUC metric.

Datasets	**IF**	**LOF**	**OC-SVM**	**CNN-AE**	**LSTM-AE**	**GRU-AE**	**BiGRU-AE**	**CNN-VAE**	**LSTM-VAE**	**GRU-VAE**	**CNN-GAN**	**LSTM-GAN**	**GRU-GAN**	**S-RNNS SF**	**Dilated LSTM AE**	**Mr.TAD**
Yahoo A1	0.2069	0.4434	0.2163	0.5593	0.5442	0.5507	0.5820	0.4703	0.3962	0.3649	0.3764	0.2668	0.2998	0.4362	0.5566	0.4506
Yahoo A2	0.0014	0.2867	0.0537	0.4334	0.4547	0.4669	0.4817	0.4064	0.3592	0.3421	0.2894	0.0038	0.0431	0.3648	0.5123	0.3740
Yahoo A3	0.1105	0.1376	0.1494	0.2436	0.2578	0.2994	0.4641	0.1845	0.0981	0.0675	0.0254	0.0353	0.0264	0.1234	0.2830	0.1518
Yahoo A4	0.0463	0.0780	0.0780	0.1862	0.2214	0.2311	0.4192	0.1641	0.0793	0.0549	0.0250	0.0216	0.0135	0.0788	0.2212	0.0952
NASA-SMAP	0.0055	0.0633	0.0172	0.1238	0.1175	0.1193	0.1144	0.1076	0.1079	0.1104	0.1058	0.1114	0.1005	0.1100	0.1201	0.1177
NASA-MSL	0.0040	0.0221	0.0774	0.5198	0.4930	0.5088	0.5127	0.5215	0.5213	0.2902	0.5197	0.1888	0.3688	0.5213	0.5089	0.5110
SMD	0.2468	0.2640	0.1621	0.4185	0.4674	0.4780	0.4817	0.4445	0.4388	0.4377	0.3991	0.4188	0.4020	0.4472	0.4589	0.4632
ECG 1	0.2392	0.3054	0.3429	0.2568	0.4709	0.5046	0.4283	0.4299	0.4398	0.4247	0.2915	0.2891	0.3579	0.4964	0.5316	0.4813
ECG 2	0.2774	0.5007	0.3303	0.2583	0.3192	0.2940	0.3392	0.3087	0.3050	0.2932	0.2600	0.2024	0.3424	0.2951	0.3757	0.3015
ECG 3	0.0616	0.1185	0.0766	0.0335	0.2138	0.2102	0.2087	0.0689	0.0613	0.0626	0.0906	0.0498	0.0514	0.0656	0.1598	0.1129
ECG 4	0.0541	0.0118	0.0767	0.1580	0.1516	0.1519	0.1514	0.1597	0.1614	0.1602	0.1419	0.1499	0.1500	0.1588	0.1383	0.1591
ECG 5	0.2186	0.5463	0.3623	0.2172	0.3406	0.2815	0.2214	0.2603	0.2396	0.2481	0.2661	0.1260	0.2373	0.2501	0.3105	0.2439
ECG 6	0.0107	0.0262	0.0265	0.1342	0.0755	0.0819	0.0873	0.0824	0.0873	0.0906	0.0887	0.0832	0.1010	0.0783	0.0724	0.0841
ECG 7	0.0105	0.0606	0.0191	0.0213	0.0868	0.0381	0.0275	0.0194	0.0196	0.0193	0.0223	0.0302	0.0164	0.0203	0.0253	0.0363
ECG 8	0.0091	0.0011	0.0263	0.0931	0.0736	0.0754	0.0764	0.0779	0.0791	0.0842	0.0945	0.0985	0.0817	0.0866	0.0727	0.0764
ECG 9	0.0217	0.1362	0.0787	0.1610	0.1843	0.2207	0.2131	0.1735	0.1594	0.1559	0.1693	0.1870	0.2990	0.1968	0.1635	0.1621
Power Demand	0.0323	0.0008	0.0720	0.0996	0.1046	0.1012	0.1205	0.0953	0.1059	0.1152	0.0989	0.3742	0.3270	0.0960	0.0990	0.0968
2D Gesture	0.3883	0.2553	0.3946	0.3075	0.3260	0.3640	0.3259	0.2815	0.2573	0.2446	0.2318	0.3424	0.2402	0.2836	0.3831	0.3952

Table 9. Performance comparison on ROCAUC metric.

Datasets	**IF**	**LOF**	**OC-SVM**	**CNN-AE**	**LSTM-AE**	**GRU-AE**	**BiGRU-AE**	**CNN-VAE**	**LSTM-VAE**	**GRU-VAE**	**CNN-GAN**	**LSTM-GAN**	**GRU-GAN**	**S-RNNS SF**	**Dilated LSTM AE**	**Mr.TAD**
Yahoo A1	0.2723	0.0613	0.4752	0.8890	0.8789	0.8933	0.9073	0.8455	0.7807	0.7450	0.7034	0.4421	0.5446	0.7583	0.9070	0.7533
Yahoo A2	0.2616	0.4995	0.6295	0.9471	0.9598	0.9646	0.9817	0.8895	0.8696	0.8413	0.7701	0.0638	0.2063	0.8846	0.9946	0.8851
Yahoo A3	0.2309	0.1213	0.4306	0.6797	0.6629	0.6974	0.8222	0.5967	0.5337	0.4369	0.2592	0.2885	0.2128	0.5077	0.6579	0.5475
Yahoo A4	0.2329	0.0917	0.4204	0.6196	0.6336	0.6861	0.7829	0.5851	0.5158	0.4355	0.2410	0.2114	0.1366	0.4123	0.6530	0.4680
NASA-SMAP	0.0032	0.0582	0.0235	0.4957	0.4724	0.4749	0.4796	0.4003	0.4010	0.4216	0.4061	0.4243	0.3737	0.4139	0.4777	0.4690
NASA-MSL	0.0002	0.0117	0.0921	0.5137	0.5108	0.5109	0.5109	0.5132	0.5132	0.5298	0.5132	0.5218	0.5132	0.5132	0.5109	0.5111
SMD	0.1788	0.3674	0.5072	0.7069	0.7277	0.7215	0.7110	0.7376	0.7508	0.7517	0.7005	0.7047	0.6942	0.7477	0.7435	0.7411
ECG 1	0.1028	0.0323	0.2609	0.4007	0.6872	0.7909	0.7653	0.6048	0.6148	0.6315	0.4767	0.5080	0.7669	0.5547	0.6787	0.5862
ECG 2	0.1079	0.0497	0.3094	0.4750	0.6671	0.6187	0.6217	0.6428	0.6378	0.5915	0.4721	0.3408	0.7531	0.5365	0.6992	0.6148
ECG 3	0.0236	0.0004	0.1099	0.3860	0.6276	0.5997	0.6152	0.5727	0.5219	0.5576	0.6689	0.4462	0.5302	0.5489	0.5230	0.5466
ECG 4	0.0226	0.0019	0.1641	0.6054	0.5965	0.5874	0.5692	0.6093	0.6298	0.6132	0.5450	0.5597	0.5900	0.6171	0.5348	0.6099
ECG 5	0.4238	0.5357	0.7258	0.8192	0.8630	0.8529	0.7998	0.7998	0.7823	0.7916	0.7061	0.4662	0.7084	0.7920	0.8329	0.7998
ECG 6	0.0092	0.0090	0.1200	0.6941	0.3403	0.4015	0.4497	0.4441	0.4738	0.4890	0.5790	0.4683	0.5870	0.4108	0.3628	0.4713
ECG 7	0.0502	0.0023	0.2257	0.5325	0.5258	0.6000	0.5741	0.5316	0.5370	0.5286	0.5506	0.6988	0.4171	0.5515	0.5981	0.6561
ECG 8	0.0100	0.0006	0.1175	0.5293	0.3716	0.3908	0.3950	0.4111	0.4372	0.4706	0.5141	0.5561	0.4626	0.4868	0.3554	0.3932
ECG 9	0.0195	0.0142	0.1579	0.2915	0.3932	0.4438	0.4018	0.3172	0.3506	0.3373	0.2704	0.4405	0.6721	0.4758	0.2929	0.3373
Power Demand	0.0566	0.0006	0.1301	0.3295	0.3477	0.3259	0.3998	0.2934	0.3646	0.4110	0.3181	0.7439	0.6852	0.3189	0.3051	0.3152
2D Gesture	0.3604	0.0785	0.3199	0.4596	0.5948	0.5666	0.5665	0.5773	0.5188	0.4980	0.4680	0.5506	0.4904	0.5646	0.6374	0.6329