Recurrent auto-encoder with multi-resolution ensemble and predictive coding for multivariate time-series anomaly detection

Heejeong Choi · Subin Kim · Pilsung Kang

Accepted: 5 June 2023 / Published online: 8 August 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract
As large-scale time-series data can easily be found in real-world applications, multivariate time-series anomaly detection has played an essential role in diverse industries. It enables productivity improvement and maintenance cost reduction by preventing malfunctions and detecting anomalies based on time-series data. However, multivariate time-series anomaly detection is challenging because real-world time-series data exhibit complex temporal dependencies. For this task, it is crucial to learn a rich representation that effectively contains the nonlinear temporal dynamics of normal behavior. In this study, we propose an unsupervised multivariate time-series anomaly detection model named RAE-MEPC which learns informative normal representations based on multi-resolution ensemble reconstruction and predictive coding. We introduce multi-resolution ensemble encoding to capture the multi-scale dependency from the input time series. The encoder hierarchically aggregates the multi-scale temporal features extracted from the sub-encoders with different encoding lengths. From these encoded features, the reconstruction decoder reconstructs the input time series based on multi-resolution ensemble decoding where lower-resolution information helps to decode sub-decoders with higher-resolution outputs. Predictive coding is further introduced to encourage the model to learn more temporal dependencies of the time series. Experiments on real-world benchmark datasets show that the proposed model outperforms the benchmark models for multivariate time-series anomaly detection.

Keywords
Time-series anomaly detection · Recurrent auto-encoder · Multi-resolution ensemble · Predictive coding

1 Introduction
Anomaly detection is an important problem studied across diverse research areas and application domains [1]. The objective of anomaly detection is to identify instances that deviate significantly from normal ones [2]. With the advent of the Internet of Things, a substantial amount of multivariate time-series data has been generated in industrial settings such as smart factories, power plants, and cybersecurity [3–5]. As these time-series data exhibit complex abnormal patterns, anomaly detection on time-series data has become increasingly prevalent across these domains [7]. Time-series anomaly detection aims to recognize abnormal subsequences or time points in time series that deviate from the temporal behaviors of the entire sequence [7]. Given that undetected failures can result in critical damage, time-series anomaly detection has become indispensable in industrial domains [7]. It directly contributes to enhancing productivity and reducing operation and maintenance costs [7]. Consequently, many industrial systems require an accurate time-series anomaly detection model to prevent potential accidents and economic losses [6].

An effective time-series anomaly detection method should sufficiently capture the diverse information from the normal time-series data and detect anomalies based on the learned normal patterns [7]. However, there are some difficulties in this task. First, labeled anomalies are limited because they rarely occur in practice and require domain experts for hard annotations [8]. Therefore, time-series anomaly detection is usually formulated in the unsupervised learning setting with only normal samples. Second, multivariate time-series data have complex inherent information such as temporal dynam-
Recurrent auto-encoder... Therefore, it is crucial to learn normal patterns by extracting diverse characteristics from normal time-series data.

Time-series anomaly detection based on deep learning has been studied in two main categories: 1) prediction-based methods and 2) reconstruction-based methods [10]. First, prediction-based methods detect anomalies based on the learned normal patterns by forecasting the future time points from the past time sequences [11–15]. These methods assume that a predictive model cannot accurately estimate abnormal patterns when trained on normal data only. Therefore, they identify time points as abnormal whose predicted value is significantly different from the actual one. While prediction-based methods have demonstrated robust performance in detecting point anomalies (singular data points that deviate abruptly from normal patterns), they often produce false detections due to difficulties in accurately predicting values around sudden changes in time series [?].

Second, reconstruction-based methods learn normal patterns by encoding latent representation to reconstruct the input time series. These approaches assume that anomalies are less likely to be accurately reconstructed because the latent space cannot capture the patterns within unseen anomalies. Therefore, these methods use the reconstruction error as the anomaly score. Reconstruction-based methods usually investigate auto-encoder [16–21] or generative adversarial networks (GANs) [10, 22–24] to reconstruct the original input time-series data. These approaches are more effective than prediction-based methods at identifying contextual and collective anomalies—a series of consecutive data points that deviate from the normal range or fail to follow learned temporal patterns [?]. However, reconstruction-based methods often face challenges in mapping input time series to informative latent representations due to potential information loss during the reconstruction process.

In this study, we introduce the Recurrent Auto-Encoder with Multi-Resolution Ensemble and Predictive Coding (RAE-MEPC) for learning rich latent representations from time-series data. Figure 1 presents the overall flow of our proposed method. First, the proposed method captures temporal dynamics in multiple scales by using multi-resolution ensemble encoding and decoding. In the proposed method, the encoder consists of multiple sub-encoders with different encoding lengths. Each sub-encoder extracts the temporal features on a different scale. A sub-encoder with a short encoding length can focus on global patterns, but a sub-encoder with a long encoding length can capture more local characteristics. Subsequently, their features are hierarchically integrated into a final latent representation with multi-scale dependency. Based on multi-resolution ensemble decoding, the reconstruction decoder of RAE-MEPC reconstructs the input time series from the encoded features. The output from the sub-decoder with the highest resolution is used as the ensemble output. Second, predictive coding was employed as an auxiliary task to extract more temporal dynamics from the input time series. The prediction decoder forecasts future time points from the latent representations extracted by the encoder. Finally, once the proposed model is trained on a normal time series, it can detect anomalies based on the difference between the input time series and the reconstructed output.

The main contributions of this study can be summarized as follows:

- We propose a multi-resolution ensemble encoding method to learn the temporal dynamics of time-series data in multiple scales.
- We introduce predictive coding to capture more temporal information from the perspective of the prediction task.
- Based on the experimental results for univariate and multivariate time-series datasets, our proposed method outperforms benchmark models.

The remainder of this paper is organized as follows. In Section 2, we briefly review reconstruction-based time-series anomaly detection and methods for improving the quality of representation. Section 3 describes our proposed method RAE-MEPC with multi-resolution ensemble reconstruction and predictive coding. In Section 4, the experimental settings are explained, followed by the experimental results. Finally, we summarize our study in Section 5.
2 Related work

2.1 Time-series anomaly detection

2.1.1 Prediction-based methods

Prediction-based methods predict future time sequences to capture temporal features from normal time-series data. These methods can be categorized based on the backbone model used for handling time-series data. First, various methods have been proposed based on Recurrent Neural Networks (RNN), which are suitable for modeling sequential data. LSTM-AD [11] was proposed to detect anomalies with higher-level temporal features. The architecture of LSTM-AD is a stacked long short-term memory (LSTM) that predicts the time series over several time steps. In this method, anomalies can be detected based on prediction errors modeled as a multivariate Gaussian distribution. In [12], LSTM was also used to learn normal patterns in prediction tasks, addressing diversity, nonstationarity, and noise issues by automatically determining thresholds for data streams characterized by varying behavior and value ranges. Second, many methods used RNN and Convolutional Neural Networks (CNN) to extract spatial-temporal information from time-series data, such as the method proposed in [14], which consists of a stack of convolutional layers, LSTM layers, and fully connected layers. Another example is LSTNet [15], which combines an RNN and a CNN to model a mixture of long- and short-term patterns in time-series data. Additionally, prediction-based methods based on Transformers, such as STOC [7], have been actively studied. STOC is based on the Transformer model to learn dynamic patterns of sequential data through a self-attention mechanism, considering both global and local information by combining the outputs of multiple encoders. Although these various prediction-based methods are effective at detecting point anomalies, they cannot detect a series of consecutive abnormal points accurately.

2.1.2 Reconstruction-based methods

Reconstruction-based methods learn normal patterns in the process of reconstructing normal time series. These methods can be classified according to the framework employed for reconstructing time-series data. Initially, numerous approaches utilizing the auto-encoder framework have been introduced. Generally, these methods identify anomalies by assessing the likelihood of a reconstruction error. In the study conducted by [16], the authors proposed the EncDec-AD model, wherein the LSTM encoder maps the input data into a latent space while the LSTM decoder reconstructs the input data from the latent representation. LSTM-VAE [17] was proposed to address the issue that the fusion of high-dimensional and heterogeneous modalities is complex in model-based anomaly detection. This model introduces a progress-based varying prior to fusing the signals and reconstructing their distribution. The RAE-ensemble [20] was proposed to avoid overfitting the recurrent auto-encoder (RAE). The two solutions were built based on the sparsely connected RNN, which can produce multiple auto-encoders with different structures. In the RAE-ensemble, multiple auto-encoders are combined in independent and shared frameworks. These frameworks can prevent some auto-encoders from being overfitted by introducing an ensemble. RAMED [21] proposed to alleviate the error accumulation problem caused by the decoding step of RAE. This model introduced multi-resolution ensemble decoding to the RAE-ensemble. It shares the information between multiple decoders with different decoding lengths using lower-resolution information for higher-resolution decoding. A multi-resolution shape-forcing loss was further introduced to encourage the reconstructed outputs at multiple resolutions to match the global characteristics of the input time series. Subsequently, the Generative Adversarial Network (GAN) architecture has been employed to learn the normal distribution by reconstructing samples. Typically, these approaches detect anomalies by utilizing a novel anomaly score, which is pertinent to both discrimination and reconstruction processes. MAD-GAN [22] was proposed to exploit the spatio-temporal correlation and dependencies of the entire variable. TadGAN [10] was proposed to address scalability and portability issues in time-series anomaly detection. This method uses the LSTM for the generator and discriminator to capture the temporal correlations of time-series distributions. TadGAN is trained with cycle consistency loss to capture normal patterns effectively. It detects anomalies based on novel anomaly scores to combine the reconstruction errors and loss from the discriminator. BeatGAN [24] was proposed to detect anomalies based on normal data. It comprises a one-dimensional CNN and RNN, which are trained adversarially. Reconstruction-based methods have demonstrated promising performance in anomaly detection, employing a diverse range of architectures. Nonetheless, it is imperative to carefully design models that effectively extract abundant information from time-series data, as information loss may occur during the reconstruction process.

2.1.3 Other approaches

Various time-series anomaly detection methods have been explored, utilizing diverse approaches beyond prediction and reconstruction. Firstly, self-supervised learning has been investigated for efficient time-series anomaly detection. [?] proposed an effective multi-resolution self-supervised discrimination framework, which generates multi-resolution samples using a downsampling module and creates different
pseudo-labels for multi-scale behaviors. The self-supervised discrimination module separates anomalies from normal samples based on the temporal dynamics extracted from time series at multiple resolutions. [7] applied the self-supervised approach within the time-series anomaly detection framework, generating pseudo-labels by applying a set of different transformations to the normal training data. Once the classifier learns to discriminate between the various transformations, it determines whether new samples are anomalous based on a decision rule. Secondly, one-class classification has been employed to address time-series anomaly detection. [8] proposed a one-class time-series classification algorithm, which discriminates between seen and unseen classes using the model error of the signal transformation network. This network aims to transform input signals into the goal signal.

2.2 Modeling of long-term and multi-scale dependency

Various methods have been researched to extract long-term and multi-scale dependency from sequential time-series data. Most approaches are based on the RNN, which is a representative model for temporal features. A hierarchical RNN [25] was proposed to improve the long-term dependency of RNN. In this model, domain-specific priori knowledge is introduced to give meaning to the hidden variables representing the past context. This model hierarchically integrates temporal information of diverse delays and resolutions. A hierarchical multi-scale RNN [26] was proposed to resolve the issue that learning both hierarchical and temporal representation is difficult in RNN. This method consists of multiple recurrent layers with different time scales. It can extract hierarchical latent features with various temporal dependencies in a sequence by sharing the information of each layer. A temporal pyramid RNN [27] was proposed to learn long-term and multi-scale dependencies in sequential data. The architecture is built by stacking multiple recurrent layers of sub-pyramids. In this method, the input sequence of the higher layer is a large-scale aggregated state sequence produced by the sub-pyramids in the previous layer. It can explicitly learn multi-scale dependencies using multi-scale input sequences of different layers. Furthermore, a shortcut path is added to the output of each sub-pyramid to shorten the gradient feedback path of each layer and avoid vanishing gradient problems in RNNs.

2.3 Representation learning with predictive coding

Various methods have been studied for learning temporal features in diverse domains relevant to time series. In video representation learning, various methods have been proposed for learning temporal embedding based on the future frame prediction. [28] improved the video representation based on the reconstruction and prediction task. This method consists of one LSTM encoder and two LSTM decoders. The encoder maps the input video sequence into fixed-length representations, while two decoders reconstruct the input sequence and predict future sequences. In [29], a model based on a CNN that generates a future frame from an input video sequence was proposed. This method is based on a multi-scale architecture in which an adversarial training method and an image gradient difference loss function are applied to address the blurry predictions obtained from the mean squared error. Inspired by predictive coding, a predictive neural network [30] was proposed to overcome the problem that unsupervised learning leverages unlabeled examples to learn about the structure of a domain. This model can learn video representations in predicting future frames from a video sequence. In this model, each layer predicts future frames locally, before passing them on to the next layer, which then predicts the entire frame. A dense predictive coding [31] was proposed to learn a spatio-temporal embedding from a video in a self-supervised way. This method can encode dense sequences based on spatio-temporal blocks, which sequentially predict future representations. Furthermore, a curriculum training scheme was proposed to learn semantic representation by encoding only spatio-temporal signals with slow changing. In this training, the future representation is predicted gradually with less temporal context.

3 Methods

3.1 Recurrent auto-encoder with multi-resolution ensemble and predictive coding

In this paper, we propose multivariate time-series anomaly detection model called RAE-MEPC, which can model multi-scale and temporal dependency in time series using multi-resolution ensemble and predictive coding. Our proposed method aims to learn the normal patterns by reconstructing the input time series and predicting future time series based on the informative encoded features. Let \( X = [x_1, x_2, \ldots, x_T] \in \mathbb{R}^{d \times T} \) be the input time window, which is \( T \) time steps in time-series length. Each \( r \)-th time step in input time window \( x_r = [x_{1,r}, x_{2,r}, \ldots, x_{d,r}] \in \mathbb{R}^d \) has \( d \) number of variables. In this study, we detect anomalies in an unsupervised manner based on the assumption that most multivariate time series in the training data are normal. The proposed RAE-MEPC comprises an encoder and two decoders. The encoder maps the input time series into compressed representations with multi-scale dependency, while two decoders are built to reconstruct the input time series and predict future time steps. Figure 2 shows the architecture of our proposed model, which learns normal patterns based on the three components as follows: 1) multi-resolution ensem-
3.1.1 Multi-resolution ensemble encoding

The goal of the encoder in the proposed RAE-MEPC is to extract temporal features from an input time series at multiple scales. The encoder has $K^{(E)}$ sub-encoders with different encoding lengths. The $k$-th sub-encoder $E_k$ captures temporal behavior in time series of length $T^{(E_k)}$. This encoding length is defined as follows:

$$T^{(E_k)} = \left[ \frac{1}{\tau - 1} \times T \right], \quad 1 \leq k \leq K^{(E)},$$

where $\tau > 1$ is the hyperparameter determining the encoding length. Each sub-encoder receives a time series whose length matches its encoding length. To learn the multi-scale dependency of the original input time series, the input of each sub-encoder has a shape similar to that of the original input. Therefore, the input sequence of each sub-encoder $X^{(E_k)}$ is obtained by downsampling the original time window $X$ to a subsequence of length $T^{(E_k)}$, as shown in (2).

$$X^{(E_k)} = [x_1^{(E_k)}, x_2^{(E_k)}, \ldots, x_{T^{(E_k)}}^{(E_k)}] = [x_i],$$

$$i = [1 + j \times \frac{T - 1}{T^{(E_k)} - 1}], \quad 0 \leq j < T^{(E_k)}$$

Figure 2 shows the example of the multi-resolution ensemble encoding in $K^{(E)} = 3$, $T = 12$, and $\tau = 2$. Given an input subsequence of each sub-encoder, sequential data are compressed into a representation of a fixed length. In RAE-MEPC, all the sub-encoders have the LSTM architectures and achieve the features at different resolutions as follows:

$$h_i^{(E_k)} = \text{LSTM}^{(E_k)}(x_i^{(E_k)}; h_{i-1}^{(E_k)}),$$

where $h_i^{(E_k)}$ is the hidden state of $k$-th sub-encoder $E_k$ at $t$-th time step, while $x_i^{(E_k)}$ is the input subsequence of $E_k$ at $t$-th time step. LSTM$^{(E_k)}$ is the LSTM model of $E_k$, where the hidden state at the $t$-th time step is obtained from the previous hidden state $h_{i-1}^{(E_k)}$ and the current input $x_i^{(E_k)}$. In this process, all the sub-encoders independently outputs the last hidden state at different resolutions. The lower-resolution sub-encoder with shorter encoding length can extract macro temporal characteristics, whereas the higher-resolution sub-encoder with a longer encoding length can focus on local temporal patterns. Finally, the integrated encoded representation is obtained by hierarchically aggregating the last hidden states from the lowest resolution to the highest resolution as follows:

$$h^{(E_k)} = \text{MLP}^{(E_k)}(h_i^{(E_k)}), \quad k = K^{(E)},$$

$$h^{(E_k)} = \text{MLP}^{(E_k)}(h_i^{(E_k)} + h^{(E_{k+1})}), \quad k \neq K^{(E)},$$

$$h^{(E)} = h^{(E_1)},$$
where $\text{MLP}^{(E_k-E_{k+1})}$ is the fully connected layer that integrates the information of $E_k$ and $E_{k+1}$. $h^{(E_i)}_{\beta(E_k)}$ is the last hidden state of $E_k$ and $h^{(E_i)}$ is the integrated features from $E^{(E)}_k$ to $E_k$. $h^{(E)}$ is the final encoded representation obtained from the ensemble of all outputs of the sub-decoders with different resolutions. Finally, the encoder of RAE-MEPC can capture the multi-scale dependency with both general and local characteristics by introducing multi-resolution ensemble encoding.

### 3.1.2 Multi-resolution ensemble decoding

The reconstruction decoder of RAE-MEPC aims to reconstruct input time series based on encoded representation effectively. It reconstructs the input time series in reverse order based on the multi-resolution ensemble decoding proposed in RAMED [21]. The reconstruction decoder consists of $K^{(RD)}$ sub-decoders. Each sub-decoder has a different decoding length to encourage each sub-decoder to model the temporal patterns at different resolutions. The decoding length $T^{(RD_k)}$ of the $k$-th sub-decoder $RD_k$ is defined in the same manner as the encoding length as shown in (7).

$$T^{(RD_k)} = \left[ \frac{1}{\tau - 1} \times T \right], \quad 1 \leq k \leq K^{(E)}$$

The resolution of the sub-decoder is the same as that of the corresponding sub-encoder because they have the same length for encoding and decoding. In the reconstruction decoder, all the sub-decoders have an LSTM architecture. Each sub-decoder reconstructs the input time series of decoding length based on the final encoded representation without teacher forcing as follows:

$$y^{(RD_k)}_t = \begin{cases} h^{(E)}_t, & \text{if } t = T^{(RD_k)} \\ \text{MLP}^{(RD_k)}(h^{(RD_k)}_t), & \text{otherwise} \end{cases}$$

$$h^{(RD_k)}_{t-1} = \text{LSTM}^{(RD_k)}(y^{(RD_k)}_t, h^{(RD_k)}_t),$$

where $y^{(RD_k)}_t$ and $h^{(RD_k)}_t$ are the output and the hidden states of the $k$-th sub-decoder $RD_k$ at $t$-th decoding time step, respectively. LSTM$^{(RD_k)}$ is the fully connected layer for the output, whereas LSTM$^{(RD_k)}$ is the LSTM model to capture temporal dependency in $RD_k$. For each sub-decoder, a small amount of noise $\epsilon \delta$ is added to the input of LSTM. It is for improving robustness to partial corruption of the input pattern as in the denoising auto-encoder [32]. In RAMED [21], the reconstruction decoder can mitigate the error accumulation of the RAE using multi-resolution ensemble decoding. For this decoding, RAMED utilizes a coarse-to-fine fusion strategy to fuse a lower-resolution sub-decoder with a higher-resolution sub-decoder. It combines the hidden state of $RD_k$ with the information extracted from the coarser-grained sub-decoder $RD_{k+1}$. From this strategy, the integrated hidden state $h^{(RD_k)}_t$ is obtained as follows:

$$h^{(RD_k)}_t = \beta h^{(RD_k)}_{t+1} + (1 - \beta) \text{MLP}^{(RD_k-RD_{k+1})}([h^{(RD_k)}_t, h^{(RD_{k+1})}_t])$$

where $h^{(RD_k)}_t$ is the previous hidden state of $RD_k$, while $h^{(RD_{k+1})}_t$ is the corresponding hidden state of a nearby coarser sub-decoder $RD_{k+1}$. $\beta$ is the hyperparameter adjusting the degree to reflect the information in coarser-grained decoders. More lower-resolution information was exploited in each decoding step with a smaller $\beta$. From this process, the reconstruction decoder can use multi-resolution information by directly using the coarser-grained information to decode of the finer-grained decoder. After multi-resolution ensemble decoding, the reconstructed input time series in reverse order can be obtained from the highest resolution decoder. Finally, we achieve the final reconstructed output $\overline{Y}_{\text{recon}}$ by reversing the output of reconstruction decoder as shown in (11).

$$\overline{Y}_{\text{recon}} = [y^{(RD_1)}_1, y^{(RD_2)}_2, \ldots, y^{(RD_t)}_T]$$

### 3.1.3 Predictive coding

We introduce predictive coding to extract the long-term dependency of input time series. The goal of the prediction decoder is to predict the future time series after $T/2$ time steps based on compressed representation from the encoder. In the proposed model, the prediction decoder is LSTM architecture to perform multi-step prediction tasks simply as follows:

$$y^{(PD)}_0 = h^{(E)}_0,$$

$$h^{(PD)}_t = \text{LSTM}^{(PD)}(x_t, h^{(PD)}_{t-1}),$$

$$y^{(PD)}_t = \text{MLP}^{(PD)}(h^{(PD)}_t),$$

where $h^{(PD)}_t$ and $y^{(PD)}_t$ are the hidden state and the predicted output at $t$-th time step in the prediction decoder, respectively. MLP$^{(PD)}$ is the fully connected layer for the output, whereas LSTM$^{(PD)}$ is the LSTM model in the prediction decoder used to model temporal dependency. Finally, the prediction decoder predicts the future time series $\overline{Y}_{\text{pred}}$ after $T/2$ time step as shown in (15). The prediction decoder enables the encoder to learn the more informative patterns in normal time series by reflecting the temporal dependency from the prediction task’s perspective.

$$\overline{Y}_{\text{pred}} = [y^{(PD)}_1, y^{(PD)}_2, \ldots, y^{(PD)}_T]$$
3.2 Objective function

The total loss of RAE-MEPC has three loss terms as follows: 1) reconstruction error $L_{\text{recon}}$, 2) multi-resolution shape-forcing loss $L_{\text{shape}}$, and 3) prediction error $L_{\text{pred}}$. First, $L_{\text{recon}}$ is the mean squared error for the difference between the input time series and the reconstructed output as shown in (17). This loss can encourage the reconstructed output to be close to the input time series.

$$L_{\text{recon}} = \sum_{i=1}^{T} \| y_t^{(RD_i)} - x_t \|_2^2$$  \hspace{1cm} (16)

Second, $L_{\text{shape}}$ is the multi-resolution shape-forcing loss proposed in RAMED. This loss is introduced to encourage the sub-decoders in the reconstruction decoder to learn consistent temporal trends as the original input time series. It can force sub-decoders at different resolutions to learn similar temporal patterns as the input in multi-resolution ensemble decoding. Because $L_{\text{recon}}$ already makes the reconstructed output similar to the input time series, $L_{\text{shape}}$ is defined in the sub-decoders except for the sub-decoder with the highest resolution. This loss is based on DTW [33], which can calculate the distance between two time series data with different lengths. However, the DTW distance is non-differentiable because it contains a min operator. Therefore, the multi-resolution shape-forcing loss introduces smoothed DTW (sDTW) [34], which computes the soft minimum of all alignment costs in the DTW distance. (18) shows the sDTW distance where the smoothed minimum operator is introduced. Finally, $L_{\text{shape}}$ is based on the sDTW distance between the input time series $X$ and the output of the $k$-th sub-decoder $\nabla^{(RD_k)}$ as follows:

$$sDTW(X, \nabla^{(RD_k)}) = -\gamma \log \sum_{A \in \Lambda} e^{-(AC)/\gamma},$$ \hspace{1cm} (17)

$$L_{\text{shape}} = \frac{1}{K(RD)} - \sum_{k=2}^{K(RD)} sDTW(X, \nabla^{(RD_k)}),$$ \hspace{1cm} (18)

where $\Lambda$ is the alignment matrix of the DTW alignment path, $C$ is the matrix for alignment costs, whose element is the Euclidean distance between the corresponding elements of the two time series, and $\gamma$ is a hyperparameter related to the smoothed minimum operator. More detailed information on the multi-resolution shape-forcing loss can be found in [21].

Finally, $L_{\text{pred}}$ is the square of the difference between the predicted output and the actual future time series as shown in (19). It encourages the prediction decoder to predict the future time series after the $T/2$ time step effectively. It also enables the encoder to capture additional temporal information, which is helpful for the prediction task.

$$L_{\text{pred}} = \sum_{i=1}^{T} \| y_t^{(PD)} - x_{t+[T/2]} \|_2^2$$ \hspace{1cm} (19)

From above three loss terms, RAE-MEPC is trained by minimizing the total loss $L_{\text{total}}$ as shown in (20). The weights $\lambda_{\text{shape}}$ and $\lambda_{\text{pred}}$ represent the hyperparameters that control the importance of $L_{\text{shape}}$ and $L_{\text{pred}}$, respectively. The detailed training procedure of the proposed RAE-MEPC is shown in Algorithm 1.

$$L_{\text{total}} = L_{\text{recon}} + \lambda_{\text{shape}} L_{\text{shape}} + \lambda_{\text{pred}} L_{\text{pred}}$$ \hspace{1cm} (20)

3.3 Anomaly detection

Once the RAE-MEPC is trained by minimizing the objective function, anomalies can be detected using the reconstruction error. Specifically, the anomalies in the test set can be detected based on the distribution of the reconstruction error in the validation set. Given a time series in the validation set, the residual $e_t$ is obtained at each time step, as shown in (21). Then we estimate the distribution $e_t \sim N(\mu, \Sigma)$ of the normal residuals based on the maximum likelihood estimation. Based on this distribution, we can detect anomalies.
Algorithm 2 Detect anomalies using the RAE-MEPC.

Require: a validation set \( X_{\text{valid}} = \{X_1^{\text{valid}}, ..., X_N^{\text{valid}}\} \), a test set \( X_{\text{test}} = \{X_1^{\text{test}}, ..., X_N^{\text{test}}\} \), encoder \( E \), reconstruction decoder \( RD \), predefined threshold \( \text{thres} \)

Ensure: Anomaly labels of all time series in the test set
1: for \( i = 1, ..., N_{\text{valid}} \) do
2: Feed \( X_i^{\text{valid}} \) to \( E \) and obtain encoding representations \( h(E) \)
3: Feed \( h(E) \) to \( RD \) and obtain reconstruction outputs \( \hat{Y}_{\text{recon}} \)
4: Calculate residual \( e_i^{\text{valid}} \) for each time step via (21)
5: end for
6: Estimate residual distribution \( N(\mu, \Sigma) \) from \( \{e_t^{\text{valid}}\} \)
7: for \( i = 1, ..., N_{\text{test}} \) do
8: Feed \( X_i^{\text{test}} \) to \( E \) and obtain \( h(E) \)
9: Feed \( h(E) \) to \( RD \) and obtain \( \hat{Y}_{\text{recon}} \)
10: Obtain Anomaly Score for each time step via (21)-(22)
11: if Anomaly Score > thres then
12: Anomaly label of \( x_i^{\text{test}} \) is abnormal
13: else
14: Anomaly label of \( x_i^{\text{test}} \) is normal
15: end if
16: end for

in the unseen time series in the test set. The anomaly score is calculated using (22). This score represents the degree to which the residual of the given test data deviates from the estimated normal distribution. Finally, the instances are detected as anomalies if the anomaly score exceeds the predefined threshold. The detailed anomaly detection procedure of the proposed RAE-MEPC is shown in Algorithm 2.

\[
e_t = y_t^{(RD)} - x_t \tag{21}
\]

\[
\text{Anomaly Score} = (e_t - \mu)^T \Sigma^{-1} (e_t - \mu) \tag{22}
\]

4 Experiments

4.1 Experimental settings

4.1.1 Data

In this study, we evaluate our proposed method on the two real-world benchmark datasets for time-series anomaly detection: Power-demand [35] and 2D-gesture [35]. The datasets analysed during the current study are available at http://www.cs.ucr.edu/~eamonn/discords.

Power-demand is univariate dataset. It contains the power consumption measured in a Dutch research facility for the entire year of 1997. In this dataset, normal patterns of power demand from 9 am to 5 pm on an ordinary week (Monday to Friday) were accumulated. Anomalies were measured during unusual weeks such as holidays.

2D-gesture is bivariate dataset. It consists of the X and Y coordinates of the right hand of actors. These coordinates were extracted from video images where the actor grabs a gun from a hip-mounted holster, moves it to the target, and returns it to the holster. The anomalies were defined as the scenes where actors did not return the gun to the holster.

We followed the experimental setting in [21]. For Power-demand and 2D-gesture, the raw dataset has only a training set and a test set. We used 30% of the training set as the validation set in each dataset to allow model selection and hyperparameter tuning. The raw training data are partitioned into a time window of fixed length using a sliding window to design the temporal data. In this study, the length of the time window was set as 512 for the Power-demand dataset and 64 for the 2D-gesture dataset. We set the sliding window to have a stride of 256 on the Power-demand dataset and 32 for the 2D-gesture dataset. For a fair comparison, we employed the same data pre-processing for all methods. Table 1 summarizes the dataset statistics.

4.1.2 Baseline methods

The proposed model was compared to the following three categories of multivariate time-series anomaly detection algorithms. The first category is the prediction-based anomaly detection model: LSTM-ED [11]. Its architecture is a stacked LSTM to detect anomalies in a time series. This network was trained on only normal data and was used as a predictor over several time steps. The prediction errors were used as anomaly scores by modeling them as a multivariate Gaussian distribution. The second category is the reconstruction-based GAN method, and MAD-GAN [22] is employed as a representative model for this category. It is an unsupervised multivariate anomaly detection method based on a GAN consisting of a generator and discriminator. In the MAD-GAN, a novel anomaly score called the DR-score was used to detect anomalies based on discrimination and reconstruction. The third category is reconstruction-based methods, in which our proposed method belongs. EncDec-AD [16] and RAMED...
were employed as benchmark models. EncDec-AD was trained to reconstruct normal time-series behavior based on the RAE architecture. In this model, the reconstruction errors were used to detect anomalies. RAMED is a simple yet efficient recurrent network ensemble. By using decoders with different decoding lengths and a new coarse-to-fine fusion mechanism, lower-resolution information can aid in long-range decoding for decoders with higher-resolution outputs. In this model, we used the output from the decoder with the highest resolution to obtain an anomaly score at each time step.

4.1.3 Evaluation metrics

We evaluated the proposed RAE-MEPC using the following metrics for time-series anomaly detection: area under the ROC curve (AUROC), area under the precision-recall curve (AUPRC), and the best F1-score. AUROC and AUPRC were used to evaluate the threshold-independent intrinsic anomaly detection ability of the model. AUROC measures the entire two-dimensional area under the ROC curve, which measures how accurately the model can detect true anomalies under a certain level of false alarm. This score indicates the performance of the anomaly detector at various threshold settings. AUPRC shows the trade-off between precision and recall for different thresholds. A high score represents both high recall and high precision, where high precision relates to a low false-positive rate, and high recall relates to a low false-negative rate. The best F1-score is the highest F1-score at the different thresholds, and it is selected from the F1-scores in 1,000 thresholds uniformly distributed from zero to the maximum anomaly score in the test dataset.

### Table 2 Overall performance comparison

| Dataset: 2D-gesture | Model Type | Model       | F1-Score | AUROC   | AUPRC   | Training Time per Epoch (m) |
|---------------------|------------|-------------|----------|---------|---------|----------------------------|
| Prediction          | LSTM-AD    | 0.5255      | 0.7409   | 0.5081  | 0.80    |
| Reconstruction (GAN)| MAD-GAN    | 0.4057      | 0.4867   | 0.2323  | 0.02    |
| Reconstruction      | EncDec-AD  | 0.5573      | 0.7745   | 0.5457  | 0.05    |
|                     | RAMED      | 0.5625      | 0.7813   | 0.5797  | 0.60    |
|                     | RAE-MEPC   | 0.5685      | 0.7973   | 0.5915  | 0.39    |

| Dataset: Power-demand | Model Type | Model       | F1-Score | AUROC   | AUPRC   | Training Time per Epoch (m) |
|-----------------------|------------|-------------|----------|---------|---------|----------------------------|
| Prediction            | LSTM-AD    | 0.2549      | 0.6580   | 0.1704  | 0.91    |
| Reconstruction (GAN)  | MAD-GAN    | 0.2621      | 0.6081   | 0.1235  | 0.10    |
| Reconstruction        | EncDec-AD  | 0.2103      | 0.5034   | 0.0982  | 0.21    |
|                       | RAMED      | 0.2953      | 0.6905   | 0.1787  | 4.82    |
|                       | RAE-MEPC   | 0.2940      | 0.6931   | 0.2372  | 2.92    |

4.1.4 Implementation details

In this study, we used the encoder with three sub-encoders at different resolutions. We also built the reconstruction decoder using three sub-decoders with different decoding lengths. In RAE-MEPC, the sub-encoder, sub-decoder, and prediction decoder were based on the single-layer LSTM architectures. We performed a grid search on the hyperparameters as follows: encoder/decoder hidden dimension in \{16, 32, 64\}, \(\tau\) in \{2, 3, 4\}, \(\beta\) in \{0.1, 0.3\}, and \(\lambda_{shape}\) in \{0.0001, 0.001\}. For the 2D-gesture dataset, we set encoder/decoder hidden dimension, \(\tau\), \(\beta\), and \(\lambda_{shape}\) to 32, 4, 0.1, and 0.001, respectively. For the Power-demand dataset, encoder/decoder hidden dimension, \(\tau\), \(\beta\), and \(\lambda_{shape}\) were set to 64, 3, 0.1, and 0.001, respectively. We set \(\lambda_{pred}\) to one. We trained our proposed model using Adam optimizer [36] with an initial learning rate of 0.001. All experiments were performed on a Linux workstation with Intel Core i7-9700X CPU, 128 GB RAM, and NVidia GeForce RTX 3090 GPU using PyTorch.

4.2 Experimental results

4.2.1 Overall performance comparison

We compared our proposed method, RAE-MEPC, with existing prediction-based and reconstruction-based methods by reproducing the performance of the baselines with hyperparameter search. Table 2 presents the best results of the proposed model and baseline methods. RAE-MEPC outperformed the baseline models for all three metrics on the 2D-gesture dataset. It also showed the best performance on AUROC and AUPRC for the Power-demand dataset. Among the three metrics, the improvement in AUPRC is particularly significant. It showed that the proposed method...
achieved lower false positive and false negative rates than other methods. Notably, the proposed model exhibited higher performance than RAMED, which employs the same multi-resolution ensemble decoding. These results underscore the positive impact of using multi-resolution ensemble encoding and predictive coding on learning rich normal time-series representations and effectively detecting anomalies. We also evaluated the computational performance of RAE-MEPC and compared it to the baseline methods by measuring the time taken per epoch for all methods with the best hyperparameters. RAE-MEPC exhibited a shorter training time than RAMED, the method with performance most similar to ours. This outcome resulted from the fact that the best resolution hyperparameter \( \tau \) of RAE-MEPC is larger than that of RAMED, and RAE-MEPC handles shorter sequences. Although RAE-MEPC required more training time than LSTM-AD, MAD-GAN, and EncDec-AD, it delivered a robust performance in unsupervised anomaly detection. These results indicate that the proposed model achieved good performance with a reasonable computational cost. Moreover, we can also conclude that RAE-MEPC works well on both univariate and multivariate datasets. On the other hand, MAD-GAN achieved lower performance on the 2D-gesture dataset compared to the Power-demand dataset. It showed that GAN-based models usually have training difficulties because they can easily suffer from mode collapse and non-convergence problems.

As shown in Fig. 3, we qualitatively evaluated the proposed reconstruction-based method. Figure 3a shows the raw input time series of the test set on the 2D-gesture dataset. Figure 3b presents the reconstructed output of RAE-MEPC. The black and gray lines indicate the X- and Y-coordinates, respectively. The red region represents the time steps when the true anomalies were obtained. The reconstructed time series showed similar trends with the input time series on two variables in a normal period. In an abnormal period, RAE-MEPC also

| Model                                      | F1-Score | 2D-gesture AUROC | AUPRC | F1-Score | Power-demand AUROC | AUPRC |
|--------------------------------------------|----------|------------------|-------|----------|--------------------|-------|
| w/o multi-resolution ensemble encoding     | 0.5681   | 0.7898           | 0.5575| 0.2794   | 0.6871             | 0.1786|
| w/o predictive coding                      | 0.5683   | 0.7952           | 0.5858| 0.2886   | 0.6953             | 0.1857|
| Full model                                 | **0.5685**| **0.7973**       | **0.5915**| **0.2940**| **0.6931**         | **0.2372**|
Table 4  Effect of $\tau$

| $\tau$ | 2D-gesture F1-Score | AUROC | AUPRC | Power-demand F1-Score | AUROC | AUPRC |
|-------|---------------------|-------|-------|-----------------------|-------|-------|
| 2     | 0.5463              | 0.7574| 0.5152|                       |       |       |
| 3     | 0.5522              | 0.7722| 0.5640|                       |       |       |
| 4     | 0.5685              | 0.7973| 0.5915|                       |       |       |

reconstructed the samples that followed an estimated normal distribution. Therefore, high anomaly scores were derived based on the significant difference between the actual and reconstructed values in this abnormal region, as shown in Fig. 3c. From these results, we can conclude that the proposed RAE-MEPC learns rich normal time-series representations by introducing multi-resolution ensemble encoding and predictive coding.

4.2.2 Effects of model components

In this experiment, we evaluated the effects of multi-resolution ensemble encoding and predictive coding in the proposed model RAE-MEPC. We considered these model components by eliminating one component from the full model. We used the encoder with three sparsely connected RNN-based sub-encoders for the model without multi-resolution ensemble encoding, similar to RAMED. Without predictive coding, the model only performed multi-resolution ensemble reconstruction, and it was trained only on the sum of reconstruction and multi-resolution shape-forcing losses. Table 3 contains the best performances obtained in hyperparameter search for all models. As shown in Table 3, both multi-resolution ensemble encoding and predictive coding are essential. Without multi-resolution ensemble encoding, the performance decreased for all datasets and evaluation metrics. This result demonstrated that it is helpful to merge the features extracted from input time series on different resolutions. The model resulted in lower performance on all datasets and evaluation metrics without predictive coding. From this result, we can conclude that extracting temporal features of normal data from the perspective of a prediction task can help achieve rich time-series representation.

4.2.3 Sensitivity to hyperparameters

We conducted a sensitivity analysis for the following two hyperparameters: 1) resolution hyperparameter $\tau$ in (1) and (7), and 2) weight of prediction loss $\lambda_{\text{pred}}$ in (20). We set the default hyperparameters to $\beta = 0.1$, $\tau = 4$, $\lambda_{\text{pred}} = 1$, and encoder/decoder hidden dimension of 32. The $\lambda_{\text{shape}}$ was set to 0.001 for the 2D-gesture and 0.0001 for the Power-demand dataset. Table 4 shows the performances with different $\tau$ from two to four. When $\tau$ is four, RAE-MEPC exhibited the best performance on all evaluation metrics. As shown in Table 4, increasing $\tau$ can improve performance as more diverse multi-resolution temporal patterns are handled in the encoder and decoder. From these results, we can confirm the effects of multi-resolution ensemble in encoding and decoding steps. Furthermore, the obtained results demonstrated that RAE-MEPC possesses notable computational efficiency. This efficiency can be attributed to the fact that, with a larger $\tau$, RAE-MEPC manages shorter sequences within the sub-encoder and decoder. This efficient handling of sequences contributes to the overall performance of the proposed method.

Finally, we studied the effects of prediction loss by increasing $\lambda_{\text{pred}}$ from zero to one on the two datasets. Figure 4 shows AUPRC with different $\lambda_{\text{pred}}$. RAE-MEPC achieved the highest AUPRC when $\lambda_{\text{pred}}$ was set to one. As shown in Fig. 4, better performance was achieved with the larger $\lambda_{\text{pred}}$ on all datasets. From these results, we can notice that it is helpful in time-series anomaly detection to learn the nor-
mal features from the perspective of a prediction task as well as a reconstruction task.

5 Conclusion

Time-series anomaly detection has been employed in industries where large-scale time series can be accessed easily, such as smart factories. Time-series anomaly detection can improve productivity and reduce economic loss by monitoring potential risks and preventing faults of the systems. Therefore, it is crucial to detect anomalies accurately in the real world. Extracting rich temporal features from normal data can improve time-series anomaly detection performance.

In this paper, we proposed RAE-MEPC for unsupervised multivariate time-series anomaly detection. RAE-MEPC improves the quality of time-series representation using multi-resolution ensemble and predictive coding. This study introduced multi-resolution ensemble encoding to learn multi-scale dependency from input time series. The encoder achieves rich time-series representation by hierarchically combining the features extracted from sub-encoders with different resolutions. The proposed model reconstructs input time series using multi-resolution ensemble decoding on a coarse-to-fine method from this encoded representation. Moreover, we introduced predictive coding to extract temporal features from the perspective of a prediction task, and it encourages the encoder to capture more temporal features. Experiments on various time-series benchmark datasets demonstrated that the proposed model can detect anomalies more accurately than the well-known benchmark models.

Despite the favorable anomaly detection performance of the proposed RAE-MEPC, there are some limitations in the current work, which lead us to future research directions. First, although RAE-MEPC is effective in capturing multi-resolution temporal information of time-series data, it is weak in modeling the inter-correlation between variables. In future work, we will focus on an anomaly detection approach that can simultaneously model the inter-metric and temporal dependency for multivariate time series. Second, the proposed method causes some false alarms, although it significantly aids in detecting anomalies. In future research, we will improve the post-processing steps to reduce the number of false positives.

Acknowledgements This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT) (No. NRF-2022R1A2C2005455). This work was also supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-00034, Clustering technologies of fragmented data for time-based data analysis)

Declarations

Conflicts of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper

References

1. Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: A survey. ACM computing surveys (CSUR) 41(3):1–58
2. Chalapathy R, Chawla S (2019) Deep learning for anomaly detection: A survey. arXiv:1901.03407
3. Lee G-Y, Kim M, Quan Y-J, Kim M-S, Kim TJY, Yoon H-S, Min S, Kim D-H, Mun J-W, Oh JW et al (2018) Machine health management in smart factory: A review. J Mech Sci Technol 32(3):987–1009
4. Laubscher R (2019) Time-series forecasting of coal-fired power plant reheater metal temperatures using encoder-decoder recurrent neural networks. Energy 189:116187
5. Pokhrel NR, Rodrigo H, Tsokos CP et al (2017) Cybersecurity: Time series predictive modeling of vulnerabilities of desktop operating system using linear and non-linear approach. J Infor Secur 8(04):362
6. Wang Y, Perry M, Whitlock D, Sutherland JW (2020) Detecting anomalies in time series data from a manufacturing system using recurrent neural networks. J Manufac Syst
7. Shen L, Li Z, Kwok J (2020) Timeseries anomaly detection using temporal hierarchical one-class network. Advances in Neural Information Processing Systems 33:13016–13026
8. Canizo M, Triguero I, Conde A, Onieva E (2019) Multi-head cnn-rnn for multi-time series anomaly detection: An industrial case study. Neurocomputing 363:246–260
9. Su Y, Zhao Y, Niu C, Liu R, Sun W, Pei D (2019) Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In: Proc 25th ACM SIGKDD Int Conf Knowledge Discovery & Data Mining pp. 2828–2837
10. Geiger A, Liu D, Alnegheimish S, Cuesta-Infante A, Veeramachaneni K (2020) Tadgan: Time series anomaly detection using generative adversarial networks. In: 2020 IEEE International Conference on Big Data (Big Data) pp. 33–43
11. Malhotra P, Vig L, Shroff G, Agarwal P (2015) Long short term memory networks for anomaly detection in time series. In: Proceedings, vol.89, pp. 89–94
12. Hundman K, Constantiniu V, Laporte C, Colwell I, Soderstrom T (2018) Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining pp. 387–395
13. Shalyga D, Filonenov P, Lavrentyev A (2018) Anomaly detection for water treatment system based on neural network with automatic architecture optimization. arXiv:1807.07282
14. Kravchik M, Shabtai A (2018) Detecting cyber attacks in industrial control systems using convolutional neural networks. In: Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy pp. 72–83
15. Lai G, Chang W-C, Yang Y, Liu H (2018) Modeling long-and short-term temporal patterns with deep neural networks. In: The 41st
