Applied Intelligence
https://doi.org/10.1007/s10489-021-02532-x

Finder: A novel approach of change point detection for multivariate time series

Haizhou Du[1] · **Ziyi Duan**[1]

Accepted: 12 May 2021

Abstract
The multivariate time series often contain complex mixed inputs, with complex correlations between them. Detecting change points in multivariate time series is of great importance, which can find anomalies early and reduce losses, yet very challenging as it is affected by many complex factors, *i.e.*, dynamic correlations and external factors. The performance of traditional methods typically scales poorly. In this paper, we propose Finder, a novel approach of change point detection via multivariate fusion attention networks. Our model consists of two key modules. First, in the time series prediction module, we employ multi-level attention networks based on the Transformer and integrate the external factor fusion component, achieving feature extraction and fusion of multivariate data. Secondly, in the change point detection module, a deep learning classifier is used to detect change points, improving efficiency and accuracy. Extensive experiments prove the superiority and effectiveness of Finder on two real-world datasets. Our approach outperforms the state-of-the-art methods by up to 10.50% on the F1 score.

Keywords Multivariate time series · Change point detection · Multivariate fusion · Multi-level attention · Transformer

1 Introduction

Multivariate time series refers to the time series that simultaneously represents multiple variables or indicators. It usually contains complex mixed inputs, including the target series' historical data, relevant variables, external factors, and so on. Still, the prior information about their correlations with the target series is very limited.

With the recent advances in data acquisition technologies and the availability of big datasets, abundant data can be collected and fused, enabling much analysis and applications of multivariate time series data [17, 34, 48]. At the same time, change point detection in multivariate time series has received extensive attention, and has important applications in many fields, *e.g.* finance, power, traffic, environment, and medicine.

Change points are the moment when the state or property of the time series changes abruptly [3]. Increasing the detecting accuracy is beneficial to operational efficiency in many aspects of society, and has important practical significance [18, 42, 50]. For example, in power failure analysis, we can mine the rules and trends of power load changes over time to discover potential mutations and take corresponding preventative measures early to reduce financial and time losses [7, 36, 53]. In urban traffic prediction, we can provide insights for urban planning and traffic management to improve the efficiency of public transportation, as well as give early warnings for public safety emergency management [12, 29, 32].

However, the real-world multivariate time series is affected by various complex factors, including relevant variables and external factors, and the information about their interactions with the target series is very limited. Due to data sources' heterogeneity and dynamic correlations, detecting change points in multivariate time series is particularly challenging.

The current approaches of change point detection in time series are mainly split up into probability and statistics-based, classification-based, and prediction-based. Most traditional methods typically use classical data mining algorithms, showing poor performance in complex real-world applications. Recently, the research on the deep learning-based approach has made significant progress. Deep neural

✉ Haizhou Du
duhaizhou@gmail.com

[1] School of Computer Science and Technology, Shanghai University of Electric Power Shanghai, Shanghai, 200090, China

Published online: 12 June 2021

networks (DNNs) have demonstrated substantial performance improvements in time series modeling [9]. The recent improvements have also employed attention-based methods [28, 33] and Transformer architectures [45].

However, many approaches use small datasets with very limited features, providing coarse-granularity analysis only, lacking the ability to detect change points on complex real-world datasets. More importantly, these methods usually fail to consider the unique features of multivariate time series, such as the heterogeneity of data sources or temporal correlations. Thus, they unable to achieve feature extraction and fusion between different types of inputs.

To address the aforementioned issues, we propose Finder, a prediction-based approach for change point detection, employing the novel multivariate fusion attention networks. It can achieve high-performance of detecting change points in multivariate time series.

In this paper, Finder consists of two key modules, the time series prediction module and the change point detection module. We process the multivariate data and learn the complex correlations for predicting the target series in the time series prediction module. And we use the deep neural network to classify the target series and further detect change points in the change point detection module.

Accordingly, the key contributions of this paper can be summarized as follows.

- We propose Finder, a change point detection approach in multivariate time series. To the best of our knowledge, this is the first time in this field combining the ideas of prediction (time series prediction module) and classification (change point detection module) in the same deep learning-based framework.
- In the time series prediction module, we integrate the multi-level attention and the Transformer-based multi-head attention, which we called multivariate fusion attention networks, achieving feature extraction and the fusion of multivariate data. Specifically, in the first level, we employ the input attention mechanism for historical data and related variables. It aims to adaptively extract features and capture the correlations with the target series. In the second level, inspired by the Transformer, the multi-head attention is applied to achieve the fusion of multivariate data with different temporal features, and capture the temporal dependencies at all time steps.
- We design the external factor fusion component to incorporate the external factors from different domains. The learned representations are fused into the multi-level attention networks through the gating mechanisms, to enhance the importance of these external factors and learn the latent valuable information they provide.
- In the change point detection module, based on the deviation between the actual and the predicted value, we use a deep learning classifier to detect change points. This method reduces the interference of human factors and the dependence on the parameters (*i.e.* threshold) selection, balances efficiency and accuracy, and improves the performance of the model.

To demonstrate the effectiveness of Finder, we conducted experiments on two public datasets in different domains. Extensive experimental results show that our approach outperforms current state-of-the-art models.

The rest of the paper is organized as follows. We discuss the challenges of change point detection and the corresponding solutions in Section 2. Then we introduce the overview and the details of Finder in Section 3. We present the evaluation results and analyze the performance in Section 4. Next, we summarize the related work in Section 5. Lastly, we conclude our paper and sketch directions for the possible future work in Section 6.

2 Motivation

Improving the accuracy of change point detection in multivariate time series is closely related to all walks of life, and has important time and economic significance. However, due to the heterogeneity of data sources, the real-world multivariate time series is affected by a variety of complex factors. Thus, the existing methods have been unable to adapt to the current requirements anymore. The current multivariate time series change point detection has the following challenges:

1. **Dynamic correlations**

   The real-world multivariate time series usually contains mixed inputs and is affected by a variety of factors, including relevant variables and external factors. They have complex correlations and strong dependencies with the target series. The correlations are highly dynamic, changing over time non-linearly [28, 52].

   However, the existing methods fail to consider different types of data in complex mixed inputs, or neglect important external factors. They just simply concatenate the time-dependent features in the input data together, and unable to achieve feature extraction and fusion between different inputs.
2. **External factors**

   External factors are the factors that have no direct correlations but have an indirect impact on the target series. For example, environmental or meteorology factors (*e.g.*, a strong wind) in wind power failure analysis, time information (*e.g.* peak periods/holiday)

in traffic forecasting, etc. External factors are generally divided into categorical factors and continuous factors, however, the categorical factors cannot be fed to the neural networks directly.

Traditional methods cannot achieve the feature extraction and fusion of these factors, especially when the temporal features of them are different from other inputs. It will significantly weaken the importance of these factors and affect the accuracy of the results.

Based on the above challenges, in order to learn the dynamic correlations, achieve the extraction and fusion of multivariate data, employing the prediction-based approach is our best option, which can overcome the inherent shortcomings of traditional methods. So, the core idea of this paper is the Finder we proposed, a prediction-based change point detection approach employing the multivariate fusion attention networks.

Specifically, in order to solve the challenge of dynamic correlations, we combine input attention and multi-head attention to achieve feature extraction and fusion of multivariate data, and capture complex correlations. To solve the challenge of external factors, we design an external factor fusion component to learn the latent features of these factors, and achieve fusion through the gating mechanism.

Taken together, Finder can solve the main challenges more comprehensively of the current multivariate time series change point detection.

3 Finder design

In this section, we first introduce the overall workflow of Finder. Then, we present the theories and details of the two core modules.

As we introduced in Section 2, Finder is a prediction-based method. The basic idea of the prediction-based method is that we first predict the target series, then detect change points based on the deviation between the predicted value and the true value. Therefore, Finder consists of two major modules. Figure 1 presents the overall workflow of our approach.

1. **Time Series Prediction Module.** It predicts the target series via the multivariate fusion attention networks, integrating the external factor fusion component. Aiming to learn the dynamic correlations and achieve the feature fusion of multivariate data.
2. **Change Point Detection Module.** According to the deviation between the predicted value and the actual value, we use the deep learning classifier to detect change points, thereby reducing the impact of human factors such as parameter selection.

3.1 Time series prediction module

In this module, firstly, for the historical data and relevant variables, we use the input attention mechanism to adaptively select the relevant input series and capture the dynamic correlations. Then, we design the external factor fusion component to learn the features of external factors from different domains, and achieve fusion through the gating mechanism based on the Gated Residual Networks (GRN). Finally, in the decoding stage, we adopt multi-head attention inspired by the Transformer architecture, which can effectively achieve feature extraction and fusion of multivariate data, as well as improve accuracy. The description of the proposed model is shown in Fig. 2.

3.1.1 Notations

In the given time series dataset, suppose the target series has n relevant variables and m historical data series. Given a time window of length T, we use $\mathbf{X} = (\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^n)^T = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T) \in \mathbb{R}^{n \times T}$ to represent the relevant variables series of length T, where $\mathbf{x}^k = \left(x_1^k, x_2^k, ..., x_T^k\right)^T \in \mathbb{R}^T$ is the k-th series, and employ $\mathbf{x}_t = \left(x_t^1, x_t^2, ..., x_t^n\right)^T \in \mathbb{R}^n$ to denote a vector of all relevant variables series at time t. In addition, we denote the historical data of the target series as $\mathbf{Y} = (\mathbf{y}^1, \mathbf{y}^2, ..., \mathbf{y}^m)^T \in \mathbb{R}^{m \times T}$, where each sub-series has the same time features as the target series.

Given the historical data, relevant variables, and external factors, to detect the change points in the target series over the next τ time, and the target series is denoted as $\hat{\mathbf{y}}_t =$

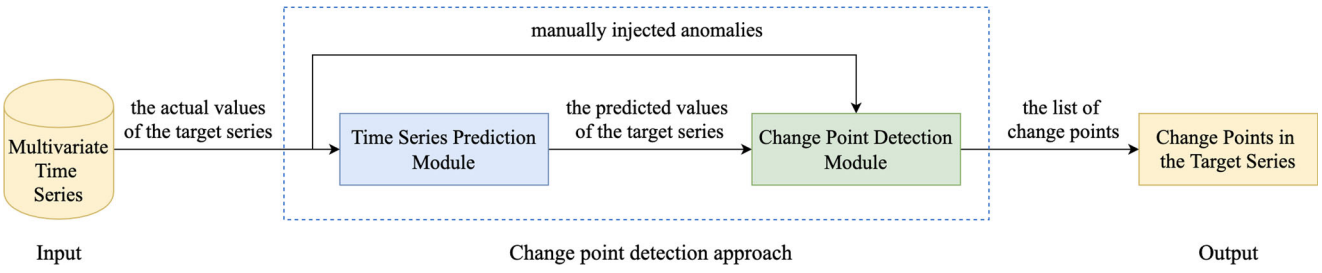


Fig. 1 The overall workflow of Finder

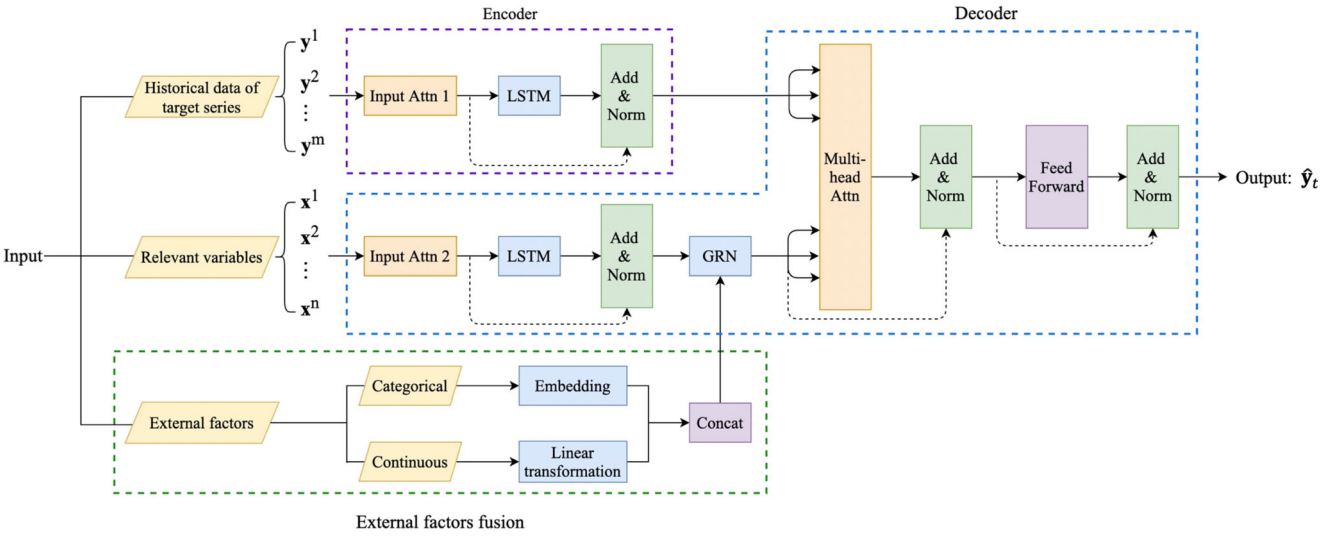


Fig. 2 The structure of the time series prediction module

$(\hat{y}_{T+1}, \hat{y}_{T+2}, ..., \hat{y}_{T+\tau})^T \in \mathbb{R}^\tau$. Among them, the relevant variables and external factors are known future inputs, and they are in the same time period as the target series.

3.1.2 Input attention mechanism

We use two input attention to process the historical data and relevant variables of the target series respectively. After the input attention, we employ LSTM units to better capture the temporal correlation of time series and improve efficiency. Each input attention combines with an LSTM unit to adaptively select relevant series and achieve feature extraction.

For the historical data, we construct a novel input attention mechanism to adaptively capture the dynamic correlations between the target series and historical data. Given the l-th series: $\mathbf{y}^l = \left(y_1^l, y_2^l, ..., y_T^l\right)^T \in \mathbb{R}^T$, by referring to the previously hidden state $\mathbf{h}_{t-1}$ and the cell state $\mathbf{s}_{t-1}$ in the encoder LSTM unit, we calculate the attention weight as follows:

$$e_t^l = \boldsymbol{\omega}_e^T \tanh(\mathbf{W}_e[\mathbf{h}_{t-1}; \mathbf{s}_{t-1}] + \mathbf{U}_e \mathbf{y}^l + \mathbf{b}_e) \tag{1}$$

and

$$\alpha_t^l = \frac{\exp(e_t^l)}{\sum_{j=1}^n \exp(e_t^j)} \tag{2}$$

where $\mathbf{W}_e \in \mathbb{R}^{T \times 2p}$ and $\mathbf{U}_e \in \mathbb{R}^{T \times T}$ are matrices, $\boldsymbol{\omega}_e, \mathbf{b}_e \in \mathbb{R}^T$ are vectors, p is the size of the hidden states for the LSTM units. They are parameters to learn.

The attention weight measures the importance of each historical data feature. With these attention weights, the output vector of input attn 1 at time step t is computed with:

$$\tilde{\mathbf{x}}_t^{his} = \left(\alpha_t^1 y_t^1, \alpha_t^2 y_t^2, ..., \alpha_t^m y_t^m\right)^T \tag{3}$$

We denote it as $\boldsymbol{\theta}_t^1$ for simplicity.

We also consider the impact of related variables in our model. To our target series, that of other variables have direct impacts on it. The impact is highly dynamic, changing over time. Since there might be some irrelevant series to the target series, directly using series of all variables as the inputs results in very high computational cost and degrades the performance.

To address this issue, we employ the input attention mechanism to capture the complex correlations between the target series and each relevant variable. Given the k-th series of relevant variables: $\mathbf{x}^k = (x_1^k, x_2^k, ..., x_T^k)^T \in \mathbb{R}^T$, we calculate the attention weight as follows:

$$g_t^k = \boldsymbol{\omega}_g^T \tanh(\mathbf{W}_g[\mathbf{h}_{t-1}; \mathbf{s}_{t-1}] + \mathbf{U}_g \mathbf{x}^k + \mathbf{b}_g) \tag{4}$$

and

$$\beta_t^k = \frac{\exp(g_t^k)}{\sum_{j=1}^n \exp(g_t^j)} \tag{5}$$

where $\mathbf{W}_g \in \mathbb{R}^{T \times 2p}$ and $\mathbf{U}_g \in \mathbb{R}^{T \times T}$ are matrices, $\boldsymbol{\omega}_g, \mathbf{b}_g \in \mathbb{R}^T$ are vectors. They are learnable parameters.

Then the output vector of the input attn 2 is computed as:

$$\tilde{\mathbf{x}}_t^{var} = \left(\beta_t^1 x_t^1, \beta_t^2 x_t^2, ..., \beta_t^n x_t^n\right)^T \tag{6}$$

3.1.3 External factors fusion component

As we mentioned, the prediction of the target series is affected by many complex external factors, such as meteorology, the weather condition. Inspired by the previous works [28, 46], we design a simple yet effective component to incorporate these factors into our model, which we call the external factor fusion component.

As shown in Fig. 2, we divide external factors into two categories. For categorical factors, since they cannot be fed to the neural networks directly, we use the entity embedding method [15] to transform each categorical attribute into a low-dimensional (p-dimensional) real vector. Comparing with the one-hot encoding, the embedding method mainly has two advantages [15]. First, our embedding method effectively reduces the input dimension and thus it is more computationally efficient. Furthermore, it helps find similar features among different series fragments.

Besides the categorical factors, we further incorporate another kind of important factor, the continuous factors. For these factors, we use linear transformations to transform them into low-dimensional vectors.

Finally, we concatenate the two kinds of obtained vectors as the output of the component, denoted as $\mathbf{ex}_t$.

3.1.4 Gating mechanisms

As relevant variables and external factors usually have a significant influence on the target series (whether this impact is direct or indirect), moreover, the correlations with the target series and their specific contribution to the output are typically unknown and complex. To address the above problems and provide flexibility to our model, we propose a Gated Residual Network (GRN) as shown in Fig. 3.

Since the relevant variables and external factors are in the same time period as the target series, they have similar temporal features. We can integrate them through the GRN module to maximize the importance of these factors.

The GRN takes in the output vector of the input attn 2 $\tilde{\mathbf{x}}_t^{var}$ and the output of external factor fusion component $\mathbf{ex}_t$ and yields:

$$\mathrm{GRN}(\tilde{\mathbf{x}}_t^{var}, \mathbf{ex}_t) = \mathrm{LayerNorm}(\tilde{\mathbf{x}}_t^{var} + \mathrm{GLU}(\boldsymbol{\eta}_1)) \tag{7}$$

$$\boldsymbol{\eta}_1 = \boldsymbol{W}_1\boldsymbol{\eta}_2 + \boldsymbol{b}_1 \tag{8}$$

$$\boldsymbol{\eta}_2 = \mathrm{ELU}(\boldsymbol{W}_2\tilde{\mathbf{x}}_t^{var} + \boldsymbol{W}_3\mathbf{ex}_t + \boldsymbol{b}_2) \tag{9}$$

where ELU is the Exponential Linear Unit activation function [11], $\boldsymbol{\eta}_1, \boldsymbol{\eta}_2 \in \mathbb{R}^p$ are intermediate layers, LayerNorm is the standard layer normalization [5].

We use component gating layers based on Gated Linear Units (GLUs) [13] to provide flexibility. The GLU takes the form:

$$\mathrm{GLU}(\boldsymbol{\eta}_1) = \sigma(\boldsymbol{W}_4\boldsymbol{\eta}_1 + \boldsymbol{b}_4) \odot (\boldsymbol{W}_5\boldsymbol{\eta}_1 + \boldsymbol{b}_5) \tag{10}$$

where $\sigma(.)$ is the sigmoid activation function, $\boldsymbol{W}_{(.)} \in \mathbb{R}^{p\times p}$, $\boldsymbol{b}_{(.)} \in \mathbb{R}^p$ are the weights and biases, and $\odot$ is an element-wise multiplication.

Through the above methods, we achieve the fusion of relevant variables and external factors and enhanced their importance. We denote the output of this module as $\boldsymbol{\theta}_t^2$.

3.1.5 Decoder with multi-head attention

Next, we capture the long-term temporal dependence through the multi-head attention module.

First, the module takes in the previously generated output vector $\boldsymbol{\theta}_t^1$ and $\boldsymbol{\theta}_t^2$, grouped them into a single matrix, denoted as $\boldsymbol{\Theta}_t = [\boldsymbol{\theta}_{t-m}, ..., \boldsymbol{\theta}_{t+\tau}]^T$. Then we apply the interpretable multi-head attention at each time step.

Our Finder employs a self-attention mechanism to learn long-term relationships across different time steps, which we modify from multi-head attention in transformer-based architectures [27, 45] to enhance explainability.

The input consists of queries and keys of dimension d_k, and values of dimension d_v. In general, attention mechanisms scale values $\boldsymbol{V}$ based on relationships between keys $\boldsymbol{K}$ and queries $\boldsymbol{Q}$ as below:

$$\mathrm{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathrm{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{11}$$

The most commonly used attention function is scaled dot-product attention [45].

To improve the learning capacity of the standard attention mechanism, multi-head attention is proposed in [45], employing different heads for different representation subspaces. It allows the model to jointly attend to

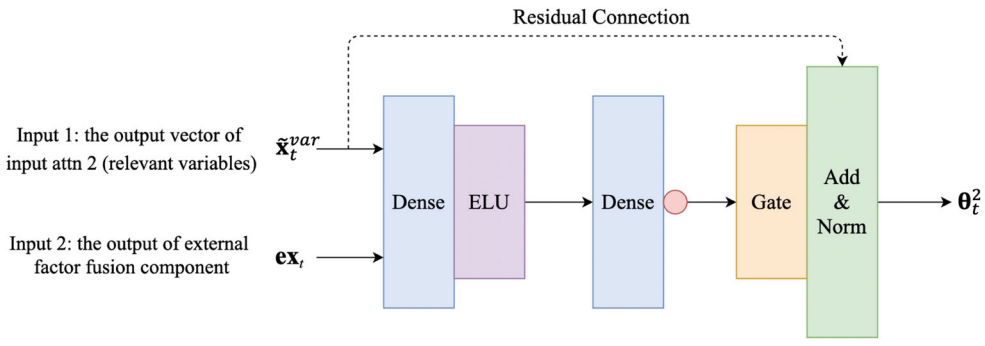


Fig. 3 The structure of Gated Residual Network

information from different representation subspaces at different positions.

$$\text{MultiHead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = [\boldsymbol{H}_1, \boldsymbol{H}_2, ..., \boldsymbol{H}_h]\boldsymbol{W}_H \tag{12}$$

$$\boldsymbol{H}_i = \text{Attention}(\boldsymbol{Q}\boldsymbol{W}^i_Q, \boldsymbol{K}\boldsymbol{W}^i_K, \boldsymbol{V}\boldsymbol{W}^i_V) \tag{13}$$

where h is the number of heads, $\boldsymbol{W}^i_Q, \boldsymbol{W}^i_K \in \mathbb{R}^{p\times d_k}$, $\boldsymbol{W}^i_V \in \mathbb{R}^{p\times d_v}$ are head-specific weights for keys, queries, and values, and $\boldsymbol{W}_H \in \mathbb{R}^{(h\cdot d_v)\times p}$ linearly combines outputs concatenated from all heads $\boldsymbol{H}_i$.

Given that different values are used in each head, attention weights alone would not be indicative of a particular feature's importance. As such, we modify multi-head attention to share values in each head and employ additive aggregation of all heads. In (11), we let $A(\boldsymbol{Q}, \boldsymbol{K}) = \text{Softmax}(\boldsymbol{Q}\boldsymbol{K}^T/\sqrt{d_k})$, so:

$$\text{FusionMultiHead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \tilde{\boldsymbol{H}}\tilde{\boldsymbol{W}}_H \tag{14}$$

$$\tilde{\boldsymbol{H}} = \tilde{A}(\boldsymbol{Q}, \boldsymbol{K})\boldsymbol{V}\boldsymbol{W}_V \tag{15}$$

$$= \left\{1/H \sum_{h=1}^{m_H} A(\boldsymbol{Q}\boldsymbol{W}^i_Q, \boldsymbol{K}\boldsymbol{W}^i_K)\right\} \boldsymbol{V}\boldsymbol{W}^i_V \tag{16}$$

$$= 1/H \sum_{h=1}^{m_H} \text{Attention}(\boldsymbol{Q}\boldsymbol{W}^i_Q, \boldsymbol{K}\boldsymbol{W}^i_K, \boldsymbol{V}\boldsymbol{W}^i_V) \tag{17}$$

where $\boldsymbol{W}_V \in \mathbb{R}^{p\times d_v}$ are value weights shared across all heads, and $\boldsymbol{W}_H \in \mathbb{R}^{d_k\times p}$ is used for final linear mapping. From (16), we see that each head can learn different temporal patterns, while attending to a common set of input features, which can be interpreted as a simple ensemble over attention weights into combined matrix $\tilde{A}(\boldsymbol{Q}, \boldsymbol{K})$ in (15). Compared to $A(\boldsymbol{Q}, \boldsymbol{K})$, $\tilde{A}(\boldsymbol{Q}, \boldsymbol{K})$ yields an increased representation capacity in an efficient way.

We apply the interpretable multi-head attention at each time step for the obtained matrix $\boldsymbol{\Theta}_t$:

$$\boldsymbol{O}(t) = \text{FusionMultiHead}(\boldsymbol{\Theta}_t, \boldsymbol{\Theta}_t, \boldsymbol{\Theta}_t) \tag{18}$$

to yield $\boldsymbol{O}(t) = [\boldsymbol{\gamma}_{t-m}, ..., \boldsymbol{\gamma}_{t+\tau}]$, and we choose $d_k = d_v = p/h$.

Decoder masking [27, 45] is applied to the multi-head attention layer to ensure that each temporal dimension can only attend to features preceding it, so preserving causal information flow. Moreover, the multi-head attention layer allows Finder to capture the long-term temporal dependence.

Following the self-attention layer, an additional gating layer is also applied to facilitate training:

$$\boldsymbol{\xi}(t) = \text{LayerNorm}(\boldsymbol{\theta}_t + \text{GLU}(\boldsymbol{\gamma}_t)) \tag{19}$$

We feed the outputs of the multi-head attention layer to a feed-forward network. In our model Finder, we apply GRNs for additional non-linear processing.

$$\boldsymbol{\delta}(t) = \text{GRN}(\boldsymbol{\xi}(t)) \tag{20}$$

Lastly, the final prediction results are produced with:

$$\hat{y}_t = \boldsymbol{\omega}_y\boldsymbol{\delta}(t) + b_y \tag{21}$$

where $\boldsymbol{\omega}_y \in \mathbb{R}^p$ and $b_y \in \mathbb{R}$ are parameters to learn.

Moreover, in the training procedure, we use the minibatch stochastic gradient descent (SGD) together with the adaptive moment estimation (Adam) optimizer [25] to optimize parameters. We implemented our approach in the TensorFlow framework.

3.2 Change point detection module

In this module, we detect change points in the target series according to the prediction results. First, we calculate the deviation between the predicted value and the actual value. Then, based on the deviation, we use a deep learning classifier to detect change points, which can improve efficiency and accuracy.

After the final prediction result $\hat{y}_t$ is produced by the time series prediction module, the predicted value is delivered to this module. The deviation between the actual and the predicted value is calculated as $l_t = y_t - \hat{y}_t$, where y_t is the actual value, and $\hat{y}_t$ is the predicted value.

The absolute value of l_t is used as the anomaly score, denoted as e_t. The larger the anomaly score, the more significant the anomaly at the given time step. Therefore, we need to judge the change point based on the anomaly score.

Next, we adopt a deep learning classifier to classify abnormal scores and detect change points. In our approach, we use a Convolutional Neural Network (CNN) as the backbone of the change point detection module. This is because CNN is currently one of the most successful and commonly used deep learning models. Due to the high efficiency and good performance, it has have been successfully applied in different domains, especially time series classification [37].

The CNN classifier we used is based on a classic structure [26]. Each convolutional layer was followed by a max-pooling layer. The last layer of the network is the fully connected (FC) layer. In addition, we use the binary cross-entropy function as the loss function.

Since the input of the module is the abnormal scores (*i.e.*, the deviation between the predicted and the actual value), the data distribution is relatively simple and clear, and there are no complicated change rules. Using the classic CNN can achieve the best performance without increasing additional complexity to the overall approach. At the same time, it reduces the impact of human factors and the dependence

Table 1 Detail of the datasets

Datasets		Load Forecasting dataset	Air-Quality dataset
Time spans		1/1/2004-6/30/2008	3/1/2013-2/28/2017
Time intervals		1 hour	1 hour
Target series		Total load (2007)	PM2.5 (3/1/2016-2/28/2017)
Historical data	Time	2004-2006	3/1/2013-2/28/2016
	Attribute	Total load	PM2.5
Relevant variables	Time	2007	3/1/2016-2/28/2017
	Attribute	Load of 8 zones	PM2.5 of 8 monitoring sites
External factors	Time	2007	3/1/2016-2/28/2017
	Attribute	Temperature	PM10, Temperature

on parameter selection, which balances both efficiency and accuracy, and is beneficial to improve the detecting performance.

4 Experiment and evaluation

Based on the above approach, we designed the following experiment scheme. In this section, we conducted experiments on the two modules of the proposed Finder on two real-world datasets. Extensive results show the effectiveness and superiority of our approach.

4.1 Experimental environments

All the experiments were executed on a single computer, and the details of the experimental environments are shown below:

The hardware configuration: 3.1GHz Intel Core i5 CPU, 16 GB 2133 MHz LPDDR3, Intel Iris Plus Graphics 650 1536 MB.

The software details: Model implementation: Python 3.7.0, TensorFlow 1.13.2, Keras 2.1.0; Operating System: macOS Catalina 10.15.3.

4.2 Datasets introduction

To demonstrate the effectiveness of Finder, we conducted experiments on two publicly real-world datasets: the Load Forecasting dataset and the Air-Quality dataset. As depicted in Table 1, each dataset contains three sub-datasets: historical data, relevant variables data, and external factors data.

In our experiment, we used the first 80% data points as the training set, the following 10% data points as the validation set, and the last 10% data points as the test set.

1. Load Forecasting dataset: [1]

   The dataset records the hourly load data from utilities in the United States, including the total load, the load of 20 zones, and the weather information (*e.g.* temperature). This dataset was used in the Global Energy Forecasting Competition held in 2012 (GEFCom 2012) [20, 40]. We set the total load in 2007 as the target series, and chose the load of 8 zones with similar magnitude in 2007 as the relevant variables.

   Moreover, the original dataset contains missing values. So we manually injected anomalies in the missing values' locations and marked them.
2. Air-Quality dataset: [2]

   The dataset can be found in the UCI (University of California, Irvine) Machine Learning Repository. It includes hourly data of 6 main air pollutants and 6 relevant meteorological variables at 12 air-quality monitoring sites in Beijing and contains 420,768 instances with 18 attributes. Among them, the primary pollutant of air quality is PM2.5 in most cases, thus we employed its reading as the target series. And we selected 8 geographically representative monitoring sites and use their PM2.5 data as relevant variables.

   Moreover, the methods of anomalies inserting and marking are similar to the Load Forecasting dataset.

4.3 Results-I: Prediction performance

In this section, we demonstrated the effectiveness of the time series prediction module of Finder through extensive experiments on two public datasets.

4.3.1 Competing methods

To demonstrate the effectiveness of Finder in predicting performance, we compared it against four baseline methods.

[1] https://www.kaggle.com/c/global-energy-forecasting-competition-2012-load-forecasting/data

[2] http://archive.ics.uci.edu/ml/datasets/Beijing+Multi-Site+Air-Quality+Data

- LSTM (Long Short-Term Memory) network [19] is a classic deep learning method to address time series prediction.
- Seq2seq [43] model is an encoder-decoder network. It uses an RNN to encode the input sequences into a feature representation and another RNN to make predictions iteratively.
- Attention [6] is an attention-based encoder-decoder network. It is suitable for solving the complex mapping relationship between dynamic series.
- DA-RNN [33] is a dual-stage attention-based recurrent neural network, which shows state-of-the-art performance in time series prediction.

4.3.2 Evaluation metrics

To measure the effectiveness of various methods for the time series prediction module, we considered three different evaluation metrics: root mean squared error (**RMSE**), mean absolute error (**MAE**), and mean absolute percentage error (**MAPE**). For each metric, the smaller the value, the greater the performance to predict.

4.3.3 Parameter settings

There are two parameters in our approach, *i.e.*, the length of time steps T, and the size of hidden states for the encoder p.

To determine the length of time steps T, we conducted a grid search over $T \in \{6, 12, 18, 24, 30\}$. We plotted the MAPE versus different lengths of time steps in Fig. 4a. It is easily observed that on the Load Forecasting dataset, our approach achieved the best performance when $T = 24$, while on the Air-Quality dataset, $T = 12$.

For the size of hidden states for encoder, we conduct grid search over $p = q \in \{16, 32, 64, 128\}$. We plotted the MAPE versus different sizes of hidden states in Fig. 4b. On the Load Forecasting dataset, we set $T = 24$ and observed that Finder achieves the best performance when $p = 64$. Similarly, on the Air-Quality dataset, by setting $T = 12$, we can determine that $p = 32$.

In addition, the other hyperparameters tuning are as follows: we set *learning rate* = 0.001, *batch_size* = 128. The above settings enable our approach to achieve stable and best performance across the provided datasets.

In order to ensure the fairness of the comparison experiments, all the competing methods use the same parameter settings as Finder.

Moreover, the reported metrics are averaged results over 10 independent runs.

4.3.4 Effectiveness verification

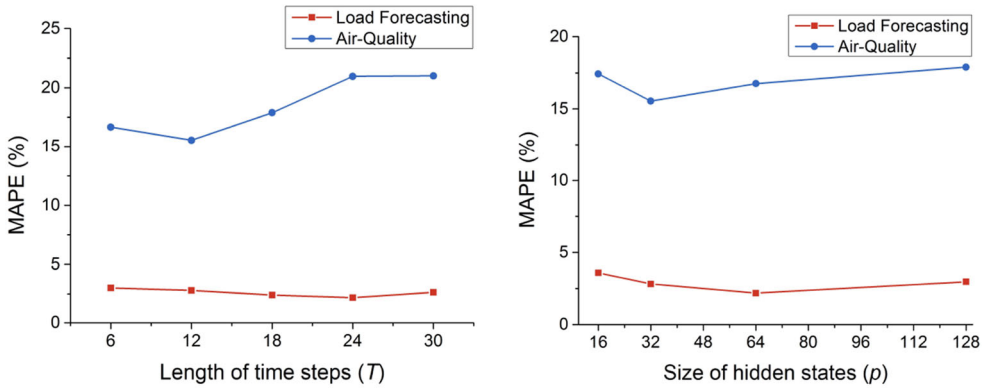
To investigate the effectiveness of Finder, we compared different combinations of various inputs, proved the importance of each type of input, so that verified the necessity of each model component. The details as follows:

- **his**: There is only historical data in the input.
- **nv**: The input contains no relevant variables, only historical data and external factors.
- **sv**: We added a single variable on the basis of nv to verify the importance of multi-variable.
- **ne**: This variant does not consider the effects of external factors. i.e., no external factors in the input.

We conducted experiments for Finder and the variants on the two real-world datasets. Table 2 summarizes the results.

From the table, we observe that the combination of relevant variables and external factors in the input shows great superiority in multivariate time series analysis.

This is because relevant variables and external factors have the same time features as the target series, especially relevant variables, which have a more direct and important impact on the target series. According to **nv** and **sv**, we add a single variable on the basis of historical data and external factors, and MAPE improved by 43.66% and 15.52% on the two datasets. And according to **nv** and our approach, through combining multiple variables, MAPE improved by 74.75% and 62.29% respectively.



(a) Length me steps T (b) Size of hidden sta r the encoder p

Fig. 4 MAPE vs. different parameters on two real-world datasets

Table 2 The predicting performance of Finder and other variants

Datasets	Load Forecasting dataset			Air-Quality dataset		
	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
his	279.632	230.469	12.844	29.578	16.474	59.446
nv	198.124	165.183	8.669	21.652	12.343	41.220
sv	111.724	91.975	4.884	21.235	11.046	34.823
ne	56.925	42.550	2.226	15.315	7.476	17.517
Finder	**55.829**	**41.992**	**2.189**	**15.365**	**7.243**	**15.545**

Bold entries are the experimental results of our approach and are better than other baseline methods

Besides, as an essential component of the input in our approach, external factors provide additional valuable information to avoid the potential negative effect on the target series, as well as boost the predictive performance.

Therefore, we extract features and learn the change rules for multiple relevant variables and external factors, which is of great significance for learning the dynamic correlations of the target series and improving the prediction accuracy.

4.3.5 Performance comparison

To further demonstrate the superiority of Finder, we compared it with all baseline methods on the two real-world datasets. Table 3 summarizes the results.

Since our datasets contain complex mixed inputs, including the target series' historical data, relevant variables, and external factors. In the competing methods, LSTM, Seq2seq, and Attention cannot process multi-input series, nor do they achieve the external factors fusion. Therefore, for the three methods, we use historical data and relevant variables as input, respectively.

Table 3 shows that our approach outperforms the other baseline models markedly on the two datasets. Compared with the state-of-the-art method, on the Load Forecasting dataset, the RMSE, MAE, and MAPE improved by 14.32%, 18.54%, and 18.38% respectively. Compared to the relatively stable power load data, the concentration of PM2.5 sometimes fluctuates tremendously, which makes it more difficult to forecast. But our approach still achieves the best performance, and the three metrics are improved by 8.70%, 11.42%, and 23.29% respectively. These experimental results demonstrated the superiority of our approach.

Specifically, in Table 3, we observed that for LSTM, Seq2seq, and Attention, the results using relevant variables as input generally better than those using historical data. On the two datasets, the MAPE for the three methods has improved by 57.04% and 38.98% on average. This is because the relevant variables have the same time features as the target series, and the correlations between them are stronger. So the impact on the target series is more direct and important.

For DA-RNN and Finder, due to employing the input attention mechanism, they can handle multiple relevant series. They improve prediction accuracy through adaptive selecting and learning more features. Therefore, they consistently outperform the other three methods.

For DA-RNN, despite that different types of data in the mixed input have different impacts on the target series, it treats all these time series as equal. DA-RNN just simply concatenates the time-dependent features in the input together, and directly feeds them into the encoder

Table 3 The predicting performance of Finder and baseline methods

Models	Load Forecasting dataset			Air-Quality dataset		
	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
LSTM (his)	302.188	239.607	12.644	30.908	16.613	60.351
LSTM (var)	125.736	103.835	5.655	31.152	16.396	39.248
Seq2seq (his)	312.385	252.877	13.243	35.166	17.439	61.260
Seq2seq (var)	120.393	99,826	5.433	31.595	16.667	35.426
Attention (his)	283.658	224.017	11.982	33.123	16.819	56.750
Attention (var)	115.573	94.438	5.168	30.300	16.019	34.159
DA-RNN	65.159	51.551	2.682	16.829	8.177	20.264
Finder	**55.829**	**41.992**	**2.189**	**15.365**	**7.243**	**15.545**

Bold entries are the experimental results of our approach and are better than other baseline methods

to select relevant series by input attention. Therefore, DA-RNN unable to achieve feature extraction and fusion of multivariate data through learning the dynamic correlations.

For further comparison, we showed the prediction results of Finder on the two datasets in Fig. 5. The blue line represents the actual value, the orange line represents the training part, and the green line represents the test part. We can observe that our model performs well in prediction performance.

In summary, Finder employs the multivariate fusion attention networks, which integrates multi-level attention, multi-head attention, and the external factor fusion component. It aims to process different types of input data, achieve feature extraction and the fusion of multivariate data. Our approach achieves the best performance on the two datasets. After that, we output the predicted values for the experiments in the next step.

4.4 Results-II: Change point detection performance

In this section, we proved the performance of the change point detection module of Finder on the two real-world datasets. By comparing it with the other three state-of-the-art approaches, we demonstrated the superiority of our approach.

4.4.1 Competing methods

As we introduced in Section 1, Finder is a change point detection approach that combines the methods of prediction and classification in the same deep learning-based framework. To demonstrate the effectiveness of this novel idea in change point detection, we compared it against three baseline methods.

- We selected Bayesian online change point detection (BOCPD) as a baseline [1], which is the classic probabilistic method of time series change point detection.
- CNN-LSTM neural network is the latest and commonly used hybrid model [24]. It combines CNN, LSTM, and deep neural networks, which can extract complex features effectively and model spatial and temporal information. It was proposed for anomaly detection, but it has inspiration for our problem. So we applied it to change point detection and selected it as the classification-based baseline.
- We use the dual-stage attention-based recurrent neural network (DA-RNN) [33] to predict the target series, and then use the Gaussian distribution-based method to determine the threshold and detect change points. This is the idea of the traditional prediction-based method, and we selected it as one of our baselines.

4.4.2 Evaluation metrics

We evaluated the efficiency of the change point detection module based on the following metrics: **Precision**, **Recall**, and **F1 score**. For each metric, the larger the value, the greater the ability to detect change points.

4.4.3 Results and analysis

Based on the previous experiments, we calculated the deviation between predicted and actual values as the input of this module. To measure the change point detecting performance of Finder, we conducted extensive experiments on Finder and other baseline methods on the two real-world datasets. Table 4 and Fig. 6 summarizes the results.

Table 4 shows that on the two real-world datasets, our approach outperforms the other baseline methods in performance. Among them, on the Load Forecasting dataset, the Precision, Recall, and F1 score improved by 5.59%, 7.50%, and 6.55% respectively. And on the Air-Quality dataset, the three metrics improved by 9.11%, 11.91%, and 10.50% respectively.

Figure 6 depicts the change point detection results of Finder over the two datasets. We can easily observe that

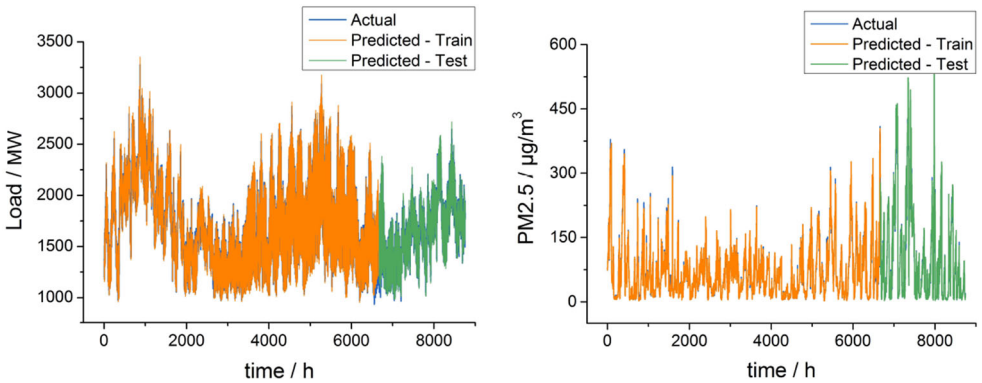


Fig. 5 The final predicting results of Finder over the two real-world datasets

(a) Load Forecasting dataset

(b) Air-Quality dataset

Table 4 The change point detecting performance of Finder and baseline methods

Models	Load Forecasting dataset			Air-Quality dataset		
	Precision	Recall	F1	Precision	Recall	F1
BOCPD	0.912	0.920	0.916	0.900	0.873	0.886
CNN-LSTM	0.943	0.917	0.930	0.889	0.917	0.903
DA-RNN	0.943	0.975	0.959	0.941	0.937	0.939
Finder	**0.963**	**0.989**	**0.976**	**0.982**	**0.977**	**0.979**

Bold entries are the experimental results of our approach and are better than other baseline methods

performance of Finder in detecting change points is well, it can detect change points accurately and timely. In the Load Forecasting dataset, the power load data is relatively stable, and change rules over time are more obvious. In the Air-Quality dataset, the concentration of PM2.5 usually fluctuates tremendously, which makes it more difficult to learn the change rules and accurately detect change points. But our approach still achieves the best performance on both datasets, and the performance improvement of the Air-Quality data set is even greater.

This is because Finder is a prediction-based approach, which employs the multivariate fusion attention networks in the prediction module, integrating multi-level attention and multi-head attention. For the data with large fluctuations and weak regularity such as the concentration of PM2.5, it is able to learn complex dynamic correlations and achieve feature extraction. Thereby, it will greatly improve the prediction accuracy, and then have a significant impact on the improvement of the change point detection accuracy.

Moreover, we use a deep learning classifier in the change point detection module. It reduces the interference of human factors and the dependence on the parameters (*i.e.* threshold) selection, balances efficiency and accuracy, and improves the change point detection performance.

In conclusion, our approach combines the ideas of prediction and classification in the same deep learning-based framework for change point detection in multivariate time series. Extensive experimental results prove the superiority of Finder compared with the state-of-the-art methods.

5 Related work

In this section, we introduced the widely used change point detection methods and their current status. According to the research [3, 9, 39], the change point detection problem is usually based on the following three types of methods.

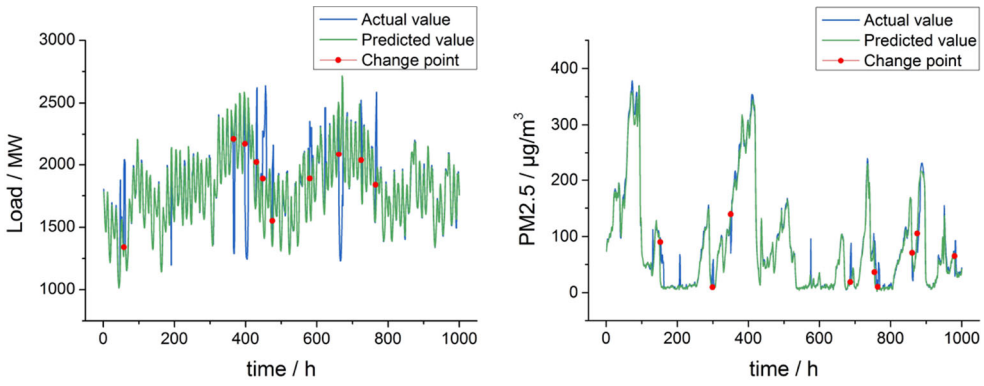
5.1 Probability and statistics-based methods

Classical time series change point detection approaches are mainly probability-based or statistical methods, such as the Bayesian method [1, 51], Gaussian Process [10], density ratio estimation [23], and wavelet transform.

These methods have simple processes, fast speed, and low computation complexity. However, these methods depend on specific prior knowledge or strong assumption, leading to poor performance, and producing unstable, even ruinous results in complex real-world applications.

5.2 Classification-based methods

Classification-based change point detection approaches are mainly divided into the following two types.



(a) Load Forecasting dataset (b) Air-Quality dataset

Fig. 6 The change point detection results of Finder on real-world datasets

The traditional machine learning algorithms are mostly supervised algorithms [3], including Decision Trees, Naive Bayes, Bayesian Net, K-Nearest Neighbors, Random Forests, etc. These methods are fast and effective in the calculation, and the principle is easy to understand and explain. The disadvantage is that for the high-dimensional or large-scale datasets, the computational complexity is high and the performance is weak.

More recently, deep learning-based methods have been widely applied for change point detection to improve performance [9]. Current popular methods include convolutional neural networks (CNN)-based [31, 37, 54], recurrent neural networks (RNN)-based [22], and variations or hybrids-based. For example, Kim et al. proposed a CNN-LSTM neural network [24], which is a classic and commonly used hybrid model. By combining a CNN, LSTM, and deep neural network, it can extract complex features effectively and model the spatial and temporal information. Inspired by this, Canizo et al. established a multi-head CNN-RNN architecture for processing multivariate time series [8].

However, the classification-based methods do not consider the unique features of the multivariate time series, such as the heterogeneity of data sources or temporal correlations. More importantly, these methods require a sufficient amount and diversity of data with a large number of annotations in the training process, aiming to cover the comprehensive data types and anomalies. It is extremely difficult to achieve in actual scenarios because such datasets are very limited.

5.3 Prediction-based methods

The prediction-based method is one of the most commonly used methods in time series change point detection. The basic idea is to determine whether it is abnormal by comparing the deviation between the predicted value and the true value. According to the research in [30, 39], the main methods are as follows:

Traditional statistical methods include the Autoregressive Moving Average (ARMA) model, Autoregressive Integrated Moving Average (ARIMA) [4] model, and their variations. Facebook proposed the open-source algorithm Prophet [44], which takes into account the holiday factors and the periodicity of data.

These methods have simple models, few parameters, and only rely on historical data, so they are fast and interpretable. But the disadvantages are also obvious. They cannot model nonlinear relationships, and more importantly, they cannot add other relevant/driving input series, and the prediction results are too dependent on the selection of parameters.

In recent years, the research on change point detection with deep learning has made great progress. Deep neural networks (DNNs) have increasingly been used in time series modeling, demonstrating strong performance improvements in capturing complex temporal patterns of dynamic time series [2, 35, 47]. Current methods include Recurrent Neural Networks (RNNs)-based [38, 49], Long Short-Term Memory Networks (LSTMs)-based [16], and variations or hybrids-based.

In recent progress, attention-based models have been used to enhance the learning ability of complex correlations [6, 41]. Fan et al. proposed an end-to-end deep-learning framework for multi-horizon time series forecasting [14]. It handles multimodal data by fusing features of different modalities based on attention. Qin et al. proposed a dual-stage attention-based recurrent neural network (DA-RNN) to adaptively select the relevant driving series, and extract features [33]. Liang et al. extended the DA-RNN and applied it to geo-sensory time series forecasting, which can capture the spatial correlations between different sensors [28].

Moreover, the well-known self-attention-based Transformer [45] has recently been proposed for sequence modeling and has achieved great success. Several recent works apply it to translation, speech, music, and image generation. Transformer has inherent interpretability for sequential data, making it more suitable for grasping the recurring patterns with long-term dependencies. Li et al. successfully apply Transformer architecture to time series forecasting and validate its potential value in better handling long-term dependencies [27]. Huang et al. proposed a dual self-attention network (DSANet) for multivariate time series forecasting [21], which is inspiring for our research. DSANet firstly divided the multivariate time series into multiple univariate sub-series as input. In the encoding stage, it utilizes two parallel convolutional components, called global temporal convolution and local temporal convolution, to capture complex mixtures of global and local temporal patterns, and employs a self-attention module to model dependencies between multiple series in the decoding stage.

However, these methods usually fail to consider different types of data in complex mixed inputs, or neglect important external factors. They just simply concatenate the time-dependent features in the input data together. Thus, the existing methods cannot effectively learn the complex correlations between different types of inputs or achieve feature extraction and fusion.

Based on the above, for the problem of multivariate time series change point detection, employing the prediction-based deep learning approach is our best option at present. Therefore, the approach proposed in this paper focuses on accuracy and efficiency, and aims to solve the main difficulties comprehensively of the current change point detection.

6 Conclusion

In this paper, we propose Finder, which employs the novel multivariate fusion attention networks. It is used for detecting change points in multivariate time series based on heterogeneous data containing mixed inputs. Finder uses multi-level attention networks based on the transformer architecture and integrates the external factor fusion component to achieve feature extraction and the fusion of multivariate data in the time series prediction module. And based on the deviation between the actual and the predicted value, Finder uses a deep learning classifier to detect change points in the change point detection module. We verify the effectiveness and superiority of Finder on two public real-world datasets. Extensive experimental results show that our approach achieves the best performance against the state-of-the-art methods. In future work, we will further extend our approach to handle the spatial dependency between data. Moreover, exploring how to improve the interpretability of the model deserves our in-depth study.

References

1. Adams RP, MacKay DJ (2007) Bayesian online changepoint detection. Stat 1050:19
2. Alaa AM, van der Schaar M (2019) Attentive state-space modeling of disease progression. In: Advances in neural information processing systems, pp 11,338–11,348
3. Aminikhanghahi S, Cook DJ (2017) A survey of methods for time series change point detection. Knowl Inf Syst 51(2):339–367
4. Asteriou D, Hall SG (2011) Arima models and the box–jenkins methodology. Appl Econom 2(2):265–286
5. Ba J, Kiros JR, Hinton GE (2016) Layer normalization. Stat 1050:21
6. Bahdanau D, Cho K, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. In: 3rd International conference on learning representations, ICLR 2015
7. Bouktif S, Fiaz A, Ouni A, Serhani MA (2020) Multi-sequence lstm-rnn deep learning and metaheuristics for electric load forecasting. Energies 13(2):391
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. Chalapathy R, Chawla S (2019) Deep learning for anomaly detection: Asurvey
10. Chandola V, Vatsavai RR (2011) A gaussian process based online change detection algorithm for monitoring periodic time series. In: Proceedings of the 2011 SIAM international conference on data mining, pp. 95–106. SIAM
11. Clevert DA, Unterthiner T, Hochreiter S (2015) Fast and accurate deep network learning by exponential linear units (elus). Computer ence
12. Dadashova B, Li X, Turner S, Koeneman P (2021) Multivariate time series analysis of traffic congestion measures in urban areas as they relate to socioeconomic indicators. Socio-Economic Planning Sciences 75:100877
13. Dauphin YN, Fan A, Auli M, Grangier D (2017) Language modeling with gated convolutional networks. In: International conference on machine learning, pp 933–941
14. Fan C, Zhang Y, Pan Y, Li X, Zhang C, Yuan R, Wu D, Wang W, Pei J, Huang H (2019) Multi-horizon time series forecasting with temporal attention learning. In: Proceedings of the 25th ACM SIGKDD International conference on knowledge discovery & data mining, pp 2527–2535
15. Gal Y, Ghahramani Z (2016) A theoretically grounded application of dropout in recurrent neural networks. In: Advances in neural information processing systems, pp 1019–1027
16. Guo T, Lin T, Antulov-Fantulin N (2019) Exploring interpretable lstm neural networks over multi-variable data. In: International conference on machine learning, pp 2494–2504
17. Hayashi T, Fujita H (2021) Cluster-based zero-shot learning for multivariate data. J Ambient Intell Human Comput 12(2):1897–1911
18. Hernandez-Matamoros A, Fujita H, Hayashi T, Perez-Meana H (2020) Forecasting of covid19 per regions using arima models and polynomial functions. Appl Soft Comput 96(106):610
19. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
20. Hong T, Pinson P, Fan S, Zareipour H, Troccoli A, Hyndman RJ (2016) Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond
21. Huang S, Wang D, Wu X, Tang A (2019) Dsanet: Dual self-attention network for multivariate time series forecasting. In: Proceedings of the 28th ACM international conference on information and knowledge management, pp 2129–2132
22. Hundman K, Constantinou 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
23. Kawahara Y, Sugiyama M (2012) Sequential change-point detection based on direct density-ratio estimation. Stat Anal Data Min 5(2):114–127
24. Kim TY, Cho SB (2018) Web traffic anomaly detection using c-lstm neural networks. Expert Syst Appl 106:66–76
25. Kingma D, Ba J (2014) Adam: A method for stochastic optimization. Computer Science
26. LeCun Y, Bengio Y et al (1995) Convolutional networks for images, speech, and time series. Handbook Brain Theory Neural Netw 3361(10):1995
27. Li S, Jin X, Xuan Y, Zhou X, Chen W, Wang Y, Yan X (2019) Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In: Advances in Neural information processing systems, pp 5243–5253
28. Liang Y, Ke S, Zhang J, Yi X, Zheng Y (2018) Geoman: Multi-level attention networks for geo-sensory time series prediction. In: IJCAI, pp 3428–3434
29. Lv Z, Xu J, Zheng K, Yin H, Zhao P, Zhou X (2018) Lc-rnn: a deep learning model for traffic speed prediction. In: Proceedings of the 27th international joint conference on artificial intelligence, pp 3470–3476
30. Makridakis S, Spiliotis E, Assimakopoulos V (2020) The m4 competition: 100,000 time series and 61 forecasting methods. Int J Forecast 36(1):54–74
31. Munir M, Siddiqui SA, Dengel A, Ahmed S (2018) Deepant: A deep learning approach for unsupervised anomaly detection in time series. IEEE Access 7:1991–2005
32. Pan Z, Liang Y, Wang W, Yu Y, Zheng Y, Zhang J (2019) Urban traffic prediction from spatio-temporal data using deep meta learning. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp 1720–1730

33. Qin Y, Song D, Cheng H, Cheng W, Jiang G, Cottrell GW (2017) A dual-stage attention-based recurrent neural network for time series prediction. In: Proceedings of the 26th international joint conference on artificial intelligence, pp 2627–2633
34. Qiu J, Jammalamadaka SR, Ning N (2020) Multivariate time series analysis from a bayesian machine learning perspective. Ann Math Artif Intell 88(10):1061–1082
35. Rangapuram SS, Seeger MW, Gasthaus J, Stella L, Wang Y, Januschowski T (2018) Deep state space models for time series forecasting. In: Advances in neural information processing systems, pp 7785–7794
36. Ribeiro GT, Mariani VC, dos Santos Coelho L (2019) Enhanced ensemble structures using wavelet neural networks applied to short-term load forecasting. Eng Appl Artif Intel 82:272–281
37. Sadouk L (2018) Cnn approaches for time series classification. In: Time series analysis-data, methods, and applications. IntechOpen
38. Salinas D, Flunkert V, Gasthaus J, Januschowski T (2019) Deepar: Probabilistic forecasting with autoregressive recurrent networks. International Journal of Forecasting
39. Sezer OB, Gudelek MU, Ozbayoglu AM (2020) Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. Appl Soft Comput 90(106):181
40. Silva L (2014) A feature engineering approach to wind power forecasting: Gefcom 2012. Int J Forecast 30(2):395–401
41. Song H, Rajan D, Thiagarajan JJ, Spanias A (2018) Attend and diagnose: Clinical time series analysis using attention models. In: 32nd AAAI Conference on artificial intelligence, AAAI 2018, pp. 4091–4098. AAAI press
42. 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: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp 2828–2837
43. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. In: Advances in neural information processing systems, pp 3104–3112
44. Taylor SJ, Letham B (2018) Forecasting at scale. Am Stat 72(1):37–45
45. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008
46. Wang D, Zhang J, Cao W, Li J, Zheng Y (2018) When will you arrive? estimating travel time based on deep neural networks. In: AAAI, vol 18, pp 1–8
47. Wang Y, Smola A, Maddix D, Gasthaus J, Foster D, Januschowski T (2019) Deep factors for forecasting. In: International conference on machine learning, pp 6607–6617
48. Wei WW (2018) Multivariate time series analysis and applications. John Wiley Sons, New York
49. Wen R, Torkkola K, Narayanaswamy BM (2017) A multi-horizon quantile recurrent forecaster. Stat 1050:29
50. Xu H, Chen W, Zhao N, Li Z, Bu J, Li Z, Liu Y, Zhao Y, Pei D, Feng Y et al (2018) Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In: Proceedings of the 2018 world wide web conference, pp 187–196
51. Zhang A, Paisley J (2018) Deep bayesian nonparametric tracking. In: International conference on machine learning, pp 5833–5841
52. Zhang J, Zheng Y, Qi D (2017) Deep spatio-temporal residual networks for citywide crowd flows prediction. In: Proceedings of the AAAI conference on artificial intelligence, vol 31
53. Zheng J, Xu C, Zhang Z, Li X (2017) Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network. In: 2017 51st Annual conference on information sciences and systems (CISS), pp 1–6. IEEE
54. Zheng Y, Liu Q, Chen E, Ge Y, Zhao JL (2014) Time series classification using multi-channels deep convolutional neural networks. In: International conference on web-age information management, pp 298–310. Springer

Haizhou Du received his Ph.D degree in computer science and technology from Tongji University , Shanghai, China. His researchinterests include Machine Learning, Data Management, Distributed System.

Ziyi Duan received the B. Eng. degree in software engineering from Taiyuan University of Technology, Taiyuan, China, in 2018, and the M.Eng. degree in computer science and technology from Shanghai University of Electric Power, Shanghai, China, in 2021.His research interests include big data analysis and Artificial Intelligence for IT Operations (AIOps).