Multivariate Time Series Prediction Based on Temporal Change Information Learning Method

Wendong Zheng, *Graduate Student Member, IEEE*, and Jun Hu, *Member, IEEE*

***Abstract*— In the multivariate time series prediction tasks, the impact information of all nonpredictive time series on the predictive target series is difficult to be extracted at different time stages. Through the emphasis on optimal-related sequences in the target series, the deep learning model with the attention mechanism achieves a good predictive performance. However, temporal change information in the objective function and optimization algorithm is completely ignored in these models. To this end, a temporal change information learning (CIL) method is proposed in this article. First, mean absolute error (MAE) and mean squared error (MSE) losses are contained in the objective function to evaluate different amplitude errors. Meanwhile, the second-order difference technology is used in the correlation terms of the objective function to adaptively capture the impact of the abrupt and slow change information in each series on the target series. Second, the long short-term memory (LSTM) network with the transformation mechanism is used in the method so that temporal dependence information can be fully extracted (i.e., avoiding the supersaturation region). Third, to effectively obtain the optimal model parameters, the current and historical moment estimation information is adaptively memorized without the introduction of additional hyperparameters, and therefore, the acquisition ability of temporal change information in the error gradient flow is greatly enhanced by the proposed optimization algorithm. Finally, three datasets with different scales are used to verify the advantages of the CIL method in computational overhead and prediction effect.**

***Index Terms*— Abrupt and slow change information, adaptive stochastic optimization algorithm, long short-term memory (LSTM), multivariate time series prediction.**

I. Introduction

TIME series prediction is a crucial research field involving different applications in different domains, such as household appliance energy consumption systems [1]–[3], air quality monitoring systems [4], traffic flow monitoring systems [5], and social network [6]. In these systems, historical time series data are used by data-driven deep learning (DL) models to obtain high-precision predictions at the state-of-the-art (SOTA) levels. Multivariate time series refers to a dataset that contains varying multiple attribute columns over time [7], and a predicted series can be considered as a target series. With the continuous increase in the dimension and scale of time series data, the impact of multiple nonpredictive series on the target predictive series and the temporal dependence information within each series has become increasingly complex. How to improve the prediction accuracy by learning the temporal change information in historical data has been an important challenge for DL models. The change law of the target series in historical data and the impact information of the nonpredictive time series on the target series can be analyzed to obtain higher prediction accuracy. The complex impact information (i.e., the temporal dependence information generated by the nonpredicted series on the target series) in time series data can be classified into the following two types of coarse-grained information: 1) abrupt change information, which rises and falls rapidly over several time steps, and 2) slow change information, which increases or decreases monotonously and slowly in several time steps. As shown in Fig. 1, the complex temporal impact information on target series is identified by arrows and lines for the intuitive expression. For example, the short-term heavy snowfall (target series) in winter is not conducive to the dispersion of PM2.5 pollution particles in the wind (such as breeze of northwest wind direction), resulting in the rapid rise of the PM2.5 index (target series) in this region. In the summer, rainfall events (abrupt change) are more frequent, and the abrupt change of the PM2.5 index occurs frequently due to the combined impact of wind direction, wind speed, and rainfall (abrupt changes). It is likely that the abrupt change events resulting from the interaction of multiple nonpredictive time series will occur again in the future. To the best knowledge, a typical DL system usually

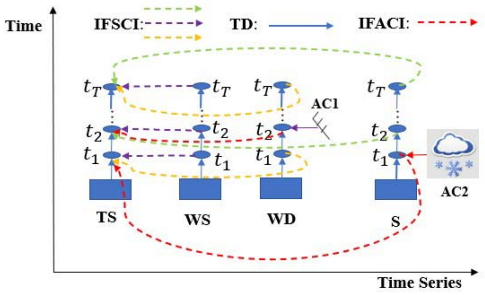


Fig. 1. Complex temporal influence information on the target series on PM2.5 Air Quality dataset. The abbreviations are introduced as follows: 1) TS represents the PM2.5 index column as a target series; 2) WS represents the wind speed column; 3) WD represents the wind direction column; 4) S represents the snow column; 5) IFSCI represents the influence from slow change information (different colored lines indicate that the influence information comes from different nonpredictive time series); 6) TD represents the temporal dependence information; 7) IFACI represents the influence from abrupt change information; and 8) AC1 and AC2, respectively, represent the abrupt change event information in the wind direction and snow time series.

Manuscript received March 31, 2020; revised September 15, 2020, April 8, 2021, and September 2, 2021; accepted December 15, 2021. This work was supported in part by the Natural Science Foundation of Hunan Province of China under Grant 2021JJ30136 and in part by the Changsha Science and Technology Plan Project under Grant kq2004015. *(Corresponding author: Jun Hu.)*

The authors are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: wendongz@hnu.edu.cn; hujun_111@hnu.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TNNLS.2021.3137178.

Digital Object Identifier 10.1109/TNNLS.2021.3137178

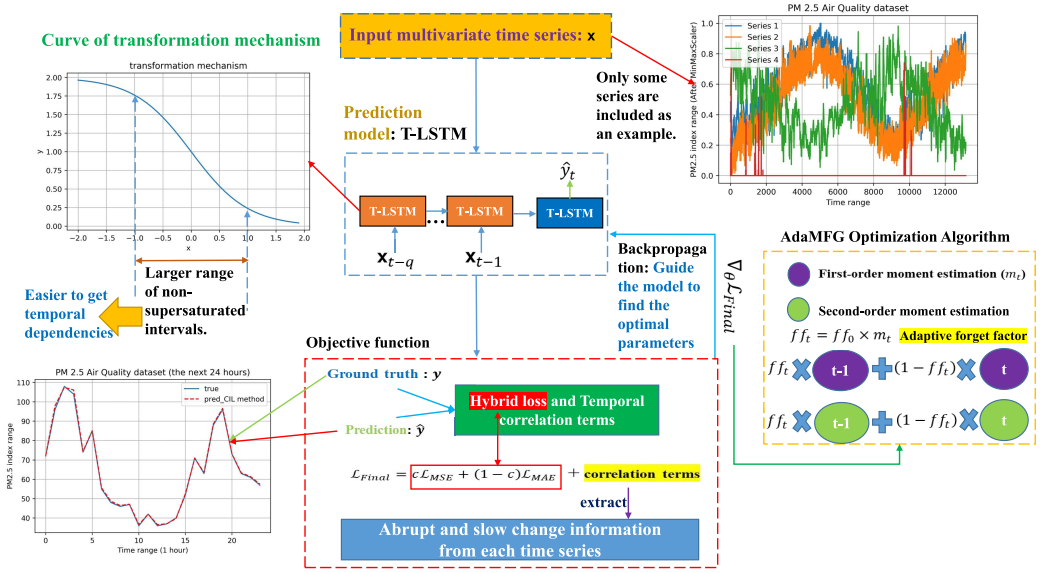


Fig. 2. Framework of CIL method.

consists of an objective function, a neural network model, and an optimization algorithm. The above-mentioned composition structure also exists in time series prediction research based on different DL methods. The strengths and limitations of some related research in these areas are outlined here (a more detailed discussion is in Section II).

A. Objective Function

Euclidean distance loss (such as mean squared errors (MSEs)) or dynamic time warping (DTW) loss [13] is mainly used as an objective function of the DL model for time series prediction. However, the possible abrupt change information of the data along the time axis is ignored in MSE loss. The Timing error can extract the temporal change information like the detection problem of change points (for instance, a loss function based on temporal distortion index (TDI) is proposed for time series error estimation [14]). However, the above variant methods based on DTW loss are not differentiable (backpropagation calculation is not allowed). Since nondifferentiable DTW loss cannot be combined with the DL model, differentiable DTW losses are proposed in some studies. Nevertheless, these methods only support prediction tasks for binary time series data [15], [16].

B. Neural Network Model

The abrupt change events generated by nonpredictive time series are not marked in the dataset. Distinctly, such change information is difficult to be captured by temporal rules of historical data in the recurrent neural networks (RNNs) [8] and their variant long short-term memory (LSTM) [9] or gated recurrent units (GRUs) [10] models in early prediction research of time series. Thereafter, the DA-RNN [11] model based on the attention mechanism can learn the long-term dependence information through the two-stage attention mechanism. Specifically, in the DA-RNN, the exogenous sequence, which is most relevant to the target sequence, is captured through the input attention mechanism, and periodic change rules of the target series along the time axis are learned through the temporal attention mechanism. Due to the inequivalent input features, the GeoMAN [12] model is proposed to further capture these features using local and global attention mechanisms. However, a large number of attention components and hyperparameters need to be tuned inside the model, leading to the huge training cost and the computational overhead for searching the best hyperparameters. How to reduce the computational overhead has become a severe challenge for large-scale time series prediction. In addition, nonpredictive sequences that are most relevant to the target series are only emphasized in the above methods, while the sequences with lower relevant that may also provide greater influence information to the target series are ignored. Therefore, the comprehensive learning of influence information is also a challenge in the research on time series prediction.

C. Optimization Algorithm

Recently, the AMSGrad [17] optimization algorithm is proposed. This method overcomes the invalid convergence of Adam [18] to the optimal solution under certain circumstances, and an internal max function is used to process the second-order moment estimation information of history and current time. However, the temporal dependence in the error gradient flow is ignored. Meanwhile, the AdaBound [19] algorithm is not sensitive to hyperparameters. Due to the random clipping of gradient information in the AdaBound, temporal dependence information may be lost. In the above existing studies, some algorithms ignore the learning of temporal dependence information, and other algorithms indirectly gain temporal correlation information indirectly at the cost of the increased difficulty of tuning hyperparameters and computational overhead.

This article proposes a novel viewpoint, that is, the accurate and efficient predictive performance resulting from the cooperation of three perspectives (i.e., training goals, model, and optimization) can be used to significantly overcome

the performance generated by the single perspective method (namely, Viewpoint 1).

To the best of our knowledge, the temporal change information learning (CIL) method is a new solution for multivariate time series prediction tasks. The framework of the proposed method is shown in Fig. 2. The main processing steps of the proposed method are briefly introduced as follows. First, the input multivariate time series data are processed by the transformation mechanism of the prediction model so that the temporal change information is mapped to a larger nonsupersaturated interval. Second, the impact information of different types of temporal change information between the time series on the target series is extracted through the processing of the objective function. Third, the optimal model parameters are obtained by the adaptive forget operation of the moment estimation information by the optimization algorithm. As the main innovation of this article, our proposed method improves the prediction performance through the cooperation of three perspectives, instead of a single perspective in previous studies. The main contributions are as follows.

1) A new objective function for multivariate time series prediction task is formally described, including the hybrid loss term (i.e., mean absolute error (MAE) and MSE loss functions) and temporal correlation term. The correlation term mainly includes operations of second-order difference. Through the above operations, the correlation term can guide the model to highlight the change amplitude information of each time series.
2) An LSTM network with two transformation mechanisms is designed (i.e., T-LSTM), which can fully capture the abrupt change and slow change information. The transformation mechanism maps the input data to a more variable range space (avoiding the supersaturated interval generated by the sigmoid function). Therefore, similar change rules can be effectively predicted in the future.
3) A new stochastic optimization algorithm (called adaptive moment estimation forgetting gradient (AdaMFG) algorithm) for time series prediction tasks is designed. It can adaptively control the proportion of forgetting the historical moment estimation information and the proportion of memory current moment estimation information, without introducing new hyperparameters. During the optimization phase, the abrupt change and slow change information can be indirectly learned through the operation of the temporal information before and after the error gradient flow.
4) Experimental results show that the proposed CIL method is significantly superior to the baseline methods in performance, and its computational overhead is much lower than that of the complex attention model. Moreover, the effect of setting hyperparameters in the objective function on prediction accuracy is explored. Experiments have confirmed that learning abrupt and slow change information in historical data can effectively improve the prediction accuracy in multivariate time series tasks with different sizes.

The rest of this article is organized as follows. In Section II, we start with an overview of existing solutions from three aspects: objective function, DL model, and first-order adaptive stochastic optimization algorithm. In Section III, the formal definition of abrupt change and slow change information is given, and the PM2.5 Air Quality dataset is used to analyze the temporal change information. In Section IV, the CIL method is comprehensively described. In Section V, the running efficiency and effectiveness of the prediction accuracy of the CIL method on three different scale real-world datasets are comprehensively analyzed and discussed. Finally, the conclusion and future study is concluded in Section VI.

II. Related Works

A. Objective Function

The objective function term of time series prediction (i.e., traffic flow prediction [20] and air quality prediction [4]) usually includes a popular loss function in regression tasks, such as MAE and MSE loss functions. Recently, alternative solutions for the Euclidean loss (i.e., MAE and MSE) have been proposed, in which the smooth approximation of the DTW [21] is widely used to guide the model to complete time series prediction tasks. Although model parameters of empirical risk minimization through backpropagation calculation can be obtained by standard DTW, the long-term temporal dependence cannot be effectively learned by emphasizing shape errors of two time series [22], [23]. Unlike the nondifferentiable DTW loss function described above, an idealized and differentiated timing error function is proposed to guide the model to learn the temporal dependence [16]. However, this method is only applicable to the prediction of binary and low-dimensional time series and unable to process the time series data of high-dimensional and multivariate. Recently, the mixing period and shape loss functions are proposed to solve univariate multistep time series prediction tasks [24]. Meanwhile, the important impact of abrupt change information on prediction accuracy is also investigated in this study. Nevertheless, in the above studies, abrupt change information and slow change information in the time series data are not extracted in the objective function. Compared with univariate, low-dimensional, and simple time series data, multivariate time series data have richer and more complex temporal change information. Therefore, it is essential to learn the change information in the objective function.

B. Model

In early studies, classic RNNs (i.e., LSTM [9] and GRU [10]) were used to solve simple univariate time series predictions. Shortly after, the Zoneout technology was used in the LSTM neural network [25] to inactivate some neuron connections in the model so that the long-term dependencies in time series data were effectively extracted. Thereafter, the IndRNN [26] network was proposed to complete the convergence in extremely long sequences tasks. However, the information is controlled by departments within the LSTM network that can be randomly lost in Zoneout, resulting in the disturbance for the temporal dependence information. The nonshared parameter operation of the IndRNN model can break the temporal change rule inside the series.

Meanwhile, encoder–decoder with temporal attention and input features attention mechanisms in a DA-RNN [11] model

can select the most relevant series from exogenous time series. Subsequently, global and local attention mechanisms were used to develop a GeoMAN [12] model, in order to enhance the selection of exogenous time series. Last year, EA-LSTM [27] was presented to accurately predict multivariate time series. In this model, an evolutionary algorithm is first introduced to filter better attention scores, and then, the LSTM layer using the attention mechanism can better learn the impact of different exogenous time series on the target series. However, the above three SOTA models have the following two limitations. First, based on the attention mechanism, a large number of hyperparameters (such as the number of recurrent network layers, the number of hidden state units, and evolutionary search space size) need to be searched and optimized; the time for searching the optimal hyperparameters and the computational complexity generated by multiple components lead to high computational overhead. Second, the most relevant input feature can be screened in such models; if a time series with mutation information is not selected, potential mutation events cannot be fully captured. Because of these limitations, the versatility of deep prediction models with strong generalization capabilities is severely compromised. When multivariate and high-dimensional large-scale time series prediction tasks are processed in prediction models running on devices with limited graphics processing unit (GPU) resources (i.e., the Internet of Things (IoT) devices), less tuning hyperparameters and simpler optimization processes are of great significance for the efficiency improvement of prediction tasks.

C. Optimization Algorithm

Adam algorithm [18] is widely used in various machine learning tasks, while the local minimum may be converged by Adam in some sequence tasks (such as [28]). In the AMSGrad [17] algorithm, the max function is used to solve such convergence problems, while the temporal dependence information in the error gradient flow may be damaged by the max function. Recently, due to the random cropping of gradient information of the AdaBound [19], this algorithm is no longer sensitive to hyperparameters and converges as fast as the Adam algorithm in machine learning tasks. However, the random cropping strategy still leads to the loss of temporal dependence information. Thereafter, the AdaHMG [29] algorithm involving temporal dependence information has slightly better performance in several simple time series prediction tasks. However, multiple hyperparameters are introduced into this strategy, leading to the complex tuning of hyperparameters. In the Corga [30] algorithm, the correlation information between the data history and the current moment can be learned in the context of concept drift. Regrettably, this algorithm is only verified in the vector autoregression (VAR) model in the field of econometrics and cannot be used as an optimization algorithm of RNNs. Thus, the Corga algorithm is not suitable for large-scale multivariate time series prediction tasks. Lately, a new stochastic optimization algorithm (i.e., AdaShift) [31] improves the convergence effect by introducing a decorrelation and temporal-shift mechanism. However, the main premise of this algorithm (the probability distribution between the data is independent and identical) is different from that of continuous time series data with temporal dependence information. Obviously, in these optimization algorithms, the temporal dependence information in the gradient error flow is not considered, and too many extra hyperparameters are introduced.

III. Problem Formulation

First, the prediction process of multivariate time series data is described formally in this section. Second, temporal change information is divided into abrupt change and slow change information according to the variation amplitude between the time series data. Moreover, the two kinds of change information are formally defined, referring to the research on change-point detection in the field of detection and signal processing. In the last (i.e., Supplementary Material), the change information of the PM2.5 Air Quality dataset is analyzed through the visualization method.

A. Prediction Process

Multivariate time series data consist of multiple nonpredictive time series and a target series. The time series data of n columns are defined as follows: $\mathbf{X} = (\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n -)^\top = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{T-1}) \in \Re^{n\times(T-1)}$, where $T-1$ represents the time window size of historical observations. Specifically, each nonpredictive time series is defined as: $\mathbf{x}^{\mathrm{i}} = (x_1^i, x_2^i, \ldots, x_{T-1}^i)^\top \in \Re^{T-1}$. Finally, $f(\cdot)$ (i.e., CIL method) is used for high-dimensional nonlinear mapping to learn the predicted value (i.e., $\hat{y}_T$) of the model at time T and y_i represents the value of the target series at the ith moment as follows:

$$\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{T-1}; y_1, \ldots, y_{T-1}\} \underset{f(\cdot)}{\longrightarrow} \{\hat{\mathbf{y}}_T\}. \tag{1}$$

B. Definition of Abrupt and Slow Change Information

We first review the classic description of abrupt change information in the change-point detection problem and then further define the temporal change information in the multivariate time series data.

In the classical change-point detection reference CUSUM [32], given an observation set $\{\mathbf{x}_t\}$, a score L_t is set for each observation, whose mean value is negative under the statistical distribution of H_0 and positive under the statistical distribution of H_1. When a condition is satisfied (i.e., $T_s = \inf\{\max_{1\le i\le t} \sum_{k=i}^{t} L_k \ge b, t > 0\}$), there (T_S) is an abrupt change point (where b represents the threshold obtained from streaming data). In CUSUM, this score is usually calculated using a log-likelihood ratio (LLR). The score is expressed as $L_t := \log(f_1(\mathbf{x}_t)/f_0(\mathbf{x}_t))$, and $f_1(\cdot)$ and $f_0(\cdot)$ are the probability density functions of $\mathbf{x}_t$ under H_0 and H_1, respectively.

Furthermore, we give the definition of the temporal change information of multivariate time series data according to the CUSUM. A time series signal $x_t \in \Re^D$ at time t is observed (D represents the number of dimensions), and it is dependent on an unknown low dimensional subspace. Specifically, this signal is defined as follows:

$$x_t = \boldsymbol{o}_t s_t + \boldsymbol{n}_t \tag{2}$$

where $\boldsymbol{o}_t \in \Re^{D\times r}$ represents an orthonormal basis for a subspace with r dimensions, $\boldsymbol{s}_t \in \Re^r$ represents time series signals in a low-dimensional space, and $\boldsymbol{n}_t \in \Re^D$ represents possible white noise terms (i.e., missing observations and low). We assume that s_t and $\boldsymbol{n}_t$ follow Gaussian distributions in independent, unknown subspaces. Subsequently, r and σ_n^2 (i.e., the variance of the noise term) are taken as prior knowledge, as the basic consensus in previous studies [33]. Moreover, the abrupt change points in the potential subspace are located in the unknown time node T_C, which can be expressed with a piecewise function as follows:

$$\boldsymbol{o}_t = \begin{cases} \boldsymbol{o}_0, & \text{if } t < T_C \\ \boldsymbol{o}_1, & \text{otherwise} \end{cases} \tag{3}$$

where $\boldsymbol{o}_0$ and $\boldsymbol{o}_1$ represents the orthonormal basis of the subspace prechange and the orthonormal basis of the subspace postchange, respectively. The abrupt change points by observing x_t, which is effectively equated to a hypothesis testing problem as follows:

$$H_0 : x_t = \boldsymbol{o}_0 \boldsymbol{s}_t + \boldsymbol{n}_t \tag{4}$$

$$H_1 : x_t = \boldsymbol{o}_1 \boldsymbol{s}_t + \boldsymbol{n}_t. \tag{5}$$

It is assumed that $T_C > 0$ indicates that some change occurs at time 0. Therefore, at the time node T_s, the score is calculated to find an abrupt change point. On the contrary, when the maximum value of time node T_s is less than b, it indicates that the time node has a slow change point. Obviously, the abrupt change information of a certain nonpredictive time series is formed by several consecutive abrupt change points at different time periods. On the contrary, some continuous slow change points constitute the slow change information.

According to the above definition, the impact of temporal change information on the target series is intuitively learned. However, there are two challenges in this direct approach. On the one hand, a huge amount of computational overhead is required for finding all the abrupt change points before learning impact information. The latest research on the detection of change points [33] reported that the computational overhead of the two main stages is $\mathrm{O}(D^3)$ and $\mathrm{O}(D(D-r))$, where D represents the number of dimensions. If there are hundreds of thousands of or even millions of multivariate time series data, the single computational overhead of detecting change points is as high as $\mathrm{O}(ND^3)$, where N represents the number of samples. On the other hand, this direct approach cannot capture the impact information of multiple consecutive abrupt change points on the target series while discovering the change points. Because different nonpredictive time series have different impact information on the target series within the same time period, the detection algorithm can only obtain the change points of one nonpredictive time series at a time period. Besides, the complexity and difficulty in obtaining information about the impact of abrupt change points on the target series are highly increased in this method. The difference technique integrated into the objective function is proposed to learn the impact of change points in nonpredictive time series on the target series (details are available in Section IV-A).

IV. CIL Method

In this section, we introduce how the CIL method accomplishes the multivariate time series prediction task from the three perspectives (i.e., objective function, neural network model, and optimization algorithm). First, the objective function guides the model to highlight and capture the impact of temporal change information in multivariate time series data on the target series. Subsequently, the LSTM neural network with a transformation mechanism can learn the abrupt and slow change information in the historical data. Finally, an optimization algorithm is presented for the indirect learning of history temporal change information in data. Also, the algorithm adaptively controls the fusion proportion of history and current moment estimation information.

A. Objective Function

To our knowledge, components of objective function are divided into the loss function and temporal correlation items (i.e., $\min_{\theta_{1,t},\ldots,\theta_{K,t}} \sum_{k=1}^{K}\sum_{t=1}^{T} \mathcal{L}(x_{k,t}(\theta_{k,t}), u_{k,t}) + \lambda\Omega(u_{1,t},\ldots,u_{K,t})$, where $\mathcal{L}(\cdot,\cdot)$ represents loss function, $\lambda\Omega(\cdot)$ represents temporal correlation items, and x_t and u_t represent the predicted and true values of the time series at time t, respectively. θ_t represents all the learnable parameters in the T-LSTM network (i.e., W, U, and b). To evaluate the effectiveness of the model during the training, a new objective function on multivariate time series prediction tasks is formally described as follows:

$$\begin{aligned} \min_{\theta_{1,t}\ldots,\theta_{K,t}} & C\frac{1}{T}\sum_{k=1}^{K}\sum_{t=1}^{T}\left(x_{k,t}(\theta_{k,t}) - u_{k,t}\right)^2 \\ & + (1-C)\frac{1}{T}\sum_{k=1}^{K}\sum_{t=1}^{T}\left|x_{k,t}(\theta_{k,t}) - u_{k,t}\right| + \lambda\sum_{k=1}^{K}\sum_{t=1}^{T}\left(\nabla_t^2 u_{k,t}\right)^2 \\ & + \lambda\sum_{m=1}^{K}\sum_{n=1,n\neq m}^{K}\sum_{t=2}^{T-1}\left(\nabla_t^2\left|u_{m,t} - \mathrm{corr}\left(u_{m,t},u_{n,t}\right)u_{n,t}\right|\right)^2 \end{aligned} \tag{6}$$

where $x_{k,t}$ is defined as the predictive value of kth series at time t, $u_{k,t}$ is defined as the true value of the kth series at time t (when $k = K$, it represents the true value of the target series at time t), C is the fusion proportionality factor of squared error and absolute error, ∇_t^2 is the second-order difference operation, $|\cdot|$ represents the absolute value operation, and hyperparameter λ is also known as the smoothing parameter (ensuring that the objective function is continuously differentiable). The value of λ is empirically adjusted according to the characteristics of multivariate time series data (see detailed information in Section V). The grid search is used to find the optimal hyperparameters. If other complex strategies are used to search for hyperparameters, the design purpose (the simpler solution and much lower computational overhead than that of the complex DL model) of CIL method is violated.

As far as we know, the MSE is the expected value of the square of the difference between the true value and the estimated value. Meanwhile, the MAE can better reflect the actual situation of the predicted error. We use a fusion constant C to fuse the two parts of errors to get a hybrid loss. Therefore, the hybrid loss considers not only the amplitude error between the predicted value and the real value but also the temporal change

information with the time axis. In addition, the third term of (6) describes the abrupt change information and slow change information within each nonpredictive series within a certain period. We give a generalized description of the amplitude of temporal change information (including abrupt and slow change information) of each time series on multivariate time series data of different sizes as follows:

$$\lambda \nabla_t^2 u_{k,t} = \begin{cases} \lambda \left| u_{k,t-1} - 2u_{k,t} + u_{k,t+1} \right|, & 2 \le t \le T-1 \\ 0, & t = 1 \text{ or } t = T. \end{cases} \tag{7}$$

According to most studies of time series analysis, the continuous range of time lag, temporal dependence, and temporal change information in the data is generally the most obvious within several time intervals [34], [35]. Thus, we need to focus on the magnitude of the temporal change information between the data in short time intervals (especially three-time intervals). In discrete mathematics, first-order differences measure the difference between two consecutive data (i.e., $\nabla_t^1 u_{k,t} = u_{k,t+1} - u_{k,t}$). We choose the absolute value method of the second-order difference to deal with the temporal change information for three reasons.

The first reason for using second-order differential operations to capture temporal change information comes from visual analysis techniques for temporal change information. Fig. 3 shows nine different types of temporal change information at three consecutive time points in the form of a line chart. In Fig. 3, t represents the continuous time (i.e., t_1, t_2, and t_3), v represents values (i.e., v_i, v_j, and v_k) corresponding to different time points, and x represents continuous data points (i.e., x_1, x_2, and x_3) of a time series in multivariate time series data.

First, Fig. 3(f) is the most special one, and the values of the three consecutive data points in this sequence have not been changed. This shows that the amplitude of this temporal change is 0, and it does not need to be focused on by the second-order difference technique. Second, there is the temporal change information, including a combination of rising or fall and no change.

As shown in Fig. 3(a), (d), (e), and (g), specifically, a case analysis is provided for Fig. 3(a). The values corresponding to the three time points are assumed to be 3, 6, and 6. After the calculation of (7), $\nabla_2^2 u_{k,2} = 3$ (i.e., the temporal change amplitude information at $t = 2$) is obtained. This calculation result suggests that there may be significant changes (i.e., abrupt change information) during this time period. Third, we can find that the temporal change information of Fig. 3(b) and (h) monotonically increases or decreases. A case analysis is provided for Fig. 3(b). The values corresponding to the three time points are assumed to be 5, 6, and 8. After the calculation of (7), $\nabla_2^2 u_{k,2} = 1$ is obtained. This calculation result reflects that there may be slow changes (i.e., slow change information) during this time period. As for Fig. 3(c) and (i), there is a similar temporal change type of the first rise and then fall or the first fall and then rise. The temporal change amplitude information can be easily extracted. The second reason comes from this study [36]. The results of this study show that second-order differences have similar competitive filtering effects to higher order differences in the image reconstruction field. For continuous time series

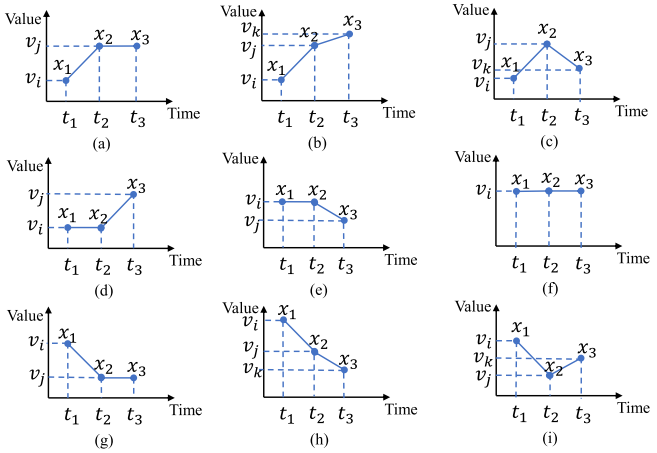


Fig. 3. Different types of temporal change information. (a)–(i) Cases 1–9, respectively.

signals, a second-order difference can also be viewed as a simpler and more efficient method. This article mainly focuses on choosing a suitable method to extract the temporal change amplitude information, and only a brief formal analysis on the extraction of temporal change amplitude information is given here. More details are available in this previous study [36]. The absolute value operation is removed in (7) as follows:

$$\lambda \left\| \text{Diif}^{(m+1)} \right\|_1 \tag{8}$$

$$\text{Diif}^1 = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ & & \ddots & \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

$$\text{Diif}^2 = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ & & & \ddots & & \\ 0 & 0 & 0 & 1 & -2 & 1 \end{bmatrix} \tag{9}$$

where $m + 1$ represents the order of the differences (i.e., $m = 1$). The expanded form of $\text{Diif}^2 = \text{Diif}^1 \cdot \text{Diif}^1$ in (9) is obtained. Based on previous research [37], if $m = 0$, (8) automatically degenerates into a classic total variation (TV) filter. When the sequence trend signal is approximately piecewise constant, the TV filter is suitable. Trend signals are usually divided into uptrend, downtrend, and unchanged trend. However, it is found that different types of temporal change information are randomly composed of the above three types of trends signals. Obviously, the TV filter (only filtering the above three types of trend signals) and first-order differences filter cannot effectively extract different types of complex temporal change information in multivariate time series data. Therefore, the proposed second-order difference with absolute value operation is a valuable and novel solution. It can effectively guide the model to highlight the influence of these amplitudes of change information on the target series.

The third reason is that the introduced calculation operations require near-linear computational overhead (this meets the requirements of computational overhead reduction in our design ideas). Specifically, the number of difference calculations for each attribute column is $N - 1$, and D times are needed for the multivariate time series data of D dimension

to process each time series. Therefore, the total computational overhead is only $O((N-1)D)$. Difference operation results in the time complexity of the learning system within an acceptable range.

Therefore, second-order differences with absolute value operation are more suitable for guiding the model to highlight the influence of this amplitude of change information on the target series. It is worth noting that because the local and the global abrupt change amplitude information are different for different time series datasets. In order to simplify the extraction of temporal change information, we use the absolute value of the second-order difference to adaptively extract the temporal change amplitude information of each sequence instead of setting the mutation threshold (i.e., the threshold b mentioned in Section III) to determine when this information exists. The third and fourth terms in (6) can be used as temporal correlation terms to prevent the overfitting phenomenon.

In the fourth term, $\text{corr}(u_{m,t}, u_{n,t}) = \text{cov}(u_{m,t}, u_{n,t})/(\sigma_{u_{m,t}} \cdot \sigma_{n,t})$ is calculated with paired nonpredictive time series information, which is used to represent the correlation between paired nonpredictive time series. Note the fact that, learning the nonlinear temporal change information mainly depends on the neural network model. This study [35] has proved that the covariance calculation module can guide the model to filter temporal change trends information in time series data. Inspired by this study [38], the correlation coefficients with inner covariance calculation contribute to capturing linear correlation information between pairs of nonpredictive series. The correlation coefficient can be seen as a deformation of the covariance (i.e., a special covariance after standardization that eliminates the influence of the dimensions of the two variables.). Since it is a special covariance, it has the following characteristics.

1) It can also reflect whether the two variables change in the same direction or the reverse direction. If they change in the same direction, it is positive, and the reverse change is negative.
2) Since it is the standardized covariance, the more important feature comes; it eliminates the influence of the change of the two variables and only reflects the similarity of the two variables per unit change.

The above content shows that the use of correlation coefficients can better characterize the temporal correlation information between time variables.

In summary, the temporal information of multivariate time series data can be comprehensively learned by the objective function through the hybrid loss function (composed of MAE and MSE losses), the second-order differences absolute value operation (for extracting temporal change information), and the extraction term of correlation information between sequences.

B. T-LSTM Neural Network

We propose a new variant of the LSTM network with a transformation mechanism (i.e., T-LSTM), which can capture abrupt and slow change information within the input time series data stream. T-LSTM is introduced to guarantee that input and forget gate output are updated with new function transformation operations without randomly losing memory cell information or hidden state information. The hyperbolic tangent function is used to adjust the output value range for forget and input gating, so as to capture the abrupt change information. The essential difference between the T-LSTM and the standard LSTM is the function mapping operation for the output value of forget and input gating [i.e., (14) and (15)] as follows:

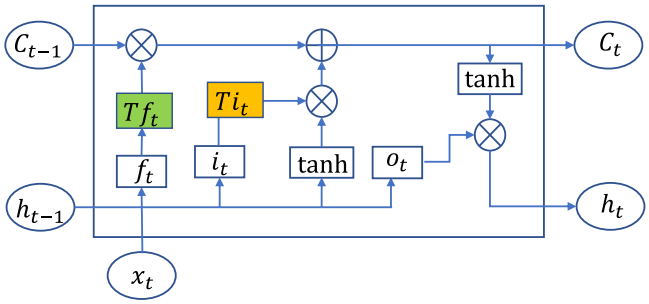


Fig. 4. Structure of T-LSTM neural network.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{10}$$
$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{11}$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{12}$$
$$g_t = \tanh(W_g x_t + U_g h_{t-1} + b_g) \tag{13}$$
$$Tf_t = 1 - t\,\text{anh}(f_t) \tag{14}$$
$$Ti_t = 1 - t\,\text{anh}(i_t) \tag{15}$$
$$c_t = Tf_t \otimes c_{t-1} \oplus Ti_t \otimes g_t \tag{16}$$
$$h_t = o_t \otimes \tanh(c_t) \tag{17}$$

where i_t, f_t, and o_t represent the input gate, forget gate, and output gate at time t, respectively. Moreover, σ represents the sigmoid activation function. $\oplus$ and $\otimes$ represent the matrix splicing and vector multiplication, respectively. Tf_t and Ti_t in the green and yellow rectangular boxes represent the output processing for the forget and input gates by the transformation mechanism at time t, respectively. In addition, c_t and h_t represent memory cell and hidden states at time t, respectively. Fig. 4 shows the processing of a T-LSTM cell at time. Within a standard LSTM, the input data go through the forget gate. If the output value meets the rule that values close to 0, input data are completely discarded; if the values close to 1, input data are completely retained. Obviously, forget gate plays a crucial role in the memory and forgetting of time series data flow at the moment before and after. The output value range of the forget gate is [0, 1], which is mapped to the interval of [0.25, 1.0] by our transformation mechanism. Corresponding to the time series data, the domain that is input into the neural network is between [0, 1]. After the transformation processing of $1-\tanh(\cdot)$ function, the range of values is obtained through the transformation gating function, corresponding to the data (i.e., [0.25, 1.0]). Through this mechanism, the original value close to 1 will decrease to 0.25, while the original output value close to 0 will become close to 1. The original values located near the middle are compressed centrally to 0.5. After mapping, the range of values of data is compressed to the interval with the most obvious changes, which is more conducive to capturing temporal change information among data, especially abrupt change information in the model. Similarly, the input gate output range is processed through a transformation mechanism. This mechanism avoids the supersaturation interval of

the sigmoid activation function to some extent. Compared with the supersaturated interval of sigmoid, the abrupt change and slow change information falling in the unsaturated interval can be effectively learned. In other words, important change information is reduced to fall into the supersaturation interval. The calculation process (i.e., backpropagation) of the partial derivative of this mechanism is presented in the Supplementary Material.

Algorithm 1 AdaMFG—the Proposed Algorithm for the First-Order Stochastic Optimization

Require: α : Learing Rate, α_t is the learning rate that varies over time by error correction
Require: $\beta_1, \beta_2 \in$[0,1) and $\epsilon = 1 * 10^{-8}$: Exponential decay rates for the moment estimates
Require: $ff_t \in [0, 1)$ (initialize $ff_0 = 0.5$, if $t \geq 1$, $f\ f_t = ff_0 \times m_t$): Forgetting factor
Require: $f(\theta)$: Stochastic objective function with all learnable parameters in the T-LSTM model θ
Require: θ_0., Initial vector: $m_0 = 0, \upsilon_0 = 0, t = 0$
1: While θ_t not converged do:
2 : $t = t + 1$
3 : $g_t = \nabla f_t(\theta_t)$
4 : $\alpha_t = \alpha/\sqrt{t}$
5 : $m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$
6 : $ff_t = ff_0 \times m_t$
7 : $\hat{m}_t = ff_t \widehat{m}^2_{t-1} + (1 - ff_t) m_t$
8: $\upsilon_t = \beta_2 \upsilon_{t-1} + (1 - \beta_2) g_t^2$
9 : $\hat{\upsilon}_t = ff_t \hat{\upsilon}_{t-1} + (1 - ff_t)\upsilon_t$
10 : $\theta_t = \theta_{t-1} - \alpha_t \cdot \hat{m}_t / \left(\sqrt{\hat{\upsilon}_t} + \varepsilon\right)$

C. AdaMFG Algorithm

To specifically process the abrupt and slow change information between time series data in a first-order adaptive stochastic optimization algorithm, a simple and effective AdaMFG algorithm is proposed. The core idea of the AdaMFG algorithm is to indirectly learn the changing rules of the complex temporal dependence information inside the data through the fusion operation of historical and current moment estimation information. Due to the rich temporal change information in the multivariate time series data, it is necessary to focus on the variation rules of moment estimation information before and after the gradient error iteration. Therefore, we learn the abrupt and slow change information between the error gradient flow information (corresponding to lines 7 and 9 of the pseudocode of Algorithm 1) by directly integrating the historical and current information of the first- and second-order moment estimations. The most critical point is that, considering that four hyperparameters are introduced in the fusion operation of first- and second-order moment estimations, the product of first-order moment estimation m_t at the current time and the ratio of initial memory factor are used to derive from the adaptive control of the fusion ratio of current and historical moment estimation information. The quadratic term of $\hat{m}^2_{t-1}$ is selected for lower time complexity (see the Supplementary Material for details). In this way, the memory ratio of the moment estimation information is realized based on the temporal information of real-time error gradient flow, and no additional hyperparameters are introduced. The ratio of memory and forgetting of current and historical information should be consistent with the first- and second-order moment estimations, which meets the requirement of signal-to-noise ratio (SNR) in the data sampling principle.

The upper bound of the computational complexity of the proposed algorithm should be determined, and the formula of first and second moment estimation fusion calculations needs to be processed by a scaling method. First, the formula derivation of information fusion operation of the second-order moment estimation is presented, corresponding to the convergence property of the algorithm

$$\begin{aligned}\hat{\upsilon}_t &= ff_t \hat{\upsilon}_{t-1} + (1 - ff_t)\upsilon_t \\ &\geq ff_t \upsilon_{t-1} + (1 - ff_t)\upsilon_t \\ &\geq ff_t \upsilon_{t-1} + (1 - ff_t)\upsilon_{t-1} = \upsilon_{t-1}. \end{aligned} \tag{18}$$

In (18), the first inequality is true if $\hat{\upsilon}_t \geq \upsilon_t$, and the derivation of the second inequality is true as follows:

$$\begin{aligned}\upsilon_t &= (1 - \beta_2) \sum_{j=1}^{T-1} \beta_2^{T-1-j} g_j^2 + (1 - \beta_2)\beta_2^0 g_T^2 \\ &= \upsilon_{t-1} + (1 - \beta_2)\beta_2^0 g_T^2 \left(g_T^2 \geq 0\right) \end{aligned} \tag{19}$$

since $g_T^2 \geq 0, \upsilon_t \geq \upsilon_{t-1}$ in (19) is true. According to the previous research, the algorithm can converge when $\hat{\upsilon}_t$ increases monotonously.

Similar to the derivation process of the second-order moment estimation fusion operation in this article, we provide the derivation process of the first-order moment estimation information fusion operation related to the scaling method

$$\begin{aligned}\hat{m}_t &= ff_t \hat{m}^2_{t-1} + (1 - ff_t) m_t \\ &\geq ff_t m^2_{t-1} + (1 - ff_t) m_t \\ &\geq ff_t m^2_{t-1} + (1 - ff_t) m_{t-1}. \end{aligned} \tag{20}$$

In (20), the first inequality is valid when $\hat{m}_t \geq m_t$, and the derivation of the second inequality is as follows (because of $g_T \geq 0$, thus, $m_t \geq m_{t-1}$ is true in (20)]:

$$\begin{aligned}m_t &= (1 - \beta_1) \sum_{i=1}^{T-1} \beta_1^{T-1-j} g_i + (1 - \beta_1)\beta_1^0 g_T \\ &= m_{t-1} + (1 - \beta_1)\beta_1^0 g_T (g_T \geq 0). \end{aligned} \tag{21}$$

The online learning framework [39] proposed by previous studies is used to analyze the convergence of the AdaMFG algorithm. Regret is used to evaluate the convergence performance of the first-order adaptive stochastic algorithm on a randomly selected sequence of convex objective functions.

$f_1(\theta), f_2(\theta), \ldots, f_T(\theta)$ is defined as

$$R(T) = \sum_{t=1}^{T} \left(f_t(\theta_t) - f_t\left(\theta^*\right)\right) \tag{22}$$

where $\theta^* = \text{argmin}_{\theta \in \chi} \sum_{t=1}^{T} f_t(\theta)$. The goal is to calculate the summation of the difference between the value of the current time step $f_t(\theta_t)$ and the optimal solution at T time. The regret of the AdaMFG algorithm is less than $O((dT)^{1/2})$,

and the proof of Theorem 4.1 is attached in the Supplementary Material.

Theorem 4.1: Assuming that the proposed objective function has a bounded gradient, then the issue of multivariate time series smoothing has been reduced to solving a convex optimization problem with a temporal correlation factor λ as a parameter. First, θ is generalized from the two-norm of the gradient: $\|\nabla f_t(\theta)\|_2 < G$ to the infinite norm: $\|\nabla f_t(\theta)\|_\infty < G_\infty$ Suppose that χ has a bounded diameter D_∞, then for all $\theta \in \chi$, there is $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\gamma = (\beta_1)^2/(\beta_2)^{1/2} < 1$. The AdaMFG algorithm supports for the first-order moment estimation by adding a momentum method (i.e., $\beta_{1t} = \beta_1 \lambda^{t-1}$ and $\alpha_t = (\alpha/(t)^{1/2})$ $t \in [1, \ldots, T]$). Finally, the following regret ($R(T)$) is provided as:

$$R(T) \leq \frac{D_\infty^2 \sqrt{T}}{\alpha(1-\beta_1)} \sum_{i=1}^{d} \hat{v}_{T,i}^{\frac{1}{2}} + \frac{\beta_1 D_\infty^2 G_\infty}{2(1-\beta_1)(1-\lambda)^2} + \frac{4\alpha(ff_0)^2\sqrt{1+\log T}}{(1-\beta_1)^3\sqrt{(1-\beta_2)(1-\gamma)}} \sum_{i=1}^{d} \left\| g_{1:T-1,i} \right\|_2^{\frac{3}{2}}. \tag{23}$$

In Theorem 4.1, when the condition of bounded gradients is satisfied, compared with previous studies $\sum_{i=1}^{d} \hat{v}_{T,i}^{(1/2)} < (d)^{1/2}$ and $\sum_{i=1}^{d} \|g_{1:T-1,i}\|_2^{(3/2)} < \sum_{i=1}^{d} \|g_{1:T,i}\|_2 \ll d(T)^{1/2} G_\infty$. Since the value of the actual loss function is less than 1, it is obvious that the gradient information obtained by iterative calculation at each time is also less than 1 (i.e., $g_{1:T-1,i} < 1$), thus, $\sum_{i=1}^{d} \|g_{1:T-1,i}\|_2^{(3/2)} < \sum_{i=1}^{d} \|g_{1:T,i}\|_2$. The result of right-hand side of inequality (23) is slightly less than the boundary of the previous studies (such as Adam and AMSGrad), indicating the faster convergence of the AdaMFG algorithm. Specifically, the initial value of ff_0 is 0.5, while $0 < (ff_0)^2 < (ff_0) < 1$, so $4\alpha(ff_0)^2$ is less than 1. Therefore, the product of this result $\sum^{d} \|g_{1:T-1,i}\|_2^{(3/2)}$ will make $R(T)$ further smaller. In addition, $i = 1$, and the use of momentum decay in Theorem 4.1 (β_{1t}) still guarantees that the $R(T)$ of the AdaMFG satisfies $\widetilde{O}((T)^{1/2})$, for all $T \geqslant 1 : (R(T)/T) = O((1/(T)^{1/2}) + ((1+\log T)^{1/2}/T))$. This result can be obtained by using Theorem 4.1 and $\sum_{i=1}^{d} \|g_{1:T-1,i}\|_2^{(3/2)} < \sum_{i=1}^{d} \|g_{1:T,i}\|_2 \ll (dT)^{1/2}$. Thus, the loss function value can be reduced to 0 as T goes to infinity: $\lim_{T\to\infty}(R(T)/T) = 0$ (i.e., the loss function can reach the convergence state).

D. Training Procedure

Our CIL method is smooth and differentiable, so all parameters can be learned with this objective function. Equation (6) can be arranged into the following form:

$$\mathcal{J}(\theta) = \mathcal{L}(\theta) + \lambda\Omega_1(\theta) + \lambda\Omega_2(\theta) \tag{24}$$

where the first term is the loss function term and the latter two terms are the temporal correlation terms that are used to describe the abrupt and slow change information within the time series data. It is worth noting that the first term in (24) is still convex because the value of the second-order derivative of each term is $a'(\theta_{k,t}) = 2 > 0$ (i.e., greater than 0). Thus, the smooth and convex objective function can learn the best parameters.

TABLE I
DETAILS OF THREE DATASETS

Datasets	Records	Attributes	partition ratio
Weather	1,003,600	18	7:2:1
PM2.5	43,801	9	3:1:1
Energy	19,736	27	8:1:1

V. EXPERIMENTS

In this section, the efficiency and effectiveness of the CIL method are evaluated by three multivariate time series datasets of different sizes. Meanwhile, different contributions of the three perspectives are verified through ablation experiments. In addition, the important point (i.e., Viewpoint 1) of this article is also validated by variant experiments. Finally, the significance test is in the Supplementary Material.

A. Datasets

Three real-world multivariate time series datasets with abundant and complex temporal dependencies in different fields are selected to verify the effectiveness of the proposed CIL method.

1) *Weather Dataset:* As a large-scale dataset, this dataset includes 32-year detailed meteorological data [40] recorded by hour from the Irish National Weather Service website (target series is cloud amount). This dataset contains temperature, pressure, precipitation, wind direction, wind speed, humidity, light intensity, cloud size, and other comprehensive meteorological attributes.
2) *PM2.5 Air Quality Dataset [41]:* It is collected by UCI public dataset website, containing a series of meteorological data related to air quality (i.e., PM2.5 index is a target series. The main meteorological attributes include wind direction, wind speed, air temperature, air pressure, precipitation, rain, and snow conditions).
3) *Energy Consumption Dataset [42]:* This is a dataset collected from the energy consumption of electrical appliances in energy-saving buildings from the UCI website. The sensor equipment of home not only collects electricity usage data but also obtains temperature and humidity in the kitchen, bathroom, living room, parents and teen rooms, as well as outdoor weather data such as temperature, pressure, and humidity.

The time interval for the Weather, PM2.5 Air Quality, and Energy Consumption datasets is set to be 1 h, 1 h, and 10 min, respectively. The datasets for Weather, PM2.5 Air Quality, and Energy Consumption span from 1986 to 2018, 2010 to 2014, and April 30 to September 15, 2017, respectively. It is worth noting that the training set, validation set, and test set of the Weather dataset are divided using the classic partition ratio in machine learning research. The partition ratios of the other two datasets are consistent with previous studies. Table I shows the details of three real-world datasets.

B. Baseline Methods and Hyperparameter Selection

All the experimental programs in this article are implemented in the Windows10 operating system (CPU: Intel i7-7850HQ @ 4.0 GHz, 64-GB DDR4 of memory, GPU:

TABLE II
DETAILS OF THE MAIN PARAMETER SETTINGS FOR ALL BASELINE METHODS

Baselines	Parameter setting details
ARIMA	p (the lag of the time series data) = 1, d (the order of the difference) = 1, q (the lag of prediction error) = 0.
GBRT	n (the number of weak regressors) = 1,000, d (the depth of the decision tree) = 2, lr (initial learning rate) = 3e-3.
LSTM+mse-loss	The number of units in LSTM = 128.
IndRNN+mse-loss	Cyclic parameters clip min and clip max = -1, the number of units in IndRNN = 128.
LSTM+Zoneout+mse-loss	Inactivation ratio of memory cells and hidden state information = 0.15, the number of units in LSTM = 128.
IndRNN+mse-loss+AdaBound	Initial learning rate = 0.1, the hyperparameter Γ = 1e-03 in AdaBound, the number of units in IndRNN = 128.
IndRNN+mse-loss+AdaHMG	The hyperparameters k_1 = 0.5, k_2 = 0.3 in AdaHMG, the number of units in IndRNN = 128.
EA-LSTM	The number of time steps T_s = 18, the number of hidden state units m = 128, the size of search space set N = 36, the encoding length for each attention weights L = 6, the size of champion attention weights subset = 6.
DA-RNN	The time window=12, the number of layers=2, the number of hidden states of the encoder-decoder=64.
GeoMAN	The trade-off parameter λ = 0.2, the length of window size = 12, the hidden units of the encoder-decoder=64, the number of layers = 2.
CIL	The number of units in T-LSTM=128, the trade-off hyper-parameters λ and C in Figure 5 and Table III.

TABLE III
HYPERPARAMETER (C) SELECTION ON THE THREE DATASETS

C	PM2.5 RMSE	Energy RMSE	Weather RMSE
0.1	2.853	21.433	**28.455**
0.2	2.849	22.371	33.267
0.3	**2.682**	20.932	34.735
0.4	2.824	**20.152**	33.189
0.5	2.847	20.588	36.787

NVIDIA GeForce GTX 1080), and the running environment of all baseline model programs is tensorflow1.4.0-gpu [43] and keras2.1.5-gpu [44]. To evaluate the performance of CIL method in multivariate time series prediction tasks, some representative studies were selected as baseline methods [detailed introduction in the Supplementary Material (Section IV)]: 1) ARIMA [1]; 2) GBRT [2]; 3) LSTM + mse-loss [9]; 4) IndRNN + mse-loss [26]; 5) LSTM + Zoneout + mse-loss [25]; 6) IndRNN + mse-loss + AdaBound [19]; 7) IndRNN + mse-loss + AdaHMG [29]; 8) EA-LSTM [27]; 9) CIL; 10) DA-RNN [11]; 11) GeoMAN [12]; and 12) CIL[1]. Since the data preprocessing method of the DL model with the attention mechanism is different from that in the above baseline model, then the values of the performance metrics on the test set are small. In particular, this method of data preprocessing prevents the model from restoring the original dimension of the data during the test phase. Thus, the CIL[1] method and DA-RNN and GeoMAN model were separately compared for experimental analysis (in Table IV, superscript 1 is used to distinguish the CIL method). The parameter settings of all comparison models are shown in Table II. It is worth noting that the value of the initialized learning rate default of $1e^{-3}$ is used for neural network methods (the Adam [18] optimizer is used by default) that are not marked with the clear value of the initialized learning rate.

We explore the sensibility of the hyperparameters λ and C corresponding to the objective function and the prediction performance in the CIL method. We choose the classic grid search method to find the best λ value (i.e., the smallest test root-mean-squared error (RMSE) on different datasets). The range of λ was empirically set to [0.1, 0.2, 0.4, 0.8, 1.0]. In Table III, we can find that the fusion ratio of the two classical loss functions on different datasets has different effects on performance. Specifically, C that is equal 0.3 is best for PM2.5 tasks, 0.4 is best for energy tasks, and 0.1 is best for weather tasks. Fig. 5 shows that λ has the best test RMSE value when $\lambda = 0.2$ in three different-scale multivariate time series prediction tasks. When λ is greater than 0.2, the test RMSE increases to different degrees on different datasets with the increase of λ. It indicates that the greater λ value (i.e., $\lambda > 0.8$) can significantly reduce prediction performance. Meanwhile, the smaller λ value (i.e., $\lambda = 0.1$) and the greater λ value (i.e., $\lambda = 0.8$) have similar RMSE values that are far from optimal. It is further shown that when $\lambda = 0.1$, less information about abrupt and slow change information can be extracted by the objective function, probably leading to the decrease of prediction accuracy. Finally, the above experimental results verify that the temporal change information extracted by the temporal correlation terms in the objective function with the value of λ will have different prediction accuracies.

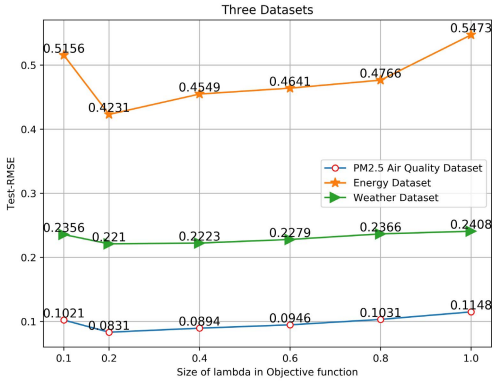


Fig. 5. Best λ value in objective function on three datasets.

C. Experimental Analysis

The CIL method and all baseline models in three different multivariate time series prediction tasks are compared and analyzed. To our knowledge, the RMSE and MAE are very popular metrics, which are used to evaluate the performance of time series prediction methods as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{25}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{26}$$

where y_i represents the real value of the ith test data, $\hat{y}_i$ represents the predicted value of the ith test data obtained by model inference, and n represents the total number of samples in the test set.

Table IV shows the median performance and its floating range of performance obtained by different baseline methods on three time series prediction datasets through five repeated experiments under optimal parameter settings. For PM2.5 air quality tasks, the Test-MAE of CIL[1] method improved by 44.12% over GeoMAN and its Test-RMSE improved by 51.00% over GeoMAN. On the Weather dataset, the test-MAE of CIL[1] method improved by 9.38% over GeoMAN, and the test-RMSE of CIL[1] method improved by 7.45% over GeoMAN. On the Energy dataset, the CIL[1] method improved by 33.58% over GeoMAN in the Test-MAE metric and 30.89% over GeoMAN in the Test-RMSE. It is clear that the CIL[1] method has overcome all baseline models with the smallest performance float for three different prediction tasks, indicating the sufficient stability and robustness of the CIL[1] method.

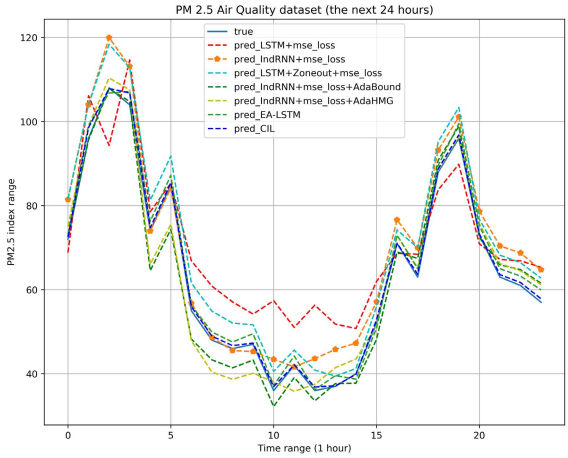
On the other hand, our proposed method also has advantages over the DL model based on evolutionary algorithms. On the PM2.5 Air Quality dataset, the Test-MAE of CIL method improved by 41.73% over EA-LSTM (i.e., the SOTA method) and its Test-RMSE improved by 28.03% over EA-LSTM. On the Weather dataset, the Test-MAE of CIL method improved by 19.53% over EA-LSTM and its Test-RMSE improved by 12.43% over EA-LSTM. On the Energy dataset, the CIL method has improved by about 38.06% and 21.30% in the Test-MAE and Test-RMSE metrics, respectively.

As shown in Table IV, ARIMA and GBRT methods have better performance than some neural network methods (i.e., LSTM and IndRNN with MSE loss) on datasets with a small number of records (i.e., PM2.5 and Energy datasets). On the Weather dataset with millions of records, the test performance of the ARIMA and GBRT methods is significantly weaker than the neural network methods. This shows that data-driven neural network methods have better generalization performance than traditional machine learning and statistical methods in large-scale time series prediction tasks. Meanwhile, with the increase of data size, the prediction accuracy is difficult to be achieved in traditional methods by learning nonlinear temporal dependencies in multivariate time series data. In Table IV, the EA-LSTM using an evolutionary search algorithm shows excellent accuracy on three different datasets, and its biggest advantage is that it uses an evolutionary search algorithm to find the model parameters that make the model's prediction performance optimal. Obviously, the biggest disadvantage comes from the huge search space and the search overhead is much larger than the expected one. The performance of the above DL methods further implies that it is difficult to further improve the prediction accuracy only by improving the structure of the neural network. The main reason is that in the above methods, an objective function and an optimization algorithm are not contained, and the best model parameters are difficult to obtain once the local minimum point is caught during the training process.

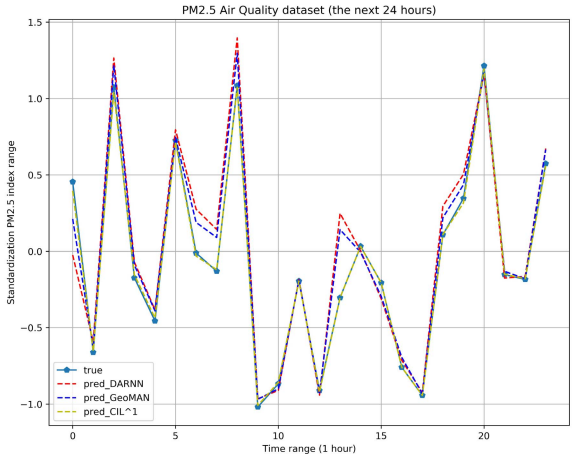
Due to space constraints, the prediction effects of PM2.5 air quality were mainly analyzed. The prediction effects of the proposed method and the SOTA methods on the other two prediction tasks were presented in the Supplementary Material. In Fig. 6(a), the blue dotted line corresponds to the predicted value generated by the CIL method. Except for a deviation of about 10 from the actual PM2.5 value for 2–3 h, the rest predicted values almost coincide with the real values. In particular, the CIL method can accurately predict the above information at the 5th, 10th, and 16th to 18th hours when there is an obvious mutation. This directly demonstrates that the CIL method can effectively learn the abrupt change and slow change information in the face of complex air quality time series prediction tasks. However, AdaHMG + IndRNN, the baseline model with the best performance on the test set, has an unsatisfactory prediction effect in the next 24 h. The abrupt change information cannot be significantly predicted by EA-LSTM in the small period of 5–15 h, and its predicted value is generally about 7–15 deviation from the real value. It implies that this method still does not solve well the problem of multivariate time series prediction. In Fig. 6(b), the CIL[1] method almost overlaps with the true curve, while other SOTA methods are still deviating from the phase of sharp ups and downs. In Fig. 7, the brown curve of CIL method basically

TABLE IV
COMPARISON OF THE PROPOSED METHOD AND ALL BASELINE METHODS ON THREE DATASETS

Methods	Test-MAE	Test-RMSE
PM2.5 Air Quality Dataset		
ARIMA	10.804±0.971	16.509±1.124
GBRT	15.308±1.218	19.333±1.983
LSTM+mse-loss	18.500±1.166	22.567±2.405
IndRNN+mse-loss	7.921±1.208	13.323±2.557
LSTM+Zoneout+mse-loss	8.269±1.354	12.544±2.187
IndRNN+mse-loss+AdaBound	6.011±0.978	9.416±1.463
IndRNN+mse-loss+AdaHMG	5.149±0.374	8.203±0.871
EA-LSTM	2.784±0.128	3.956±0.987
CIL	**1.561±0.092**	**2.682±0.351**
DA-RNN	0.1423±0.014	0.1795±0.023
GeoMAN	0.1310±0.012	0.1696±0.017
CIL[1]	**0.0728±0.010**	**0.0824±0.019**
Weather Dataset		
ARIMA	117.174±0.683	288.186±1.015
GBRT	122.666±0.712	267.199±1.343
LSTM+mse-loss	114.640±1.227	163.632±1.709
IndRNN+mse-loss	110.666±0.208	158.750±0.557
LSTM+Zoneout+mse-loss	113.100±1.181	162.931±1.372
IndRNN+mse-loss+AdaBound	103.716±1.478	157.200±1.563
IndRNN+mse-loss+AdaHMG	104.050±1.127	156.981±1.165
EA-LSTM	26.254±0.853	42.012±1.137
CIL	**13.230±0.125**	**28.455±0.193**
DA-RNN	0.0866±0.011	0.2458±0.013
GeoMAN	0.0810±0.012	0.2338±0.017
CIL[1]	**0.0721±0.002**	**0.2185±0.010**
Energy Dataset		
ARIMA	18.188±0.674	39.735±1.123
GBRT	25.606±0.227	36.192±0.845
LSTM+mse-loss	33.628±0.095	39.069±0.187
IndRNN+mse-loss	24.510±0.318	35.150±0.741
LSTM+Zoneout+mse-loss	33.511±0.129	39.004±0.372
IndRNN+mse-loss+AdaBound	21.552±0.096	32.399±0.163
IndRNN+mse-loss+AdaHMG	17.351±0.145	28.248±0.198
EA-LSTM	15.428±0.238	26.163±0.785
CIL	**10.022±0.075**	**20.152±0.087**
DA-RNN	0.2835±0.012	0.6473±0.015
GeoMAN	0.2608±0.017	0.6123±0.024
CIL[1]	**0.1702±0.002**	**0.3857±0.005**



(a)



(b)

Fig. 6. Comparison of 24-h air quality prediction curves of all models. (a) Comparison of true and predicted curves of the CIL methods and baseline models on the PM2.5 Air Quality dataset. (b) Comparison of true and predicted curves of the CIL[1] method and two SOTA models on the PM2.5 Air Quality dataset.

converges after 14 epochs, which is significantly lower than the validation MAE values of all baseline models. However, the purple curve is second to the curve of our proposed model after 30 epochs. Although the AdaHMG algorithm is more suitable for time series prediction tasks than the AdaBound algorithm, its validation MAE curve is still significantly higher than that of the CIL method. In addition, the LSTM + Zoneout model only uses ten epochs to reach the convergence. However, this also suggests the limitations of the Zoneout approach, which uses random loose information to accelerate convergence. Fig. 7 shows that the CIL method is better than all baseline methods in terms of convergence effectiveness.

In Fig. 8, the validation MAE curve of CIL[1] method is smoother than that of the DA-RNN. The CIL[1] method has a smaller initial validation MAE value than DA-RNN, suggesting that the hybrid loss of the objective function provides a lower initial loss value. The CIL[1] method is basically smooth and convergent after 40 epochs, while the DA-RNN model continues to oscillate slightly. GeoMAN's validation curve stopped decreasing after 28 epochs and the value is slightly higher than the CIL[1] method.

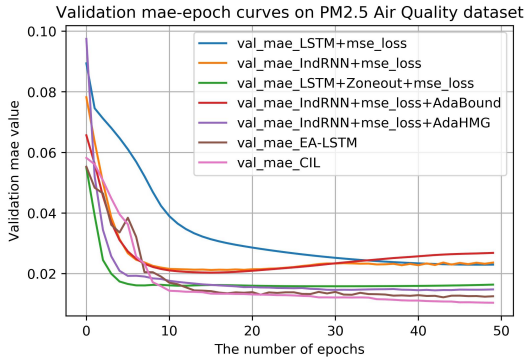


Fig. 7. Comparison of validation MAE curves of the CIL method and baseline models on the PM2.5 Air Quality dataset.

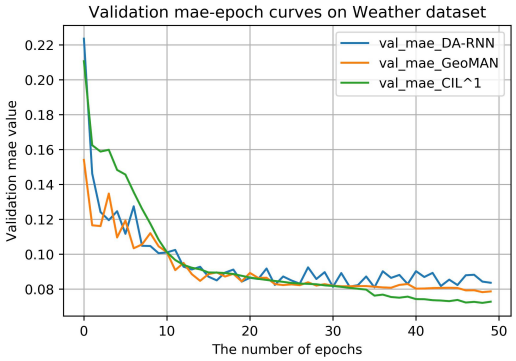


Fig. 8. Comparison curves of the CIL[1] and the SOTA methods validation MAE performance on the Weather dataset.

TABLE V

ROBUSTNESS ANALYSIS: TEST PERFORMANCE OF CIL METHOD ON THREE DATASETS

Datasets	Test-MAE with different SNRs				Without noise
	10	20	40	60	
PM2.5	8.660	4.426	2.058	**1.525**	1.622
Energy	14.607	9.483	**9.305**	15.491	9.555
Weather	26.186	28.807	21.236	21.550	**21.126**

Generally, in a training period of an epoch, the model will perform multiple iteration calculations (i.e., according to the number of the mini-batch samples) based on BPTT to update the model parameters. In order to observe the training accuracy of different optimization algorithms more fine-grained, we show how the training accuracy (i.e., the MAE value on the training dataset) of these algorithms varies with the number of iterations in Fig. 9. Observing Fig. 9, we can find that the AdaMFG algorithm has the lowest initial training MAE value but also has a faster decline speed rate and finally maintained the lowest training MAE value (i.e., the best training performance) after 150×100 iterations. The above phenomenon shows that although different optimization algorithms have oscillations during iterative training based on mini-batch samples, our proposed AdaMFG algorithm has achieved better convergence results and faster entered the convergence state.

We follow the suggestions of reviewers to add experiments (i.e., some type of distribution interference is incorporated into the input data and specific SNRs) to demonstrate the robustness of CIL method in Table V. Referring to the latest research [49], the dynamic Gaussian mix noise based on a

TABLE VI
COMPUTATIONAL OVERHEAD AND PARAMETERS

Models	PM2.5				Energy				Weather			
	Params (b)	Training (s)	Test (s)	Memory (KB)	Params (b)	Training (s)	Test (s)	Memory (KB)	Params (b)	Training (s)	Test (s)	Memory (KB)
1.	-	24.12	0.82	-	-	9.93	0.17	-	-	119.88	0.79	-
2.	-	18.81	0.69	-	-	8.32	0.15	-	-	80.72	0.64	-
3.	70,273	45.30	1.07	287	80,513	32.33	0.36	327	374,405	392.73	1.62	305
4.	1,409	40.40	0.89	18	3,969	23.64	0.33	28	12,805	323.78	1.37	22
5.	76,448	152.87	1.60	289	88,743	139.58	0.45	335	415,648	1,745.17	2.49	308
6.	1,409	42.26	0.93	17	3,969	26.01	0.33	29	12,805	326.54	1.34	24
7.	1,409	42.83	0.94	19	3,969	30.59	0.35	31	12,805	336.08	1.38	25
8.	201,857	1,060.82	21.34	805	214,683	1,212.76	22.76	845	1,018,568	16,204.65	53.94	857
9.	29,235	590.04	1.16	487	31,923	297.47	1.15	497	152,575	3,298.59	1.72	499
10.	34,227	977.54	1.79	519	39,603	514.01	1.79	521	183,935	6,005.82	2.41	533
11.	72,560	**63.74**	**1.08**	**284**	84,268	**35.95**	**0.37**	**325**	390,348	**468.30**	**1.56**	**302**

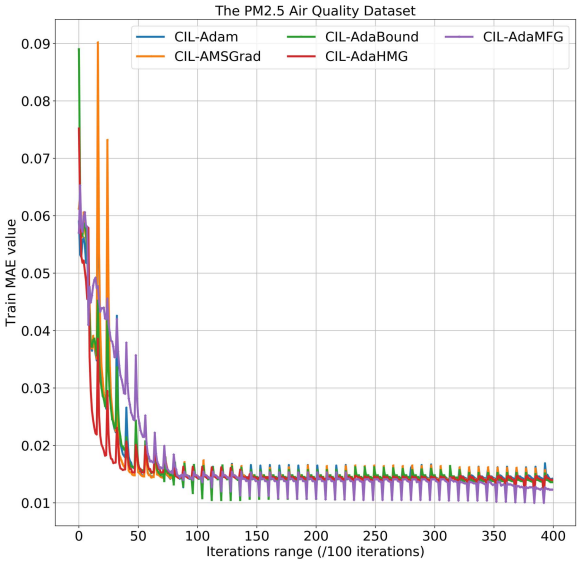


Fig. 9. Training MAE curves of different optimization algorithms on the PM2.5 Air Quality dataset.

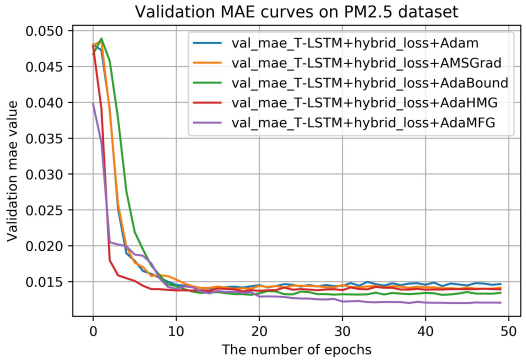


Fig. 10. Validation MAE curves obtained by the CIL method using different optimization algorithms on the PM2.5 Air Quality dataset.

series of SNRs is used to add to the original time series data. It is found that the accuracy of the CIL method declines obviously when the SNR is 10 and better prediction performance can be achieved on most datasets when the SNR is 40. The CIL method is competitive with the SOTA model even in the low SNR environment (with abundant noise interference).

D. Computational Complexity

First, a comparative analysis is performed on the computational overhead of the CIL method and all DL methods (time complexity in the Supplementary Material). We give the training cost and inference cost of all baseline models on the training set and the test set, respectively. Due to table space constraints, the names of the models in Table VI are replaced with ordinal numbers. The model name corresponding to the serial number is as follows: 1) ARIMA; 2) GBRT; 3) LSTM + mse-loss; 4) IndRNN + mse-loss; 5) LSTM + Zoneout + mse-loss; 6) IndRNN + mse-loss + AdaBound; 7) IndRNN + mse-loss + AdaHMG; 8) EA-LSTM; 9) DA-RNN; 10) GeoMAN; and 11) CIL. In summary, the following conclusions are drawn. First, the GBRT algorithm has the lowest calculation overhead, and the overhead required by the ARIMA algorithm is only higher than the GBRT. Second, high computational complexity is generated by DL baseline models (i.e., baselines 8–10) due to attention scores calculation modules and encoder–decoder structures. In particular, the EA-LSTM model with the evolutionary—search algorithm—has the highest computational complexity of all models. Finally, the objective function of the CIL method adds some $O(N)$ level calculations. The CIL method has similar computational complexity to neural network methods (i.e., baselines 3–7). It is worth noting that the running time of the CIL method is only about one-tenth of the DL method (i.e., DA-RNN). This means that the CIL method can be used to solve problems in a larger scale time series prediction task, and more accurate perdition results can be obtained rapidly with fewer GPU computing resources. Finally, we save model h5 files using checkpoint techniques and report the required memory (i.e., storage capacity occupied by checkpoint files). Table VI shows the advantages of CIL over attention-based models [10], [11], [27] in terms of computational and storage overheads. These advantages come from the design of our objective function, T-LSTM model, and AdaMFG algorithm.

E. Ablation Study

In order to prove Viewpoint 1, a comprehensive component analysis of the CIL method on the largest dataset (i.e., Weather dataset) is conducted, as shown in Table VII. Different from general ablation experiments, the following eight variants (i.e., contains or excludes the components proposed in this article) are given by the permutation and combination to analyze the effectiveness of the proposed component collaboration: 1) LSTM + MSE_loss + Adam; 2) T-LSTM + MSE_loss + Adam; 3) LSTM + MSE_loss + AdaMFG; 4) T-LSTM + MSE_loss + AdaMFG; 5) LSTM + Hybrid-loss + Adam; 6) T-LSTM + Hybrid-loss + Adam;

TABLE VII
COMPARISON OF CIL METHOD COMPONENTS ON THE WEATHER DATASET

Components	$\lvert\hat{y}-y\rvert$	Test-MAE	Test-RMSE
1.	106.783	114.640±1.227	163.632±1.709
2.	101.029	112.225±0.271	162.417±0.563
3.	91.593	111.635±0.854	157.715±1.235
4.	91.239	110.886±0.246	155.972±0.382
5.	50.157	50.947±0.516	127.536±0.953
6.	39.855	41.572±0.152	99.767±0.309
7.	38.411	35.025±0.429	78.920±0.813
8.	15.056	**21.126±0.114**	**36.787±0.187**

TABLE VIII
COMPARISON OF PREDICTION PERFORMANCE OF CIL METHOD AND SOTA METHOD ON THE TWO ADDITIONAL DATASETS

Datasets	Models	Test-MAE	Test-RMSE
Traffic	EA-LSTM	75.416	96.835
	CIL	**53.943**	**73.401**
Water Quality	EA-LSTM	2.412	3.414
	CIL	**1.556**	**2.452**

7) LSTM + Hybrid_loss + AdaMFG; and 8) T-LSTM + Hybrid-loss + AdaMFG (i.e., CIL method). The absolute value form of the error (mean value) between the prediction and ground truth is provided for each component of the 24-h forecast on the Weather dataset in Table VII. Meanwhile, the names of the components are numbered in Table VII. In Table VII, the following conclusions are obtained.

1) Compared with variants 1 and 2, the switching of T-LSTM only has a limited improvement in prediction accuracy.
2) Compared with variants 3 and 4, T-LSTM combined with AdaMFG algorithm can further improve the Test-RMSE metric.
3) Compared with variants 4–6, the hybrid loss has the most contributions in improving prediction accuracy. Thus, the introduction of a better objective function into the neural network model is a key factor for performance improvement. Moreover, the combination of T-LSTM and Hybrid loss can also significantly improve the Test-RMSE metric, indicating that the transformation mechanism can effectively capture the abrupt and slow change information under the guidance of Hybrid loss.
4) Comparing variant 8, through the cooperation of three perspectives, the best prediction performance can be achieved on the Weather dataset. This proves that Viewpoint 1 can better learn the complex temporal dependencies in multivariate time series data.

We have added figures comparing the performance of the AdaMFG algorithm and multiple baseline optimization algorithms to verify the advantages of the proposed algorithm. In addition, in terms of the convergence effect of the algorithms, in Fig. 10, the proposed AdaMFG algorithm has the smallest initial error and enters the convergence state after 20 epochs to gradually achieve the best convergence performance. At the same time, combined with the experimental phenomenon in Fig. 9, the proposed AdaMFG algorithm has a statistically significant improvement in prediction accuracy.

F. Additional Experiments

To fully verify the effectiveness of the proposed CIL method, experiments on traffic volume prediction (from UCI) [47] and water quality prediction tasks (from Kaggle) [48] have been added in the revised manuscript. Finally, five complex multivariate time series datasets in different fields, different sizes, and different numbers of dimensions are used to verify the proposed CIL method. We show the basic information of these two datasets in Table VIII (see for detailed performances). For more detailed information, please refer to the corresponding web links of [47] and [48].

VI. CONCLUSION

In this article, a new CIL method with lower time overhead and higher prediction accuracy is proposed to solve the multivariate time series prediction problem. In this study, an objective function, model, and adaptive stochastic optimization algorithm are designed for comprehensively capturing the temporal change information, and the three perspectives are reasonably organized into a CIL method to infer the temporal change rules in the future time. Through comprehensive experiments, the following conclusions are obtained. First, the novel objective function can be used to effectively guide the model to find the abrupt and slow change information. Therefore, the comparison predicted curve of the CIL method is the closest to the true value in Section VI. Second, the AdaMFG algorithm not only has a better convergence effect than that of the Adam-type algorithm in multivariate time series prediction but also indirectly learns the temporal change information between continuous errors gradient flow. Through comprehensive component analysis experiments, we found that the best prediction results can only be achieved when the three components cooperate with each other. This inspired us to design a reasonable solution for data characteristics from multiple perspectives for different types of time series prediction tasks, rather than just considering the improvement of neural network model.

REFERENCES

[1] I. Koprinska, D. Wu, and Z. Wang, "Convolutional neural networks for energy time series forecasting," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.
[2] Y. Wu, Y. Liu, S. H. Ahmed, J. Peng, and A. A. A. El-Latif, "Dominant data set selection algorithms for electricity consumption time-series data analysis based on affine transformation," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4347–4360, May 2020, doi: 10.1109/JIOT.2019.2946753.
[3] L. S. Saoud and R. Ghorbani, "Metacognitive octonion-valued neural networks as they relate to time series analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 2, pp. 539–548, Feb. 2020.
[4] J. Hu and W. Zheng, "Transformation-gated LSTM: Efficient capture of short-term mutation dependencies for multivariate time series prediction tasks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.
[5] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. 6th Int. Conf. Learn. Represent.*, Apr./May 2018, pp. 1–8.
[6] S. Xiao, J. Yan, M. Farajtabar, L. Song, X. Yang, and H. Zha, "Learning time series associated event sequences with recurrent point process networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 10, pp. 3124–3136, Oct. 2019.
[7] K. Nam and N. Seong, "Financial news-based stock movement prediction using causality analysis of influence in the Korean stock market," *Decis. Support Syst.*, vol. 117, pp. 100–112, Feb. 2019.

[8] J. Zhang and K. F. Man, "Time series prediction using RNN in multi-dimension embedding phase space," in *Proc. SMC*, 1988, pp. 1868–1873.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[11] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. W. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," in *Proc. IJCAI*, Aug. 2017, pp. 2627–2633.

[12] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng, "GeoMAN: Multi-level attention networks for geo-sensory time series prediction," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3428–3434.

[13] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 26, no. 1, pp. 43–49, 1978.

[14] L. Frías-Paredes, F. Mallor, M. Gastón-Romeo, and T. León, "Assessing energy forecasting inaccuracy by simultaneously considering temporal and absolute errors," *Energy Convers. Manage.*, vol. 142, pp. 533–546, Jun. 2017.

[15] A. Mensch and M. Blondel, "Differentiable dynamic programming for structured prediction and attention," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 3459–3468.

[16] F. Rivest and R. Kohar, "A new timing error cost function for binary time series prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 1, pp. 174–185, Jan. 2020.

[17] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of Adam and beyond," in *Proc. 6th Int. Conf. Learn. Represent.*, Apr./May 2018, pp. 1–8.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, May 2015, pp. 1–8.

[19] L. Luo, Y. Xiong, Y. Liu, and X. Sun, "Adaptive gradient methods with dynamic bound of learning rate," in *Proc. 7th Int. Conf. Learn. Represent.*, 2019, pp. 1–8.

[20] S. Fang, Q. Zhang, G. Meng, S. Xiang, and C. Pan, "GSTNet: Global spatial-temporal network for traffic flow prediction," in *Proc. IJCAI*, Aug. 2019, pp. 1–8.

[21] A. Abid and J. Zou, "Learning a warping distance from unlabeled time series using sequence autoencoders," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 10547–10555.

[22] M. Cuturi and M. Blondel, "Soft-DTW: A differentiable loss function for time-series," in *Proc. ICML*, 2017, pp. 894–903.

[23] T. Durand, N. Thome, and M. Cord, "Exploiting negative evidence for deep latent structured models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 337–351, Feb. 2019.

[24] V. L. Guen and N. Thome, "Shape and time distortion loss for training deep time series forecasting models," in *Proc. NeurIPS*, 2019, pp. 4191–4203.

[25] D. Krueger *et al.*, "Zoneout: Regularizing RNNs by randomly preserving hidden activations," in *Proc. Int. Conf. Learn. Represent.*, Apr. 2017, pp. 1–8.

[26] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "Independently recurrent neural network (IndRNN): Building a longer and deeper RNN," in *Proc. CVPR*, Jun. 2018, pp. 5457–5466.

[27] Y. Li, Z. Zhu, D. Kong, H. Han, and Y. Zhao, "EA-LSTM: Evolutionary attention-based LSTM for time series prediction," *Knowl.-Based Syst.*, vol. 181, Oct. 2019, Art. no. 104785.

[28] M. Johnson *et al.*, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 339–351, Dec. 2017.

[29] J. Hu and W. Zheng, "An adaptive optimization algorithm based on hybrid power and multidimensional update strategy," *IEEE Access*, vol. 7, pp. 19355–19369, 2019.

[30] K. Miyaguchi and H. Kajino, "Cogra: Concept-drift-aware stochastic gradient descent for time-series forecasting," in *Proc. AAAI*, 2019, pp. 4594–4601.

[31] Z. Zhou, Q. Zhang, G. Lu, H. Wang, W. Zhang, and Y. Yu, "AdaShift: Decorrelation and convergence of adaptive learning rate methods," in *Proc. ICLR*, 2019, pp. 1–8.

[32] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, nos. 1–2, pp. 100–115, 1954. [Online]. Available: http://www.jstor.org/stable/2333009

[33] Y. Jiao, Y. Chen, and Y. Gu, "Subspace change-point detection: A new model and solution," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1224–1239, Dec. 2018.

[34] H. Liao, M.-K. Liu, M. S. Mariani, M. Zhou, and X. Wu, "Temporal similarity metrics for latent network reconstruction: The role of time-lag decay," *Inf. Sci.*, vol. 489, pp. 182–192, Jul. 2019.

[35] S. Du, G. Song, L. Han, and H. Hong, "Temporal causal inference with time lag," *Neural Comput.*, vol. 30, no. 1, pp. 191–271, 2018.

[36] K. Papafitsoros and C.-B. Schönlieb, "A combined first and second order variational approach for image reconstruction," *J. Math. Imag. Vis.*, vol. 48, no. 2, pp. 308–338, Feb. 2014.

[37] T. F. Chan, S. Osher, and J. Shen, "The digital TV filter and nonlinear denoising," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 231–241, Feb. 2001.

[38] Y. Zhou and H. Zou, "Non-parametric outliers detection in multiple time series a case study: Power grid data analysis," in *Proc. AAAI*, 2018, pp. 4605–4612.

[39] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. ICML*, 2003, pp. 928–936.

[40] C. Rothwell. *Met Eireann*. Accessed: 2008. [Online]. Available: https://www.met.ie/climate/available-data/historical-data

[41] X. Liang *et al.*, "Assessing Beijing's $PM_{2.5}$ pollution: Severity, weather impact, APEC and winter heating," *Proc. Roy. Soc. A, Math. Phys. Eng. Sci.*, vol. 471, no. 2182, 2015, Art. no. 20150257.

[42] Y. Liu, C. Gong, L. Yang, and Y. Chen, "DSTP-RNN: A dual-stage two-phase attention-based recurrent neural networks for long-term and multivariate time series prediction," *CoRR*, vol. abs/1904.07464, pp. 1–20, Apr. 2019.

[43] M. Abadi *et al.* (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: https://www.tensorflow.org/

[44] F. Chollet *et al.* (2015). *Keras*. [Online]. Available: https://github.com/fchollet/keras

[45] G. E. P. Box and D. Pierce, "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models," *J. Amer. Stat. Assoc.*, vol. 65, no. 332, pp. 1509–1526, 1968.

[46] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[47] J. Hogue. *Metro Interstate Traffic Volume Data Set, UCI Dataset*. Accessed: May 7, 2019. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Vol.

[48] L. Zhang. Real time water quality data. Kaggle. Accessed: Jul. 1, 2020. [Online]. Available: https://www.kaggle.com/ivivan/real-time-water-quality-data

[49] Y. Wu *et al.*, "Dynamic Gaussian mixture based deep generative model for robust forecasting on sparse multivariate time series," in *Proc. AAAI*, 2021, pp. 651–659.

Wendong Zheng (Graduate Student Member, IEEE) was born in Taiyuan, Shanxi, China, in 1994. He received the B.E. degree in software engineering from the North University of China, Taiyuan, in 2017, and the master's degree in computer science from Hunan University, Changsha, China, in 2020.

His main research interests are time series prediction and first-order stochastic optimization algorithm.

Prof. Zheng is serving as a Reviewer for the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, *Knowledge-Based Systems*, and IEEE ACCESS.

Jun Hu (Member, IEEE) was born in 1971. He received the M.Sc. degree in computer application from the Kunming University of Science and Technology, Kunming, China, in 2003, and the Ph.D. degree in computer science and technology from Zhejiang University, Hangzhou, China, in 2006.

In 2010, he was an Academic Visitor with the University of Southampton, Southampton, U.K., working on a multiagent system. He is currently an Associate Professor with Hunan University, Changsha, China. His research interests include multiagent systems, deep learning, and software engineering.

Dr. Hu is a Senior Member of the China Computer Federation (CCF).