Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 9523230, 16 pages
https://doi.org/10.1155/2023/9523230

Research Article

Spatiotemporal Self-Attention-Based LSTNet for Multivariate Time Series Prediction

Dezheng Wang and Congyan Chen

School of Automation, Southeast University, Nanjing 210096, China

Correspondence should be addressed to Congyan Chen; chency@seu.edu.cn

Received 14 November 2022; Revised 17 April 2023; Accepted 28 April 2023; Published 23 May 2023

Academic Editor: Lianyong Qi

Multivariate time series prediction is a critical problem that is encountered in many fields, and recurrent neural network (RNN)-based approaches have been widely used to address this problem. However, traditional RNN-based approaches for predicting multivariate time series are still facing challenges, as time series are often related to each other and historical observations in real-world applications. To address this limitation, this paper proposes a spatiotemporal self-attention mechanism-based LSTNet, which is a multivariate time series forecasting model. The proposed model leverages two self-attention strategies, spatial and temporal self-attention, to focus on the most relevant information among time series. The spatial self-attention is used to discover the dependences between variables, while temporal attention is employed to capture the relationship among historical observations. Moreover, a standard deviation term is added to the objective function to track multivariate time series effectively. To evaluate the proposed method's performance, extensive experiments are conducted on multiple benchmarked datasets. The experimental results show that the proposed method outperforms several baseline methods significantly. Therefore, the proposed spatiotemporal self-attention-based LSTNet is a promising approach for predicting multivariate time series.

1. Introduction

Multivariate time series plays a crucial role in daily life and is a subject of active research. Forecasting multivariate time series has numerous applications, including predicting stock prices [1], weather forecasting [2], traffic prediction [3], complex industrial system analysis [4], and COVID-19 widespread disease prediction [5, 6]. Accurate predictions or trends play a vital role in helping individuals and businesses make informed decisions and judgments. For instance, investors rely on accurate predictions or trends to make investment decisions that can result in reasonable returns [1, 7]. Similarly, commuters can choose suitable travel routes based on predicted traffic conditions to avoid congestion and save time [8]. In addition, energy factories can adjust their production strategy based on predicted energy consumption to optimize their operations and reduce costs [9]. In essence, accurate predictions or trends can help individuals and businesses make proactive and well-informed decisions that lead to better outcomes. This can lead to increased efficiency, productivity, and profitability, as well as improved quality of life for individuals.

Deep neural networks (DNNs) have advanced significantly in recent years [10, 11] and have had a significant impact on solving a wide range of time series forecasting tasks. There are three commonly used DNN structures for time series analysis: convolutional neural network (CNN)-based approaches, RNN-based approaches, and transformer-based approaches. RNN-based approaches such as RNN [12] and its variants, namely, long- and short-term memory (LSTM) [13] and gated recurrent unit (GRU) [14], are widely used to analyze time series. Meanwhile, CNN-based approaches, such as temporal convolutional networks (TCN) [15], implement time series analysis using dilation convolution operation. As the volume of data increases, the transformer-based approaches [16] are developed to discover the relationship among time series by computing the score matrix. By capturing the temporal dependencies, these sequence models have significantly improved the performance in multivariate time series analysis tasks. Furthermore, integrating an attention mechanism with the RNN structure can capture the long-distance dependence [17–21].

However, multivariate time series forecasting is facing a significant research challenge. The existing methods mainly focus on temporal relationships to discover the importance of different sampling times [22, 23]. Nevertheless, there are several variables being measured at the same sampling time in multivariate time series, as shown in Figure 1. Here, x_t^k denotes the value of the k–th variable at sampling time t, and the different color presents different importance to prediction results. Therefore, it is essential to identify which variables are relevant and which ones are not in order to make accurate predictions. Some variables maybe irrelevant or even noisy, having little impact on the outcome, while others may play a crucial role in determining the future trend. Therefore, capturing the correlation between each variable and the target variables is necessary, as it allows for a more comprehensive understanding of the underlying patterns and trends. Overall, it is necessary to consider temporal relationships and variable dependencies in reliable prediction methods.

To better illustrate the abovementioned issue, the SML2010 datasets as an example are used to explain the temporal relationship and variable dependencies (more details about these datasets are listed in Section 4.1). The several variables of the SML2010 are presented in Figure 2. It appears that there are two distinct types of dependencies: spatial and temporal. The recurring pattern of variable 1 suggests a temporal dependency, meaning that it varies over time. Moreover, the strong correlation between variable 1 and variables 7, 11, and 17 indicates a spatial dependency, which means that they are related in a variable sense. The key to an effective time series prediction method is to capture both kinds of dependencies' patterns among input. This work aims to uncover temporal relationships and capture variable interdependencies among the input data to ensure optimal prediction performance. The spatial relations reflect the correlation of different variables, and temporal relations reflect the dependencies of historical measurements. Again, without considering both kinds of patterns, the prediction accuracy will be noticeably reduced. Nevertheless, the classical methods fall short in this aspect and do not focus on these strong correlations that play a vital role in prediction [24]. This work focuses on addressing the mentioned problem of multivariate time series forecasting.

The transformer-based architecture has shown remarkable performance in time series processing tasks. However, it is true that training a high-performance transformer model requires a significant amount of data, which can be challenging to obtain in specific domains. A comparative study to compare the performance of the transformer and LSTM on insufficient samples is listed in Section 4.5. Owing to the better performance of LSTM, this paper utilizes the benefits of the LSTM model to propose a prediction model. Consequently, the proposed spatiotemporal self-attention is to integrate with long- and short-term temporal neural networks (LSTNet) [24] that take advantage of the recent LSTM framework. There are three main contributions to improving the performance of multivariate time series prediction models.

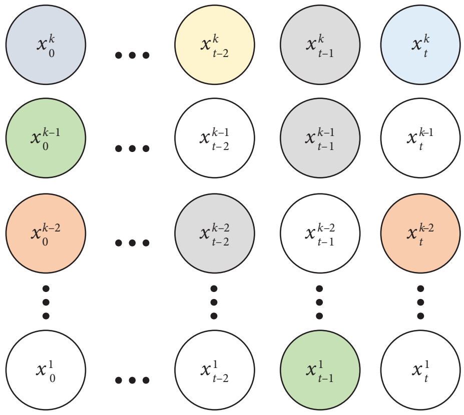

Figure 1: The complex relationship in multivariate time series. The different color presents different importance to prediction results. Some variables may have little impact on the outcome, while others may play a crucial role.

(1) The proposed spatiotemporal self-attention-based LSTNet is an effective method for analyzing complex data structures by extracting dependencies between historical observations and obtaining correlations among variables. This approach uses spatial self-attention to identify relationships between variables and temporal self-attention to capture relationships among historical observations. The proposed approach outperformed nine baseline models, including transformer, on three benchmarked datasets, demonstrating its effectiveness in capturing spatiotemporal relationships in multivariate time series data.

(2) A new objective function has been proposed to improve multivariate time series forecasting. The function considers the standard deviation of the loss for each variable, ensuring that losses for each variable are minimized, in addition to the total loss. This approach addresses the issue of imbalanced errors among variables, resulting in improved accuracy and robustness of the forecasting model. Overall, the contributions of this new objective function represent a significant advancement in the field of multivariate time series forecasting.

(3) Extensive experiments show that due to its lightweight and efficient structure, LSTM-based methods remain the most potent tool for tackling prediction problems, achieving similar performance with the transformer when trained on insufficient samples.

The rest of this paper is structured as follows: Section 2 presents the related work, while Section 3 introduces the details of the proposed method. In Section 4, a comparative study, ablation study, main results, and additional experimental details are listed. Finally, the conclusion is shown in Section 5.

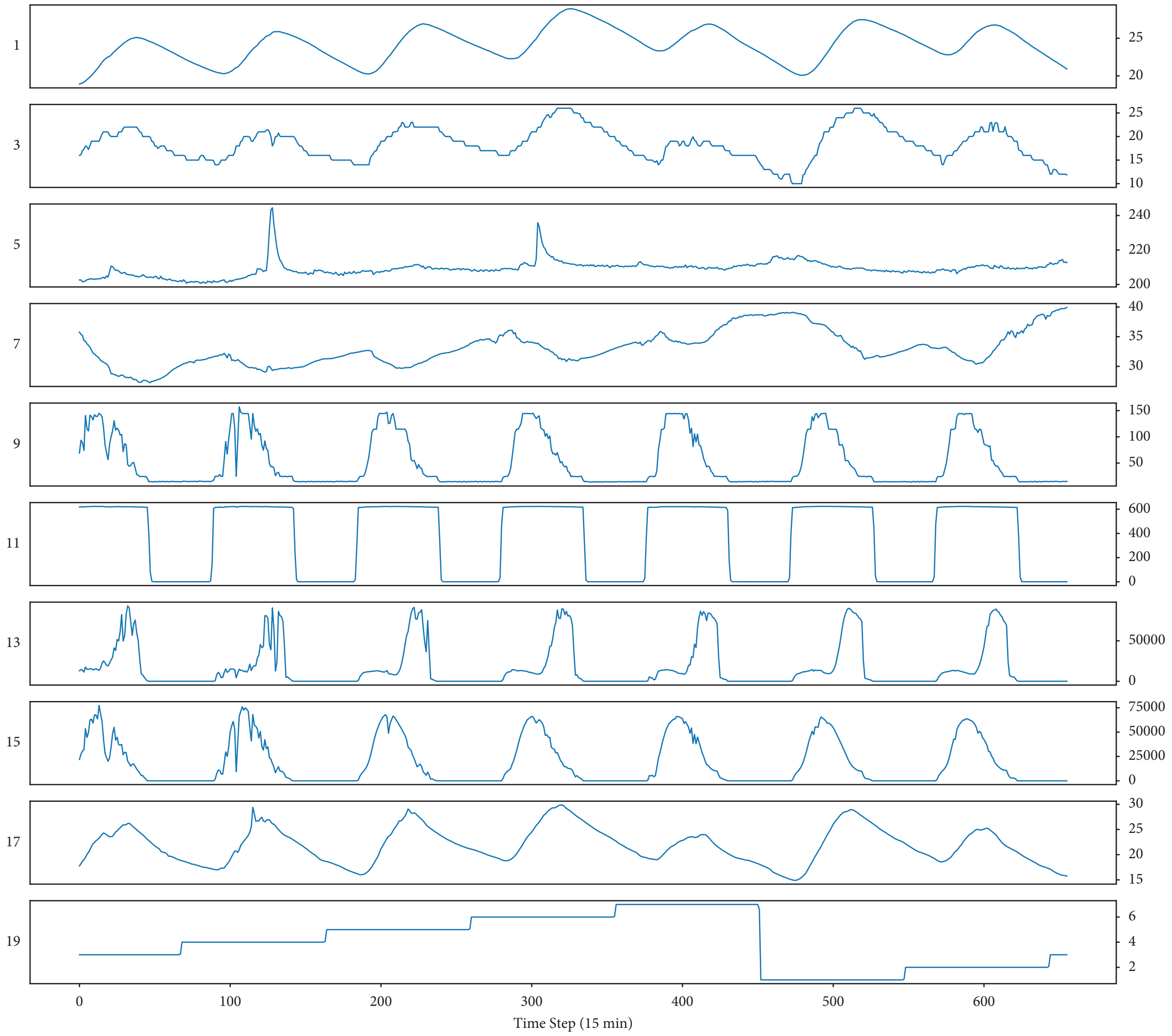

Figure 2: The trend curve of the SML2010 datasets (part). It presents the trend curve of ten variables, including 1, 3, 5, 7, 9, 11, 13, 15, 17, and 19. There is a repeating pattern in variable 1. In addition, variable 1 appears to have a strong correlation with variables 7, 11, and 17.

2. Related Background

The most commonly used multivariate time series forecasting method is ARIMA, one of the autoregressive models. It is a typical linear model that cannot solve most of the multiple time series prediction tasks in everyday life [25]. To satisfy multivariate time series analysis needs, a growing number of researchers have recently enrolled in the research of multivariate time series prediction methods based on DNN instead of machine learning algorithms [26]. RNN is one of the classical sequence deep learning models. It extracts informative patterns from sequential data, and one of the main disadvantages is prone to vanishing or exploding gradient problems [27]. The modified RNN version, namely, LSTM, is widely used to overcome the vanishing gradient problem in the field of sequential series analysis. Sagheer and Kotb [28] utilized the genetic algorithm to select the best LSTM structure in the petroleum industry. Liu et al. [29] used LSTM for greenhouse climate prediction, which performs better than others. Much relevant researches are also based on LSTM and its variants [30–34]. Nevertheless, the performance of the LSTM model decreases as the lengths of the inputs of the model increase. To deal with this problem, Chung et al. [14] claimed a GRU for predicting time series by using a gating mechanism, which has a simpler architecture than LSTMs, which can make it faster to train and more efficient in some cases. These methods do not distinguish the long-term and short-term features. The LSTNet is presented to address this problem, which combines the strengths of both RNN and CNN to capture the short-term features and long-term dependencies [24].

To improve the accuracy of time series forecasting, many researchers have incorporated self-attention mechanisms into LSTM models [24, 25], which used attention mechanisms (1) for time series forecasting [20]. By incorporating attention mechanisms, these models can learn which time steps are the most important for making predictions. Li et al. [19] proposed an attention-based LSTM for time series forecasting tasks. The single-attention approach will magnify irrelevant information in some cases. Qin et al. [20] proposed the dual-stage attention-based RNN (DA-RNN) to avoid this problem.

$$e_t = v_e^t \tanh\left(W_e\left[h_{t-1}; s_{t-1}\right] + U_e x_t\right),$$
$$\alpha_t = \frac{\exp\left(e_t\right)}{\sum_{i=1}^{T}\exp\left(e_t\right)}, \tag{1}$$

where h_t and s_t denote the hidden states of the LSTM at time stamp t, respectively, in (1). v_e, W_e, and U_e are parameters to learn. α_t is the attention score indicating the relative importance of the sample at time t. From (1), it is evident that this attention approach is used to obtain temporal dependencies and it cannot distinguish the difference between variables.

Meanwhile, lots of CNN-based methods have been applied to solve the time series prediction problems [35]. One such method is the TCN, which utilizes dilation convolution to extract features from the entire input sequence to forecast target variables [36]. The dilation convolution operation in TCN can be useful for capturing long-term dependencies and improving the model's ability to predict future values accurately without increasing the number of parameters in the model [15]. By considering the entire input sequence, TCN can effectively learn complex temporal patterns and relationships between past and future time steps. This makes it a powerful tool for time series forecasting tasks [37, 38]. Assaf et al. reported that the MTEX-CNN is used for making multivariate time series predictions, which consist of two stages and utilize particular kernel sizes [39].

Transformers are a type of deep learning model that has been shown to be highly effective for processing time series data [16]. Zhou et al. [40] reported that the informer, a variant of transformer, improves the inference speed of long-sequence predictions by using the ProbSparse self-attention mechanism and self-attention distilling. Shen and Wang [41] proposed TCCT for forecasting time series with much lower computation and memory costs, which applies transformed CNN architectures into a transformer. Moreover, Lam et al. [42] reported the GraphCast for the medium-range global weather forecasting task, which employs a graph neural network (GNN). Due to requiring a large amount of data for training, it may not be suitable for most prediction cases.

As mentioned previously, multivariate time series often entails a complex relationship pattern containing temporal and spatial relations in the real world. To address this problem, the DA-RNN is proposed to extract the spatial features using a CNN structure [17]. Shi et al. [43] employed convolutional LSTM (convLSTM) to capture spatiotemporal correlations. This method replaces the full connection unit with a convolutional structure in the LSTM model for the precipitation nowcasting problem and consistently outperforms FC-LSTM. Hou et al. [34] proposed a modified graph convolutional network (GCN) to discover the correlation between variables for the stock market prediction task. They claimed that the modified graph convolutional network achieves superior improvement over baseline methods.

Although these forecasting methods that use the convolutional methods to obtain spatial relations are effective in capturing spatial dependencies between variables, it ignores the fact that different variables have different levels of correlation. In the context of time series prediction, it is essential to focus on variables with high correlation and disregard those with low correlation, similar to standard attention approaches. However, it is important to note that there is no inherent physical location relationship between time series variables. Therefore, most attention approaches that have been successful in the image domain may not be as effective in the time series domain.

In general, these standard self-attention approaches mainly focus on acquiring temporal dependencies. Most of them do not take spatial correlations into consideration, although it is as crucial as temporal dependencies. Therefore, in this work, a novel spatiotemporal self-attention approach is proposed to increase the application scope of the self-attention algorithms. Moreover, the objective function is designed specially to model multivariate time series prediction.

3. Framework

The multivariate time series forecasting problem is formulated in this section. What follows are the details of the proposed spatiotemporal self-attention approach. Finally, the modified objective function is introduced.

3.1. Problem Formulation. Given a fully observed multivariate time series $X = \{x_1, x_2, \ldots, x_n\}$ and assume these series are available, where $x_n \in \mathbb{R}^m$ and $X \in \mathbb{R}^{m*n}$. x_n denotes the value of all variables x at sampling time n, and m is the number of variables. x^i represents the i–th variable and x_t^i to represent the i–th variable at sampling time t. Moreover, the objective is to predict a series x_{n+h+1} by sliding window w, where h is the forecast horizon ahead of the current sampling time stamp. Namely, this task needs to predict m variables at sampling time $n+h+1$. By the convention, let $y_n = x_{n+h+1} \in \mathbb{R}^m$, and the sliding window size w is defined as n. Therefore, this forecasting problem can be formulated as follows:

$$y_n = x_{n+h+1} = \Phi(X), \tag{2}$$

where $\Phi\colon \mathbb{R}^{m*n} \longrightarrow \mathbb{R}^m$ is the mapping function that is needed to fit via the proposed approach, and $X = \{x_1, x_2, \ldots, x_n\}$.

3.2. Model Structure

3.2.1. Standard Self-Attention Mechanism. The self-attention method is adopted in most sequence models with excellent performance in terms of dependence reduction and parallel training, such as the transformer model [44]. Therefore, the spatiotemporal self-attention is designed based on scaled dot product attention [45]. There are three vectors, Q, K, and V, which are generated by encoding the original embedding vector input X in standard self-attention. Then, self-attention computes the attention score vector S via the similarity function of Q and K.

$$S = \mathrm{softmax}\left(\frac{\left(Q, K^T\right)}{\sqrt{d}}\right), \tag{3}$$

where d denotes the size of Q, and the self-attention output vector can be written as follows:

$$O = S \times V = \Omega(X), \tag{4}$$

where $\Omega(\cdot)$ indicates the scaled dot product attention operation.

3.2.2. *Temporal Self-Attention Unit.*

Given multivariate time series $X = \{x_1, x_2, \ldots, x_n\}$ as shown in the following equation:

$$X = \begin{pmatrix} x_1^1 & x_2^1 & \cdots & x_n^1 \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \cdots & \cdots & \cdots & \cdots \\ x_1^m & x_2^m & \cdots & x_n^m \end{pmatrix}, \tag{5}$$

$$X_{\text{temporal}} = \{x_1, x_2, \cdots, x_n\}, \tag{6}$$

where x_t^i represents the value of i–th variable at sampling time t. x_t denotes the value of all input variables at sampling time t. X_{temporal} is a temporal vector. The temporal self-attention (TSA) unit is shown in Figure 3.

The TSA automatically determines the relative importance of historical measurements for predicting future outcomes. This is achieved through a process of calculating correlation scores between each historical measurement and the predicted values. The TSA then assigns higher weights to those measurements that demonstrate stronger correlation, indicating their greater significance in predicting future outcomes. This is achieved through the application of equation (4), which adjusts the weights accordingly. The historical measurements with higher correlation scores will have a greater impact on the forecast results, as their contribution to the overall predictive model is deemed to be more significant. This weighting mechanism helps ensure that the prediction model is able to make accurate predictions by prioritizing the most important historical data in the forecasting process.

Owing to the better performance of the LSTNet in the aspect of temporal correlation extraction, the identity function, which is used to encode X_{temporal}, is good enough. Therefore, the temporal dependency vectors C_{temporal} are computed by the following expression:

$$C_{\text{temporal}} = \Omega\left(I\left(X_{\text{temporal}}\right)\right) = S_{\text{temporal}} \times X_{\text{temporal}}, \tag{7}$$

where $I(x) = x$ is the identity encode function and S_{temporal} denotes the score matrix of temporal dependencies.

3.2.3. *Spatial Self-Attention Unit.*

Given X as shown in (5), the spatial vector X_{spatial} can be formulated as follows:

$$X_{\text{spatial}} = \left\{x^1, x^2, \ldots, x^m\right\}^T, \tag{8}$$

where x^i represents the value of the i–th variable. Figure 4 shows the spatial self-attention (SSA) unit.

The SSA can automatically calculate the impact of various variables on prediction results, similar to the TSA. However, the key difference lies in the fact that SSA takes into account spatial correlations, allowing it to adjust weights in a way that more effectively distinguishes between important and unimportant variables. By leveraging these spatial correlations, the SSA approach can more accurately identify which variables are most critical to a given forecast and thus assign greater weight to those variables while downplaying the influence of less significant ones. This enables a more targeted and efficient forecasting approach that can provide highly accurate predictions while minimizing the impact of irrelevant factors.

The X_{spatial} is encoded by an LSTM approach to enrich information and obtain spatial correlations. This strategy will enrich the information on each variable in X_{spatial} to effectively obtain the underlying correlations between variables via SSA. The spatial correlations are computed by the following expression:

$$C_{\text{spatial}} = \Omega\left(\mathrm{LSTM}\left(X_{\text{spatial}}\right)\right) = S_{\text{spatial}} \times h_{\text{spatial}}, \tag{9}$$

where $\mathrm{LSTM}(\cdot)$ is the encode function by using the LSTM approach and h_{spatial} is the output of LSTM when taking X_{spatial}. S_{spatial} and C_{spatial} denote the score matrix of spatial correlations and spatial correlation vector computed by spatial self-attention, respectively.

3.2.4. *Proposed Model.*

The architecture of the proposed approach, based on LSTNet, is presented in Figure 5. Initially, the multivariate time series X is passed through both TSA and SSA to extract spatiotemporal features, C_{temporal} and C_{spatial}. The resulting C_{temporal} is then fed into a convolutional component to capture short-term temporal patterns. The output of the convolutional component is then sent to both the recurrent and recurrent-skip components of LSTNet to capture complex temporal dependencies. The recurrent component is capable of memorizing historical information, while the recurrent-skip component captures long-term patterns. Meanwhile, the C_{spatial} is sent to an MLP component in LSTNet to capture spatial correlations. Consequently, the proposed approach considers both temporal and spatial features while also focusing on the strongly correlated variables and sampling times.

Finally, these two aspect features are sent to the last layer using the concatenate operation to obtain the prediction result $\widehat{y}_n = \widehat{x}_{n+h+1}$ of the sampling time step $n + h + 1$.

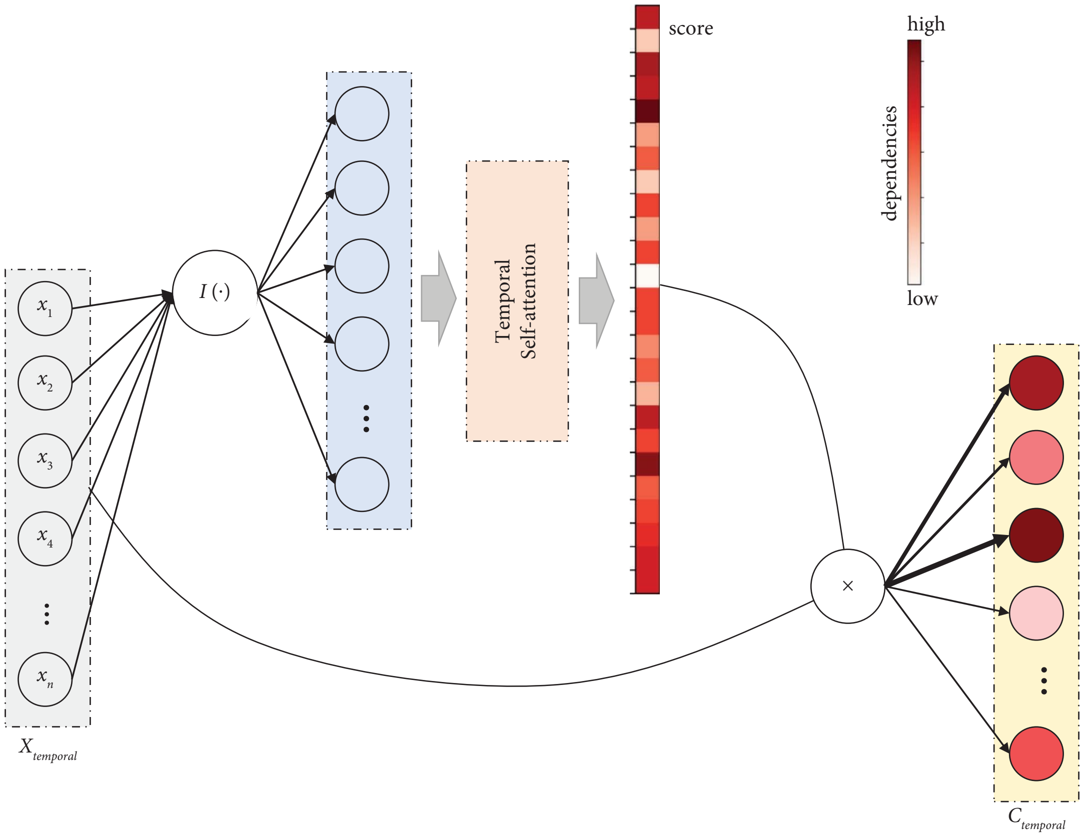

Figure 3: The structure of the TSA. After calculating the importance of different historical measurements on the prediction results, the TSA adjusts the corresponding weight. The historical measurements with high correlation have higher scores.

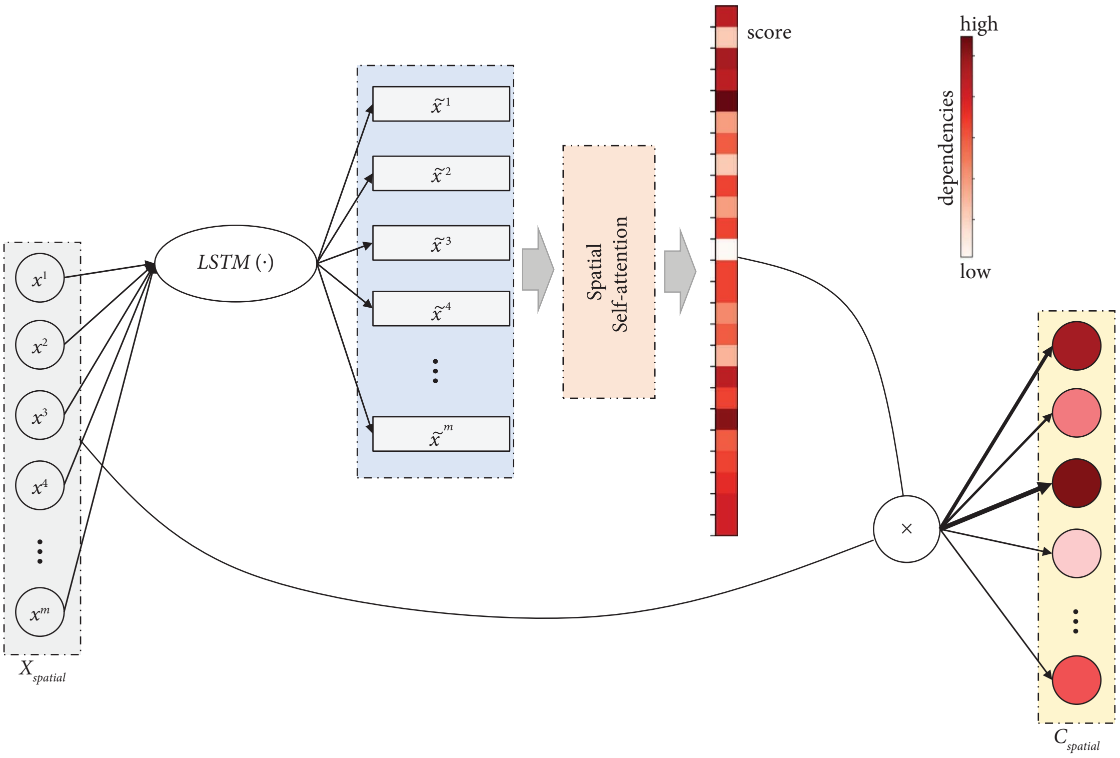

Figure 4: The structure of the SSA. After calculating the impact of different variables on the prediction results, the SSA adjusts the weights according to the spatial correlations.

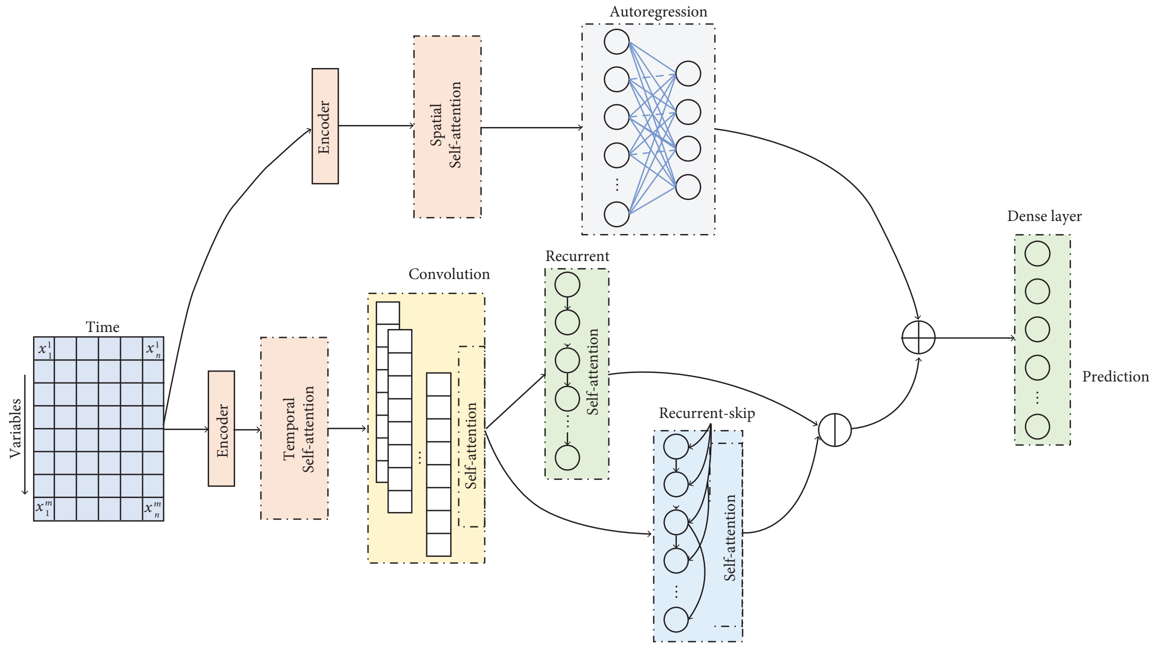

Figure 5: An overview of the proposed approach structure. The input is fed into TSA and SSA units to extract the spatiotemporal features. Then, C_{temporal} is sent to convolutional, recurrent, and recurrent-skip components to obtain complex temporal dependencies. Meanwhile, C_{spatial} is fed into the MLP to discover the spatial correlations. Finally, these features are sent to the last layer to calculate the prediction result.

3.3. Objective Function. The squared error as the default objective function is in common use for most of the time series prediction tasks, and the optimization objective function is formulated as follows:

$$\underset{\Theta}{\operatorname{argmin}}\, J\left(y_n, \widehat{y}_n\right) = \underset{\Theta}{\operatorname{argmin}}\, \left\|y_n - \widehat{y}_n\right\|_F^2, \tag{10}$$

where $J(\cdot)$ denotes the objective function and $\|\cdot\|_F^2$ is the Frobenius norm. $J(\cdot)$ makes sure that the truth y_n and prediction result $\widehat{y}_n$ are remarkably close overall. However, it cannot guarantee that each i–th item $e^i = y_n^i - \widehat{y}_n^i$ is minimum. The standard deviation of e^i is incorporated into the objective function to figure out this problem. The modified objective function is as follows:

$$J\left(y_n, \widehat{y}_n\right) = \left\|y_n - \widehat{y}_n\right\|_F^2 + \eta * \operatorname{std}(e), \tag{11}$$

$$\operatorname{std}(e) = \sqrt{\frac{\sum_{i=1}^{m}\left(e^i - \overline{e}\right)}{m-1}}, \tag{12}$$

where $\overline{e}$ represents the mean value of e^i and η is a hyperparameter. The first part of the modified objective function ensures that the total loss is minimum, and the second part ensures that any prediction item is close to its truth.

4. Experiments and Analysis

There are extensive experiments with nine baseline methods and the proposed method on three benchmarked datasets for multivariate time series forecasting tasks. All of the datasets and baseline methods are available online.

4.1. Datasets and Evaluation Metrics. There are three benchmarked datasets used, which are available online. The corpus statistics are summarized in Table 1 as follows:

(1) SML2010 datasets [46]: a collection of 40 days of monitoring data from a remote intelligent monitoring system in a house (https://archive.ics.uci.edu/ml/datasets/SML2010).

(2) Gas sensor array temperature modulation datasets (GSATM) [47]: there are 14 temperature-modulated metal oxide (MOX) gas sensors obtained from a chemical detection platform. These datasets are collected for three weeks in a gas chamber (https://archive.ics.uci.edu/ml/datasets/Gas+sensor+array+temperature+modulation).

(3) NASDAQ 100 datasets [20]: the datasets consist of stock prices of 82 corporations on NASDAQ 100 ranging from July 26, 2016, to December 22, 2016. The sampling period of this dataset collection is one-minute (https://cseweb.ucsd.edu/~yaq007/NASDAQ100_stock_data.html).

Based on the chronological order of the data, two benchmarked datasets are split into training, validation, and test sets, as outlined in Table 2. It is important to note that the NASDAQ 100 datasets do not have a timestamp; therefore, it is split proportionally.

Table 1: Dataset statistics. The size of datasets for some prediction tasks is small, especially for SML2010.

Datasets	Length	# Variables	Sampling rate
SML2010	4046	19	15 minutes
GSATM	37700	19	15 minutes
NASDAQ 100	40470	82	1 minute

Table 2: Splitting strategy. The SML2010 and GSATM datasets are split into the training, validation, and test datasets, according to the chronological order. The NASDAQ 100 is divided in proportion.

Datasets	Training datasets	Validation datasets	Test datasets
SML2010	From Mar 13. 2012 To Apr 11. 2012	From Apr 18. 2012 00:00 To Apr 25. 2012 03:30	From Apr 25. 2012 03:45 To May 02. 2012 07:30
GSATM	From Sep 30. 2016 00:00 To Oct 13. 2016 23:59	From Oct 14. 2016 00:00 To Oct 14. 2016 23:59	From Oct 16. 2016 00:00 To Oct 16. 2016 23:59
NASDAQ 100	35070	2700	2700

The mean absolute error (MAE), mean absolute percent error (MAPE), root mean square error (RMSE), and empirical correlation coefficient (CORR) to measure forecast accuracy are as follows:

$$
\begin{aligned}
\mathrm{MAE} &= \frac{1}{m}\sum_{j=1}^{l}\left\|y_n^j - \hat{y}_n^j\right\|,\\
\mathrm{MAPE} &= \frac{1}{m}\sum_{j=1}^{l}\left\|\frac{y_n^j - \hat{y}_n^j}{y_n^j}\right\|,\\
\mathrm{RMSE} &= \frac{1}{m}\sum_{j=1}^{l}\sqrt{\left\|y_n^j - \hat{y}_n^j\right\|^2},\\
\mathrm{CORR} &= \frac{1}{m}\sum_{j=1}^{l}\frac{\left[y_n^j - \overline{y_n}\right]\times\left[\hat{y}_n^j - \overline{\hat{y}_n}\right]}{\sqrt{\left[y_n^j - \overline{y_n}\right]^2\times\left[\hat{y}_n^j - \overline{\hat{y}_n}\right]^2}},
\end{aligned}
\tag{13}
$$

where y_n^j and $\hat{y}_n^j$ denote the j–th true signals and its prediction at sampling time $n + h + 1$. l is the length of y_n. It is always known that CORR's higher value is better. On the contrary, for MAE, MAPE, and RMSE, the lower value is better. It should be pointed out that all errors are calculated on the original data rather than the normalized data.

4.2. Baseline Methods. The nine baseline methods for comparison are as follows:

(1) MLP is a full connection multilayer perceptron [48].

(2) LSTM is widely suited to predict time series because the algorithm is simple and effective [6]. It is available at https://pytorch.org/docs/stable/_modules/torch/nn/modules/rnn.html#LSTM.

(3) GRU is an LSTM variant algorithm with a forget gate [49]; nevertheless, it has fewer parameters than LSTM [50]. It is available at https://pytorch.org/docs/stable/generated/torch.ao.nn.quantized.dynamic.GRU.html?highlight=gru#torch.ao.nn.quantized.dynamic.GRU.

(4) ConvLSTM replaces the dense layer with a convolutional structure in the LSTM model [51]. It is available at https://github.com/rogertrullo/pytorch_convlstm.

(5) TCN uses a hierarchy of temporal convolutions to discover long-range temporal relations efficiently [52]. It is available at https://github.com/locuslab/TCN.

(6) LSTNet contains the recurrent component, recurrent-skip, and VAR-MLP component [24, 53]. It captures long- and short-term patterns. It is available at https://github.com/laiguokun/LSTNet.

(7) LSTNet Att is a typical LSTNet with the standard self-attention.

(8) Transformer is a standard transformer approach with three encoder layers and two decoder layers [16]. It is available at https://github.com/zhouhaoyi/Informer2020.

(9) Informer is reported in the best paper in AAAI 2021, which used ProbSparse self-attention to replace standard attention in the transformer [40]. It is available at https://github.com/zhouhaoyi/Informer2020.

4.3. Training Procedure. The Adam optimizer algorithm [54] is used to train models. The procedure of the training process is shown in Table 3. The core code is available at https://github.com/DezhengWang/LSTNetWithSTA.

4.4. Experimental Details. In the experiment, the structure of MLP is set as $\{m*n, m*n*10, m*n*4, m*n*2, 512, 256, m\}$, and this structure has sufficient width and depth to predict y_n. The LSTM model combines three dense layers, and the GRU model with one dense layer. The default model settings are adopted for the newest multivariate time series forecasting models ConvLSTM, TCN, LSTNet, transformer, and informer.

To verify the generalization ability of the algorithms, the sliding window is set as a fixed value rather than a hyperparameter. Therefore, the value twenty-five is selected as the

Table 3: The procedure of the proposed algorithm.

```
Require: epoch, batch size, # training iterations, learning rate, training datasets, validation datasets, test datasets, and truth
Standardize training datasets, validation datasets, and test datasets separately
for i in epoch:
  model.train ()
  model.forward (training datasets)
  loss = J (training datasets, ground truth)
  loss.backward ()
  optimizer.step ()
  if i % LogInterval == 0:
    model.eval ()
    model.forward (validation datasets)
    ValidLoss = J (validation datasets, truth)
  end
```

sliding window for those three benchmarked datasets SML2010, GSATM, and NASDAQ100.

The number of training iterations for all these nine baseline models and the proposed model on the SML2010, GSATM, and NASDAQ100 datasets are 20, 10, and 5 epochs, respectively. The learning rate of GRU is 0.0005 and that of TCN and LSTM is 0.005. Moreover, the learning rate of transformer and informer is 0.0001. The others are 0.001. Let $h = 2$, which means that the prediction horizon is set with two sampling time stamps for the forecasting over the benchmarked datasets.

4.5. Comparative Study: Why Choose LSTM-Based Models. The additional experiments on GSATM are conducted with comparative consideration to compare and analyze two commonly used methods: RNN-based and transformer-based methods. As the amount of data increases, transformer-based models have been shown to exhibit excellent performance. However, it is important to note that due to the higher model capacity of the transformer, it is more prone to overfitting when trained on insufficient samples. This is because the transformer architecture has significantly more parameters than RNN-based models and thus requires a larger dataset to avoid overfitting.

In this section, an analysis of the performance of LSTM and transformer models as benchmarks with insufficient samples is conducted. Several experiments are conducted to illustrate the abovementioned phenomenon, as shown in Figure 6. To ensure that the algorithms converged to an optimal or a closer solution, the learning rates are 0.008 and 0.0001 for LSTM and transformer, respectively, with a total number of 20 epochs. The experimental setup is aligned with the settings mentioned in Section 4.4. From Figure 6, it can be observed that the RMSE values of LSTM and transformer are quite similar when trained on a varying number of insufficient samples, ranging from 250 to 2000. However, the training time of the transformer is significantly greater than that of LSTM. Furthermore, the number of parameters of the transformer is 10,064,403, which is approximately 45 times larger than that of LSTM (which has only 219,347 parameters).

The experimental results show that LSTM with lower time and spatial complexity achieves similar performance to the transformer. It can conclude that LSTM-based methods are still the most powerful tools for tackling prediction problems when trained on insufficient samples. According to the reports [20, 55], there are many prediction tasks that require working with a small sample size. Therefore, this paper is dedicated to researching algorithm performance improvement based on LSTM architecture.

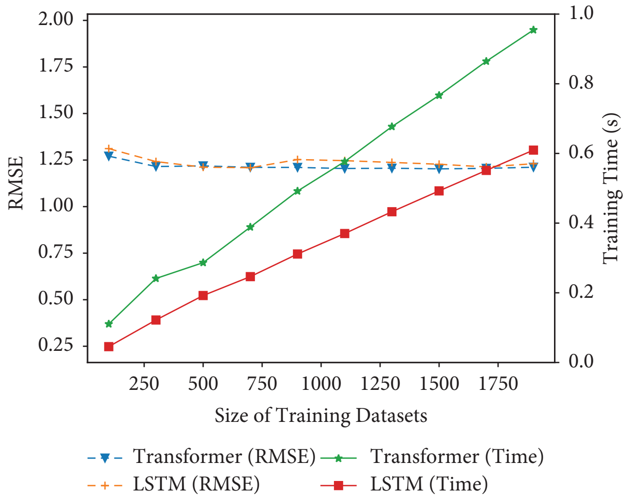

Figure 6: The trend of training time and RMSE. Transformer (RMSE) and LSTM (RMSE) denote the RMSE over the size of training datasets. Transformer (time) and LSTM (time) denote the training time over the size of training datasets. (*It should be pointed out that all errors are calculated on the original data rather than the normalized data).

4.6. Ablation Study: How Well Does Our Proposed Method Work? In this section, extensive experiments are conducted to explore the role of each component in the proposed model thoroughly. The maintenance setups are aligned with the settings mentioned in Section 4.4.

4.6.1. The Performance of the TSA Self-Attention Mechanism. Table 4 shows that the RMSE, MAE, and MAPE of the proposed method without the TSA component have increased compared to the proposed method on GSATM and NASDAQ100 datasets, except for SML2010. This is because

Table 4: Ablation study of the proposed method.

Model	SML2010				GSATM				NASDAQ100			
	MAE	MAPE	RMSE	CORR	MAE	MAPE	RMSE	CORR	MAE	MAPE	RMSE	CORR
LSTNet	0.27034	0.00894	0.90874	0.99149	4.55509	0.33331	1.24315	0.96263	0.07919	0.00117	0.82154	0.98539
OURS without TSA	0.17376	0.00580	0.87278	0.99593	2.60801	0.41757	1.23542	0.96448	0.08085	0.00120	0.82378	0.98449
OURS without SSA	0.26102	0.00861	0.90613	0.99166	3.57170	0.35723	1.24234	0.96227	0.08418	0.00125	0.82708	0.98339
OURS single loss	0.16277	0.00544	0.86847	0.99645	2.61735	0.24150	1.23765	0.96375	0.08082	0.00120	0.82415	0.98453
OURS	0.16808	0.00562	0.87153	0.99612	2.63559	0.33610	1.22660	0.96945	0.07718	0.00114	0.81949	0.98615

OURS without TSA denotes the proposed method without the TSA component. OURS without SSA denotes the proposed method without the SSA component. OURS single loss denotes that the proposed method is optimized with a standard MSE objective function rather than the proposed objective function. *It should be pointed out that all errors are calculated on the original data rather than the normalized data.

the SML2010 dataset exhibits a stronger cyclical trend of change than the other datasets, as shown in Figure 2, which can be captured by models without temporal self-attention. Nevertheless, without TSA, the proposed method cannot efficiently focus on these complicated relevant trends among data such as GSATM and NASDAQ100. Due to the SSA and the modified objective function, the proposed model still performs better.

Figure 7 presents the temporal dependencies observed in three benchmark datasets, highlighting the effectiveness of the proposed TSA component, which is a part of the spatiotemporal self-attention mechanism. The results show that the TSA component can effectively distinguish the dependencies of different historical sampling time steps on the prediction task. This enables the forecasting method to focus more on the historical moments that have significant dependencies, as indicated by their high score in the prediction results. Overall, these findings suggest that the TSA component plays a critical role in improving the forecasting accuracy of the proposed method by effectively identifying and leveraging the most relevant temporal dependencies.

4.6.2. The Performance of the SSA Self-Attention Mechanism. According to Table 4, there is a notable decline in model performance when compared to the proposed method without the SSA. This is because time series data are often interrelated, and there are complex dependencies among variables in real-world applications. Therefore, prediction models must take these relationships into account to discover useful information.

The experimental results demonstrate that the proposed SSA can capture spatial correlations among variables, as shown in Figure 7. The experiments reveal the correlations between different variables and the prediction results. It is evident that each variable has a distinct correlation with the results. As a result, the forecasting model should focus on high-scoring variables to perform better. The proposed SSA approach increases the weights of such robust dependent measurements and highly correlated variables to achieve higher prediction accuracy.

4.6.3. The Performance of the Proposed Objective Function. In Section 3.3, it is mentioned that the commonly used objective function ensures that the predicted results are close to the ground truth values on average; however, it does not guarantee that for each item error is minimized. To address this limitation, the standard deviation is incorporated into the objective function.

Including the standard deviation in the objective function ensures that the algorithm converges into a solution that is optimal or closer to optimal. Experimental results have shown that without this modification to the objective function, there is a slight increase in errors such as RMSE, MAE, and MAPE, except for MAE on GSATM, as presented in Table 4. Thus, the proposed objective function, which incorporates the standard deviation, is essential in guaranteeing that the algorithms converge to an optimal or closer solution, resulting in improved performance metrics.

4.7. Main Results. The heat map presented in Figure 7 displays the spatiotemporal dependencies observed in three benchmarked datasets. The proposed TSA effectively distinguishes the dependencies of different historical sampling time steps on the prediction tasks. Consequently, the TSA enables the forecasting method to focus on the historical moments that have significant dependencies (indicated by high scores) in the prediction results. This improved attention to relevant historical information enhances the accuracy of the forecasting method.

The proposed spatiotemporal self-attention method goes beyond capturing the temporal dependencies by also incorporating spatial correlations among variables, as illustrated in Figure 7. By examining the results of the experiments, it can discern the varying degrees of correlation between different variables and the prediction outcomes. Notably, certain variables have stronger correlations with the results than others. Consequently, a successful forecasting model must prioritize these high-scoring variables to achieve optimal performance. The proposed spatiotemporal self-attention approach achieves this by assigning greater weight to these strong dependent measurements and highly correlated variables, ultimately resulting in higher prediction accuracy.

The experimental results of all the methods (mentioned in Section 4.2) on all the test datasets (mentioned in Section 4.1) in all the evaluation metrics (13) are summarized as shown in Table 5. Clearly, the forecasting model with the proposed spatiotemporal self-attention significantly achieves superior improvement over the other baseline models on the datasets. Besides, the proposed model outperforms the baseline standard LSTNet by 0.037, 0.017, and 0.002 in

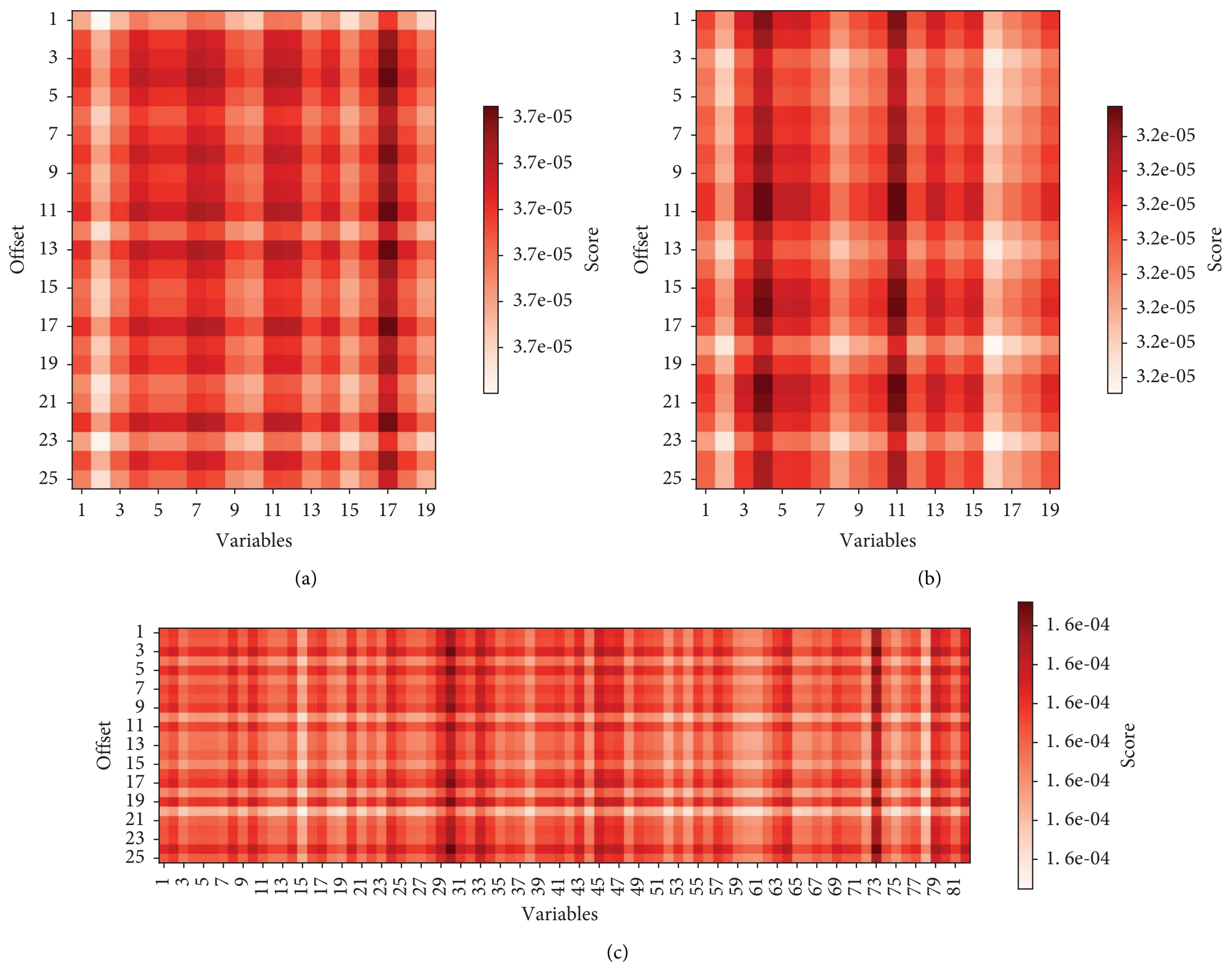

Figure 7: The spatiotemporal attention score. The different variables have varying levels of importance at different sampling times. This means that the relevance or impact of a particular variable on the prediction may change over time. (a) SML2010. (b) GSATM. (c) NASDAQ100.

Table 5: Metric results for the model and baseline models on three benchmarked datasets.

Model	SML2010				GSATM				NASDAQ100			
	MAE	MAPE	RMSE	CORR	MAE	MAPE	RMSE	CORR	MAE	MAPE	RMSE	CORR
MLP	0.63838	0.02164	0.98231	0.95989	20.59712	0.46255	1.37513	0.92804	0.54500	0.00806	0.97785	0.22437
LSTM	0.63722	0.02164	0.98598	0.95516	4.14283	**0.26364**	1.24097	0.96301	0.59471	0.00880	0.97173	0.59591
GRU	0.50851	0.01697	0.96513	0.97588	15.77770	0.38983	1.34624	0.94839	0.33149	0.00492	0.93270	0.85718
ConvLSTM	0.46332	0.01567	0.95389	0.97957	5.56830	0.32900	1.25887	0.96892	0.38968	0.00578	0.93844	0.89860
TCN	0.25622	0.00902	0.90435	0.99384	3.86711	0.32546	1.25404	0.94602	0.08242	0.00122	0.82505	0.98434
LSTNet	0.27034	0.00894	0.90874	0.99149	4.55509	0.33331	1.24315	0.96263	0.07919	0.00117	0.82154	0.98539
LSTNet Att	0.26516	0.00875	0.90719	0.99182	4.71850	0.32771	1.25272	0.95450	0.08370	0.00124	0.82684	0.98349
Transformer	0.46210	0.01539	0.95635	0.98094	3.16311	0.47925	1.23539	0.96451	0.38075	0.00564	0.93654	0.79833
Informer	0.37068	0.01228	0.94434	0.98514	3.07807	0.49539	1.23384	0.96591	0.40855	0.00605	0.94117	0.76445
OURS	**0.16808**	**0.00562**	**0.87153**	**0.99612**	**2.63559**	0.33610	**1.22660**	**0.96945**	**0.07718**	**0.00114**	**0.81949**	**0.98615**

LSTNet ATT denotes LSTNet with the standard self-attention. Bold face indicates the best result of each column in a particular metric. *It should be noted that all errors are calculated on the original data rather than the normalized data.

the RMSE metric on SML2010, GSATM, and NASDAQ100 datasets. With the integration of the spatiotemporal self-attention and a standard deviation term (12), the LSTNet with the proposed spatiotemporal self-attention achieves the best MAE, RMSE, and CORR across three datasets. This is because it not only tries to obtain the temporal dependencies but also employs a spatial self-attention to capture the correlations across other variables.

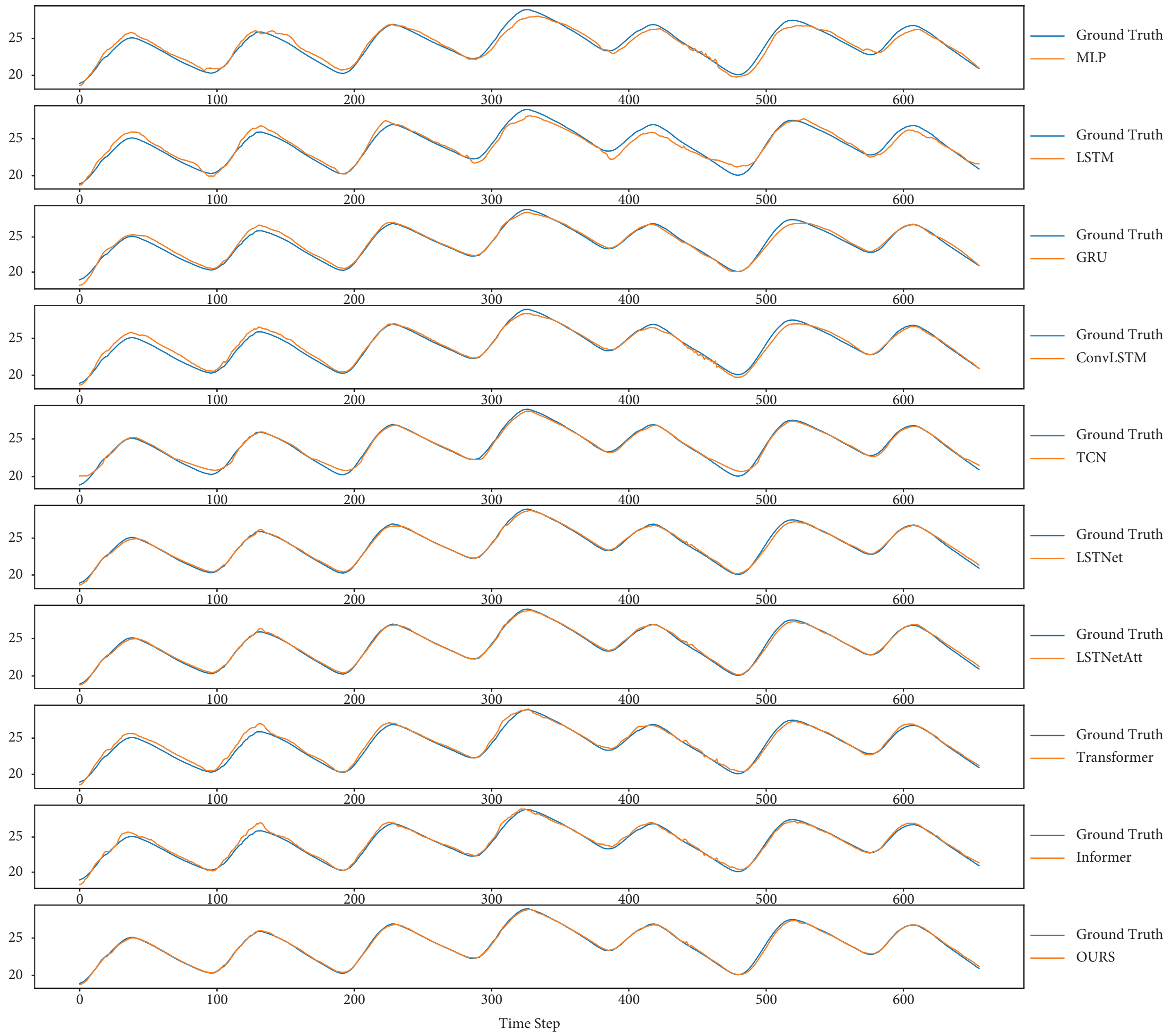

Figure 8: Forecast results of variable 1 for the models on SML2010.

In addition, the RMSE of state-of-the-art ConvLSTM, TCN, LSTNet, transformer, and informer performs better than MLP, LSTM, and GRU approaches. This result is obtained because these five state-of-the-art models have an effective feature extraction capability. Therefore, these state-of-the-art models can obtain richer information than others. Nevertheless, such state-of-the-art models ignore that different variables have different spatial correlations on the forecast results. It should be pointed out that LSTNet with the standard self-attention (LSTNet Att) has an insignificant effect on model performance. It is on account of the standard LSTNet model that considers short-term and long-term patterns. Namely, it can obtain complex temporal dependencies. Therefore, the common self-attention is hardly helpful regarding LSTNet model performance.

Moreover, the transformer and informer algorithms have a higher model capacity compared to the proposed algorithm, which can make them more prone to overfitting when trained on insufficient samples. As a result, the transformer and informer algorithms have higher RMSE of 0.95635 and 0.94434 on the SML2010, 1.23539 and 1.23384 on the GSATM, and 0.93654 and 0.94117 on the NASDAQ100, respectively. In contrast, the proposed algorithm has an RMSE of 0.87153, 1.22660, and 0.81949 on the SML2010, GSATM, and NASDAQ100, respectively, which is lower than the RMSE of the transformer and informer algorithms.

Figures 8 and 9 show that any prediction of variables fits the truth well. It is due to the standard deviation term in the objective function described in Section 3.3. The modified objective function ensures the minimum total loss, and any prediction is close to its truth.

Figure 10 denotes that MLP, LSTM, GRU, and ConvLSTM approaches cannot handle complex datasets such as NASDAQ100. This result is because those basic methods cannot distinguish the short-term and long-term patterns. MLP, LSTM, and GRU network structures do not even obtain spatial features. Moreover, Figure 11 indicates that the proposed approach tracks the trend of the original data well, even with extreme values.

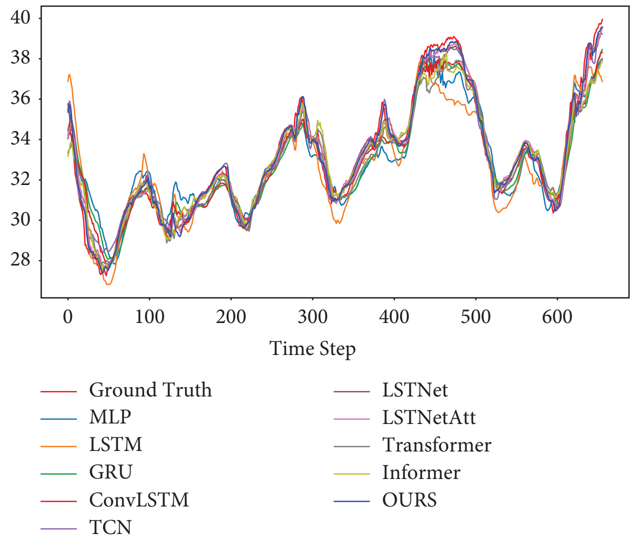

Figure 9: Forecast results of variable 7 for the models on SML2010.

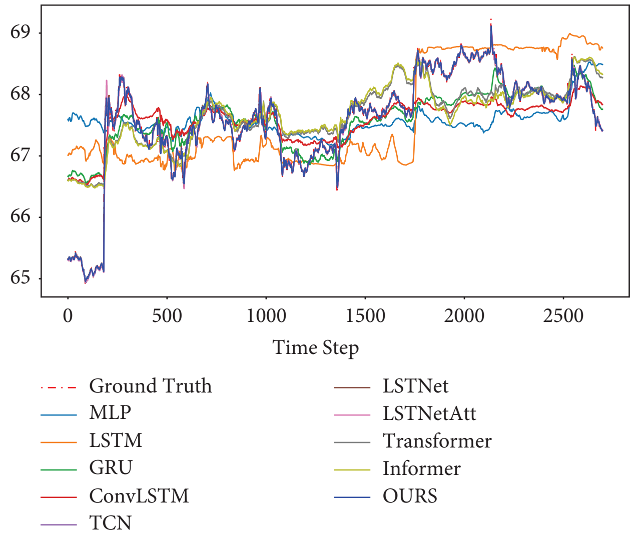

Figure 10: Forecast results of variable 7 for the models on Nasdaq100.

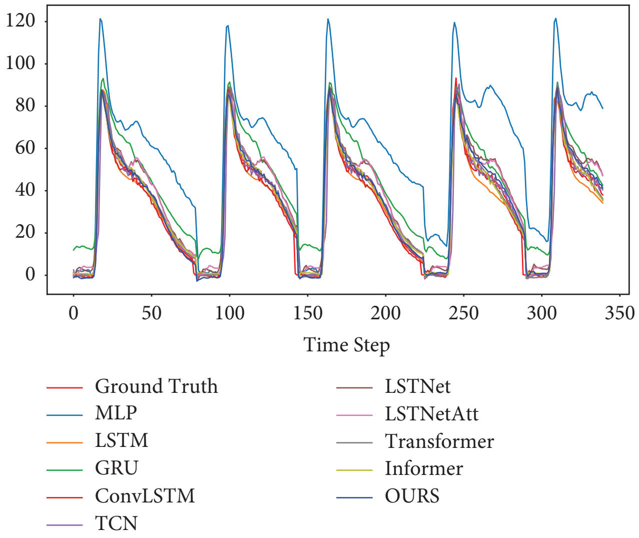

Figure 11: Forecast results of variable 11 for the models on GSATM.

Above all, the proposed approach accurately obtains the spatiotemporal patterns between its historical measurement and the variables. It significantly improves the state-of-the-art results in multivariate time series prediction on three benchmarked datasets.

5. Conclusion

This paper presents a novel approach for multivariate time series forecasting based on spatiotemporal self-attention. The proposed approach utilizes a spatial and temporal self-attention mechanism, a standard LSTNet, and a modified objective function. The spatial self-attention is able to capture correlations among variables, while the temporal self-attention enhances the temporal feature extraction capabilities of LSTNet. The modified objective function incorporates the standard deviation of each loss item, ensuring that predictions for each variable fit the ground truth accurately. The comparative study indicates that LSTM methods remain the most potent tool for tackling prediction problems, achieving similar performance with transformer when trained on insufficient samples. Meanwhile, the ablation study shows that the proposed method without SSA has a notable decline in the performance. In addition, the proposed objective function incorporating the standard deviation is helpful for converging into an optimal or closer solution. The approach yields significant improvements in MAE, MAPE, and RMSE by an average of 2.62, 0.13, and 7%, respectively, compared to other comparison algorithms. Moreover, CORR has increased by an average of 10%. The abovementioned experiments on the benchmarked datasets demonstrate that the proposed approach consistently enhances state-of-the-art methods for multivariate time series forecasting tasks.

Data Availability

(1) The SML2010 datasets supporting this work are from previously reported studies and datasets, which have been cited. The processed data are available at https://archive.ics.uci.edu/ml/datasets/SML2010. (2) The Gas Sensor Array Temperature Modulation datasets supporting this work are from previously reported studies and datasets, which have been cited. The processed data are available at https://archive.ics.uci.edu/ml/datasets/Gas+sensor+array+temperature+modulation. (3) The NASDAQ 100 datasets supporting this work are from previously reported studies and datasets, which have been cited. The processed data are available at https://cseweb.ucsd.edu/~yaq007/NASDAQ100_stock_data.html.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Science and Technology Research Program of Chongqing Municipal Education Commission (grant nos. KJZD-K202201901, KJQN202201109, and KJQN202101904). Moreover, this work was also supported by the Scientific Research Foundation of Chongqing Institute of Engineering (grant no. 2020xzky05) and Natural Science Foundation of Chongqing (grant no. cstc2020jcyj-msxmX0666). Additionally, this work was supported by Shenzhen Fundamental Research Program (grant no. JCYJ20210324135006016) and Guangdong Basic and Applied Basic Research Foundation (grant no. 2022A1515010345).

References

[1] B. M. Henrique, V. A. Sobreiro, and H. Kimura, "Literature review: machine learning techniques applied to financial market prediction," *Expert Systems with Applications*, vol. 124, pp. 226–251, 2019.

[2] M. Benhaddi and J. Ouarzazi, "Multivariate time series forecasting with dilated residual convolutional neural networks for urban air quality prediction," *Arabian Journal for Science and Engineering*, vol. 46, no. 4, pp. 3423–3442, 2021.

[3] P. S. Nyaki, H. Bwire, and N. K. Mushule, "Comparative assessment of dynamic travel time prediction models in the developing countries cities," *International Journal of Traffic and Transportation Engineering*, vol. 10, no. 1, pp. 96–110, 2020.

[4] Z. Liu and M. Hauskrecht, "A regularized linear dynamical system framework for multivariate time series analysis," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, pp. 1798–1804, 2015.

[5] P. S. Desai, "News sentiment informed time-series analyzing AI (SITALA) to curb the spread of COVID-19 in houston," *Expert Systems with Applications*, vol. 180, Article ID 115104, 2021.

[6] V. K. R. Chimmula and L. Zhang, "Time series forecasting of COVID-19 transmission in Canada using LSTM networks," *Chaos, Solitons & Fractals*, vol. 135, Article ID 109864, 2020.

[7] J. W. Goodell, R. J. McGee, and F. McGroarty, "Election uncertainty, economic policy uncertainty and financial market uncertainty: a prediction market analysis," *Journal of Banking & Finance*, vol. 110, Article ID 105684, 2020.

[8] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: a graph multi-attention network for traffic prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 1, pp. 1234–1241, 2020.

[9] J. Q. Wang, Y. Du, and J. Wang, "LSTM based long-term energy consumption prediction with periodicity," *Energy*, vol. 197, Article ID 117197, 2020.

[10] Y. Liu, H. Wu, K. Rezaee et al., "Interaction-enhanced and time-aware graph convolutional network for successive point-of-interest recommendation in traveling enterprises," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 635–643, 2023.

[11] L. Qi, Y. Liu, Y. Zhang, X. Xu, M. Bilal, and H. Song, "Privacy-aware point-of-interest category recommendation in internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21398–21408, 2022.

[12] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," 2014, https://arxiv.org/abs/1409.2329.

[13] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," *Esann*, vol. 89, pp. 89–94, 2015.

[14] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, https://arxiv.org/abs/1412.3555.

[15] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 156–165, Piscataway, NJ, USA, July 2017.

[16] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[17] Y. Xiao, H. Yin, Y. Zhang, H. Qi, Y. Zhang, and Z. Liu, "A dual-stage attention-based Conv-LSTM network for spatio-temporal correlation and multivariate time series prediction," *International Journal of Intelligent Systems*, vol. 36, no. 5, pp. 2036–2057, 2021.

[18] B. Lim and S. Zohren, "Time-series forecasting with deep learning: a survey," *Philosophical Transactions of the Royal Society A: Mathematical, Physical & Engineering Sciences*, vol. 379, Article ID 20200209, 2021.

[19] Y. Li, Z. Zhu, D. Kong, H. Han, and Y. Zhao, "EA-LSTM: e," *Knowledge-Based Systems*, vol. 181, Article ID 104785, 2019.

[20] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," 2017, https://arxiv.org/abs/1704.02971.

[21] X. Yin, Y. Han, H. Sun, Z. Xu, H. Yu, and X. Duan, "Multi-attention generative adversarial network for multivariate time series prediction," *IEEE Access*, vol. 9, pp. 57351–57363, 2021.

[22] A. Galassi, M. Lippi, and P. Torroni, "Attention in natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 10, pp. 4291–4308, 2021.

[23] E. Fu, Y. Zhang, F. Yang, and S. Wang, "Temporal self-attention-based Conv-LSTM network for multivariate time series prediction," *Neurocomputing*, vol. 501, pp. 162–173, 2022.

[24] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," in *Proceedings of the 41st international ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 95–104, Ann Arbor, MI, USA, June 2018.

[25] S. Panigrahi and H. S. Behera, "A hybrid ETS–ANN model for time series forecasting," *Engineering Applications of Artificial Intelligence*, vol. 66, pp. 49–59, 2017.

[26] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: a systematic literature review: 2005–2019," *Applied Soft Computing*, vol. 90, Article ID 106181, 2020.

[27] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[28] A. Sagheer and M. Kotb, "Time series forecasting of petroleum production using deep LSTM recurrent networks," *Neurocomputing*, vol. 323, pp. 203–213, 2019.

[29] Y. Liu, D. Li, S. Wan et al., "A long short-term memory-based model for greenhouse climate prediction," *International Journal of Intelligent Systems*, vol. 37, no. 1, pp. 135–151, 2022.

[30] H. Nguyen, K. P. Tran, S. Thomassey, and M. Hamad, "Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management," *International Journal of Information Management*, vol. 57, Article ID 102282, 2021.

[31] X. Jin, W. Zheng, J. Kong et al., "Deep-learning forecasting method for electric power load via attention-based encoder-decoder with bayesian optimization," *Energies*, vol. 14, no. 6, p. 1596, 2021.

[32] J. Wang, X. Sun, Q. Cheng, and Q. Cui, "An innovative random forest-based nonlinear ensemble paradigm of improved feature extraction and deep learning for carbon price forecasting," *Science of the Total Environment*, vol. 762, Article ID 143099, 2021.

[33] D. Ponnoprat, "Short-term daily precipitation forecasting with seasonally-integrated autoencoder," *Applied Soft Computing*, vol. 102, Article ID 107083, 2021.

[34] X. Hou, K. Wang, C. Zhong, and Z. Wei, "St-trader: a spatial-temporal deep neural network for modeling stock market movement," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 5, pp. 1015–1024, 2021.

[35] I. E. Livieris, E. Pintelas, and P. Pintelas, "A CNN–LSTM model for gold price time-series forecasting," *Neural Computing & Applications*, vol. 32, no. 23, pp. 17351–17360, 2020.

[36] P. Lara-Benítez, M. Carranza-García, J. M. Luna-Romera, and J. C. Riquelme, "Temporal convolutional networks applied to energy-related time series forecasting," *Applied Sciences*, vol. 10, no. 7, p. 2322, 2020.

[37] Y. He and J. Zhao, "Temporal convolutional networks for anomaly detection in time series," *Journal of Physics: Conference Series*, vol. 1213, no. 4, Article ID 042050, 2019.

[38] D. Li, F. Jiang, M. Chen, and T. Qian, "Multi-step-ahead wind speed forecasting based on a hybrid decomposition method and temporal convolutional networks," *Energy*, vol. 238, Article ID 121981, 2022.

[39] R. Assaf, I. Giurgiu, F. Bagehorn, and A. Schumann, "Mtex-cnn: multivariate time series explanations for predictions with convolutional neural networks," in *Proceedings of the 2019 IEEE International Conference on Data Mining (ICDM)*, pp. 952–957, IEEE, Beijing, China, November 2019.

[40] H. Zhou, S. Zhang, J. Peng et al., "Informer: beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, pp. 11106–11115, Beijing, China, May 2021.

[41] L. Shen and Y. Wang, "TCCT: tightly-coupled convolutional transformer on time series forecasting," *Neurocomputing*, vol. 480, pp. 131–145, 2022.

[42] R. Lam, A. Sanchez-Gonzalez, M. Willson et al., "GraphCast: learning skillful medium-range global weather forecasting," 2022, https://arxiv.org/abs/2212.12794.

[43] X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W.-K. Wong, and W. C. Woo, "Convolutional LSTM network: a machine learning approach for precipitation nowcasting," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[44] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: the long-document transformer," 2020, https://arxiv.org/abs/2004.05150.

[45] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, https://arxiv.org/abs/1409.0473.

[46] F. Zamora-Martinez, P. Romeu, P. Botella-Rocamora, and J. Pardo, "On-line learning of indoor temperature forecasting models towards energy efficiency," *Energy and Buildings*, vol. 83, pp. 162–172, 2014.

[47] J. Burgués and S. Marco, "Multivariate estimation of the limit of detection by orthogonal partial least squares in temperature-modulated MOX sensors," *Analytica Chimica Acta*, vol. 1019, pp. 49–64, 2018.

[48] P. H. Borghi, O. Zakordonets, and J. P. Teixeira, "A COVID-19 time series forecasting model based on MLP ANN," *Procedia Computer Science*, vol. 181, pp. 940–947, 2021.

[49] P. T. Yamak, L. Yujian, and P. K. Gadosey, "A comparison between arima, lstm, and gru for time series forecasting," in *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, pp. 49–55, Sanya, China, December 2019.

[50] X. Li, X. Ma, F. Xiao, C. Xiao, F. Wang, and S. Zhang, "Time-series production forecasting method based on the integration of bidirectional gated recurrent unit (Bi-GRU) network and sparrow search algorithm (SSA)," *Journal of Petroleum Science and Engineering*, vol. 208, Article ID 109309, 2022.

[51] S. W. Lee and H. Y. Kim, "Stock market forecasting with super-high dimensional time-series data using ConvLSTM, trend sampling, and specialized data augmentation," *Expert Systems with Applications*, vol. 161, Article ID 113704, 2020.

[52] P. Hewage, A. Behera, M. Trovati et al., "Temporal convolutional neural (TCN) network for an effective weather forecasting using time-series data from the local weather station," *Soft Computing*, vol. 24, no. 21, pp. 16453–16482, 2020.

[53] H. Sano and J. Rokui, "Multivariate time series forecasting accuracy improvement method based on LSTNet," *IEICE Technical Report; IEICE Tech. Rep*, vol. 121, no. 304, pp. 71–76, 2021.

[54] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014.

[55] J. Geng, C. Yang, Y. Li, L. Lan, and Q. Luo, "MPA-RNN: a novel attention-based recurrent neural networks for total nitrogen prediction," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 10, pp. 6516–6525, 2022.