energies

MDPI

Article

TS2ARCformer: A Multi-Dimensional Time Series Forecasting Framework for Short-Term Load Prediction

Songjiang Li [1], Wenxin Zhang [1] and Peng Wang [1,2,*]

[1] College of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China; lsj@cust.edu.cn (S.L.); zwx@mails.cust.edu.cn (W.Z.)
[2] Changchun University of Science and Technology Chongqing Research Institute, Chongqing 401120, China
* Correspondence: wangpeng@cust.edu.cn

Abstract: Accurately predicting power load is a pressing concern that requires immediate attention. Short-term load prediction plays a crucial role in ensuring the secure operation and analysis of power systems. However, existing research studies have limited capability in extracting the mutual relationships of multivariate features in multivariate time series data. To address these limitations, we propose a multi-dimensional time series forecasting framework called TS2ARCformer. The TS2ARCformer framework incorporates the TS2Vec layer for contextual encoding and utilizes the Transformer model for prediction. This combination effectively captures the multi-dimensional features of the data. Additionally, TS2ARCformer introduces a Cross-Dimensional-Self-Attention module, which leverages interactions across channels and temporal dimensions to enhance the extraction of multivariate features. Furthermore, TS2ARCformer leverage a traditional autoregressive component to overcome the issue of deep learning models being insensitive to input scale. This also enhances the model's ability to extract linear features. Experimental results on two publicly available power load datasets demonstrate significant improvements in prediction accuracy compared to baseline models, with reductions of 43.2% and 37.8% in the aspect of mean absolute percentage error (MAPE) for dataset area1 and area2, respectively. These findings have important implications for the accurate prediction of power load and the optimization of power system operation and analysis.

Keywords: multivariate time series; load prediction; TS2Vec; transformer; attention

Citation: Li, S.; Zhang, W.; Wang, P. TS2ARCformer: A Multi-Dimensional Time Series Forecasting Framework for Short-Term Load Prediction. *Energies* **2023**, *16*, 5825. https://doi.org/10.3390/en16155825

Academic Editors: Antonio Gabaldón, María Carmen Ruiz-Abellón and Luis Alfredo Fernández-Jiménez

Received: 5 July 2023
Revised: 31 July 2023
Accepted: 3 August 2023
Published: 5 August 2023

1. Introduction

Electricity is an indispensable component of our daily lives, playing a vital role in powering various aspects of modern lifestyles. Ensuring the stable and reliable operation of power systems is a key objective for electric power companies. To achieve this, a dynamic balance between electricity supply and demand must be maintained, efficiently meeting the energy needs of consumers without interruption. Accurate load forecasting is fundamental to achieving this dynamic balance and holds significant practical value for power companies. It enables cost reduction, improved efficiency, and contributes to the realization of "dual-carbon" goals in the power system transformation. Short-term load forecasting (STLF) has been widely applied in recent years as a time series forecasting problem [1,2]. This article focuses primarily on short-term load forecasting, aiming to construct a multidimensional time series model using historical load data and relevant influencing factors to predict the load for the next day.

Recently, deep learning-based methods have gained popularity in power load forecasting due to the development of deep learning techniques and the availability of abundant data. These methods leverage neural networks and other deep learning models to capture complex load patterns and correlations with relevant factors, aiming to improve prediction accuracy. For example, Kong et al. [3] used LSTM for short-term load prediction, but it struggles with long input sequences, diluting historical information and losing sequence

details. Future information, such as weather data, is also overlooked. Lu et al. [4] explored relationships in data using GRU and proposed a multi-energy coupling short-term load forecasting model. Liu et al. [5] used LSTNet to predict short-term electricity load. This neural network is adept at capturing the long-term relationships among multiple variables and extracting both highly nonlinear long-term and short-term characteristics, as well as linear characteristics, from the data. Zhang et al. [6] combined AR's interpretability with LSTM's predictive capability, successfully applying it to forecast COVID-19 cases with promising outcomes. Bai et al. [7] introduced TCN, which incorporates convolutional layers to handle sequential data, achieving better performance on certain tasks. Guo et al. [8] proposed a hybrid model that combines CEEMDAN and TCN with adaptive noise for time series prediction. To account for external factors such as price, weather, and calendar, studies have explored incorporating this information into short-term load forecasting (STLF) models [9]. However, limited research has been conducted on analyzing the dimensional relationships between electrical loads and exogenous data as a multivariate time series. Improving the feature analysis can enhance the accuracy of deep learning-based STLF models. Multiple time series prediction tasks have also been explored, such as the combination of Transformer and other models for traffic graph prediction [10] and the use of deep learning methods to predict highway passenger volume [11]. Kim et al. [1] proposed a novel approach for extracting features from multivariate time series data, including electrical load and related data. Their framework consists of two processes: tagging and embedding. These processes identify patterns within the data and capture their temporal and dimensional relationships. Thorough experimentation demonstrated impressive performance in short-term electrical load forecasting. However, these methods face challenges and room for improvement in encoding and modeling multiple features. Traditional deep learning methods often have low encoding efficiency and overlook the interrelationships between different features. Moreover, deep learning models are insensitive to input scale, limiting their adaptability and accuracy for the periodic variations in power load data.

These research methods primarily focus on long- and short-term forecasting tasks in the time domain, neglecting the interrelationships among multidimensional features in power load data. This results in inefficient encoding of multidimensional data using traditional deep learning methods. Analysis of electricity load data has revealed a clear correlation between meteorological data and electricity load forecasting (as shown in Figure 1). Building on the findings of Hernández et al. [12], their discovery of the relationship between meteorological variables and electric power demand through experiments underscores the importance of taking this correlation into account when conducting electricity load forecasting within the context of a smart grid. However, existing models such as LSTM only consider temporal dependencies and fail to fully capture the interrelationships among multidimensional features, limiting their accurate modeling of multidimensional information. Additionally, deep learning models struggle to adapt to the varying periodicity of power load data due to their fixed input scale. Power load data exhibit different cyclic patterns, such as seasonal variations or holidays, and traditional deep learning models lack the flexibility to adapt to such changes, leading to inaccurate predictions.

To address these challenges, this paper proposes a novel framework for short-term load prediction named TS2ARCformer, comprising TS2Vec, Transformer, and AR components. TS2ARCformer leverages the TS2Vec layer [13] to embed the original time series data, transforming it into higher-dimensional feature vectors. These feature vectors capture more abstract information, enabling a better representation of long-term dependencies and periodic variations in the time series data. For prediction tasks, the encoded data are fed into the Transformer model, where a Cross-Dimensional-Self-Attention mechanism is introduced to enhance the utilization of inter-task correlations. The Cross-Dimensional-Self-Attention considers both the internal dependencies within the electricity load sequence and the dependencies with other relevant tasks, effectively extracting multidimensional feature information from the data. Additionally, an autoregressive (AR) component is incorporated to independently forecast the electricity load, thereby improving the model's

short-term prediction capability. Compared to conventional methods, the TS2ARCformer demonstrates significant performance advantages. To sum up, the contributions of this paper are shown as follows:

- Efficient Multi-Dimensional Encoding. We incorporate the TS2Vec layer into the multi-dimensional time series forecasting task to improve the encoding efficiency of diverse features;
- Enhanced Interdependency Learning. By introducing a Cross-Dimensional-Self-Attention mechanism to the Transformer model, we enable better exploration of interdependencies among multi-dimensional features, enhancing the model's learning capabilities;
- AR Integration for Scale Sensitivity. To address the insensitivity of traditional deep learning models to input scale, we integrate an autoregressive (AR) component into our framework, enhancing the model's ability to extract linear features and adapt to varying input scales;
- Comprehensive analysis and validation of TS2ARCformer.The proposed TS2ARCformer model's predictive performance is thoroughly analyzed and validated, providing insights into its effectiveness for power load forecasting.

The remainder of this paper is structured as follows: Section 2 provides a description of the related work. The methodology of this is presented in Section 3. Section 4 introduces the dataset used for the case study and analyzes and compares the results. Lastly, Section 5 provides a summary of the paper.

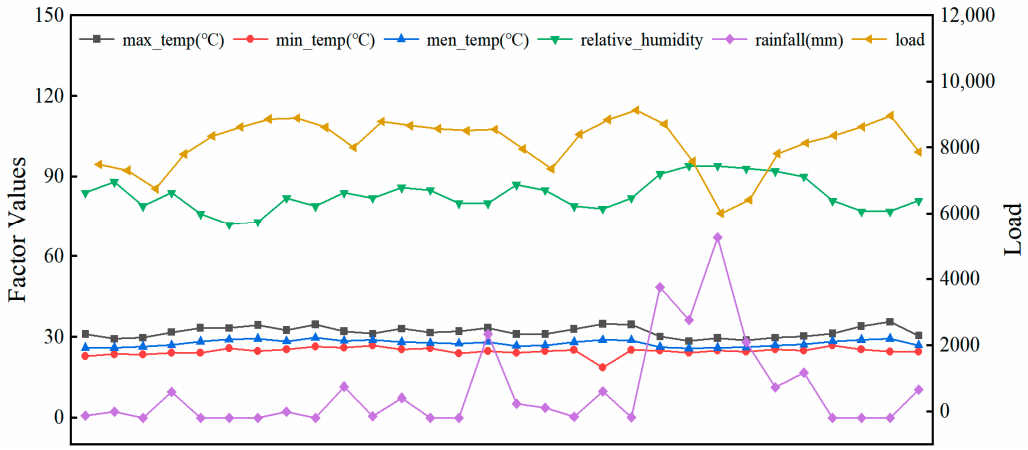

Figure 1. Impact of Meteorological Data on Load.

2. Related Work

Currently, both domestic and international research on electricity load forecasting methods can be roughly categorized into three types: (1) traditional methods, (2) machine learning methods, and (3) deep learning methods. Traditional electricity load forecasting methods include multivariate linear regression [14], Kalman filtering [15], exponential smoothing models [16], etc. These methods utilize historical load data to predict future loads and consider the temporal nature of the data. However, they have limited capability in handling nonlinearity. Machine learning methods encompass techniques such as random forests [17], support vector machines [18], and artificial neural networks [19]. By incorporating machine learning algorithms, these methods address the nonlinear relationships among data effectively. However, they still have limitations in fully utilizing the temporal information in time series data. In recent years, deep learning-based methods have been widely applied in short-term load forecasting [20]. Commonly used deep learning models such as RNN [21], LSTM [22], and GRU [23] have been widely adopted. However, when dealing with long time series data, these models may suffer from issues such as exploding or vanishing gradients, insufficient exploitation of nonlinear relationships among sequential data, and difficulty in capturing long-term dependencies between sequences.

Moreover, these models often require sequential data input, leading to low training efficiency. Therefore, there is a need to explore more innovative and efficient deep learning models to address these challenges. Dong et al. [24] proposed a short-term load forecasting method that combines k-means clustering and CNN to accommodate large-scale power load data. The high-order features extracted by CNN were found to effectively improve the accuracy of load forecasting. Park et al. [25] proposed a load forecasting method based on the Long Short-Term Memory (LSTM) neural network, utilizing load feature decomposition techniques to predict the load of the previous day. Rafi et al. [26] introduced a combined approach using Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks for short-term load forecasting. This network performed well in short-term load forecasting tasks but had limitations in handling inputs and outputs of different lengths. While LSTM can handle long and short-term dependencies to some extent, issues such as dilution of historical information and loss of sequential information still persist when the input sequence is too long. To address this problem, a novel sequence-to-sequence (Seq2Seq) structure was first applied to load forecasting tasks by Gong et al. [27]. Wu et al. [28] proposed a hybrid neural network model, GRU-CNN, which combines the GRU model with the CNN model.

The Transformer model [29], as a novel deep learning model, has gradually been applied in various fields such as speech recognition, image recognition, and machine translation. Recently, research has shown that the Transformer model has better potential in capturing long-term dependencies [30]. Some scholars have attempted to apply the Transformer model to time series forecasting and achieved promising results [31]. Guo et al. [32] constructed an attention-based spatiotemporal graph network for traffic flow prediction, where the attention mechanism is implemented using the Transformer model. L'Heureux et al. [33] proposed a Transformer-based load forecasting architecture by modifying the NLP Transformer workflow, introducing n-space transformations, and designing a new technique for handling contextual features. Zhao et al. [34] proposed a novel model based on the Transformer network to provide accurate load forecasting for the previous day. The model includes a similar day selection method involving LightGBM and k-means algorithms. Compared to traditional RNN-based models, the proposed model can avoid falling into local minima and outperform global search. Koohfar et al. [35] employed the Transformer model to predict electric vehicle charging demand for short-term and long-term forecasting of electric vehicle charging load. The performance of the model was evaluated using RMSE and MAE. The results demonstrated that the Transformer model outperformed other models in both short-term and long-term forecasting, showcasing its ability to address time series problems, particularly in electric vehicle charging prediction. Li et al. [36] proposed a novel hybrid neural network, FDG-Transformer, which combines the GRU, LSTM, and multi-head attention (MHA) Transformer. The integrated Transformer network can encode the varying weights of the influence from each past time step to the current time step, thus establishing a time series model at a deeper granularity level. Wang et al. [29] developed a multi-task model, MultiDeT (Multi-Decoder Transformer), which employs a single encoder-multiple decoder structure to achieve a multi-task architecture and jointly predicts multi-energy loads. Ran et al. [37] proposed a hybrid model that combines complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN), sample entropy (SE), and Transformer.

In summary, compared to LSTM and GRU, Transformer model can better handle time series relationships, capturing long-term dependencies, and uncovering latent features. However, traditional Transformers overlook correlations between data dimensions, limiting their use with multivariate data and complex relationships. To address these issues, we propose a Cross-Dimensional-Self-Attention mechanism to enhance feature extraction and improve anomaly handling in the Transformer model.

3. Materials and Methods

3.1. Preliminary

We represent the multi-dimensional time series prediction task as a function approximation problem. Given historical observed data $X_T = \{y_1, y_2, \ldots, y_T\} \in R^{T \times S}$, where each column $y_t \in R^S$ represents the values of S-dimensional variables at different time steps, our goal is to predict the future signal sequence $Y' = \{y_{T+1}, y_{T+2}, \ldots, y_{T+h}\} \in R^{h \times 1}$ by learning a function f. Here, h represents the desired prediction time horizon, and the predicted Y' corresponds to the one-dimensional electricity load value sequence that we need.

We represent the function f as a mapping relationship: $Y' = f(X_T)$, where f is a function that maps the input matrix X_T to the output sequence Y'. The objective of this function is to capture patterns and dependencies in the historical observed data and apply them to future predictions.

In the modeling process, we can select various deep learning models such as RNNs, LSTMs, CNNs, or Transformers to capture features and patterns from historical data. These models enable us to forecast future time steps of the signal by learning from past observations.

3.2. Overview

The overview of TS2ARCformer is depicted in Figure 2, offers several advancements over current methods for load forecasting. By utilizing the TS2Vec layer, TS2ARCformer effectively captures temporal features and maps them to a high-dimensional space. Furthermore, it combines the predictions of the autoregressive (AR) component and the enhanced Transformer model, harnessing their individual strengths. The AR component enhances the model's ability to capture temporal features, dependencies, and contextual information. Meanwhile, the Cross-Dimensional-Self-Attention module employed by the enhanced Transformer model enables a comprehensive consideration of relevant information in time series data, resulting in more accurate load forecasting This integration of the Cross-Dimensional-Self-Attention module enhances the Transformer model's expressive power and generalization ability for the task of electricity load prediction.

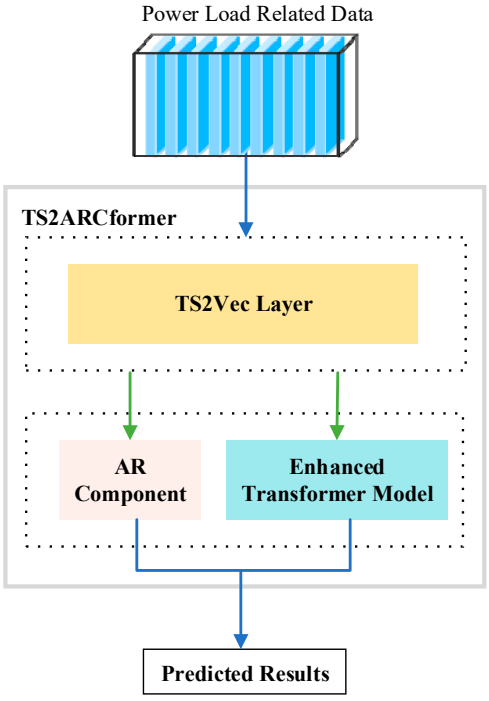

Figure 2. The flowchart of TS2ARCformer.

3.3. Framework

In this section, we will provide detailed information about each module involved in the model.

3.3.1. TS2Vec Layer

TS2Vec layer is a neural network-based method that generates embeddings for time series data, transforming time features into a high-dimensional space. Similar to word embedding layers in NLP, TS2Vec layer provides a stable representation of timestamps through contrastive learning, improving performance. The universal framework of TS2Vec learns time series representations by comparing sequences to identify hierarchical features and comparing timestamps within sequences to identify temporal features. The essence of sequence learning is maximizing the utilization of historical data. Let us assume we have N sets of time series $\{X_1, X_2, ..., X_k\}$ as input, where each set $X_i = \{y_1, y_2, \ldots, y_T\} \in R^{T \times S}$. After using the TS2Vec layer, the output will consist of N sets of representation vectors $\{R_1, R_2, ..., R_k\}$. Each vector's feature dimension is denoted as F, indicating that the dimension of a set of representation vectors is $F \times T$. Thus, each set of representation vectors $R_i = \{r_1, r_2, \ldots, r_T\} \in R^{T \times F}$. The network model f_θ of TS2Vec consists of three parts: an input mapping layer, a timestamp masking layer, and an expanded convolution module; that is, $R_i = f_\theta(X_i)$. The TS2Vec layer incorporates various modules to capture temporal information and enhance data feature learning. It leverages dilated convolutional layers for robust feature extraction and employs temporal contrastive loss and instance-wise contrastive loss for comprehensive learning. These contrastive learning techniques enable the model to capture specific load data features and dynamic trends over time, facilitating information expression at multiple scales. By effectively capturing temporal features, the TS2Vec layer is well-suited to handle random, complex nonlinear, and multiscale changes in power load-related time series. Leveraging these advantages, we integrate the TS2Vec layer into our hybrid deep learning model. This integration allows for improved extraction of temporal features and simplifies data processing, contributing to more accurate load forecasting. The basic structure of TS2Vec layer is shown in Figure 3. The TS2Vec layer is capable of handling multivariate time series data as input. It encodes the multivariate data into multidimensional feature vectors using its encoder. These encoded feature vectors are then passed to the Hierarchical Contrasting component for contrastive learning. This process enables the model to capture and represent complex patterns and relationships present in the multivariate time series data.

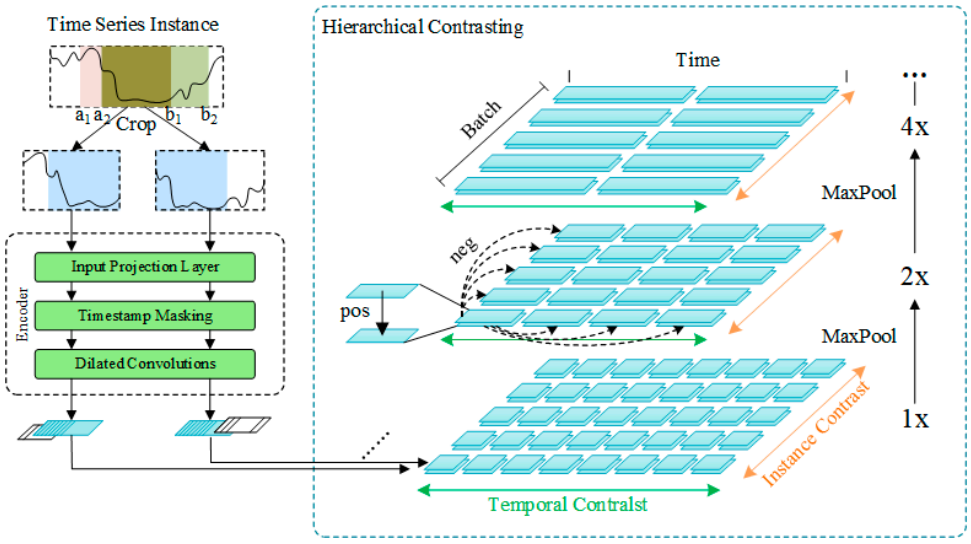

Figure 3. The structure of the TS2Vec Layer.

For X_i, randomly select its two subsequences with overlapping parts. It is expected to obtain consistent context expression from overlapping features. Let i be the index of the input time series sample and t be the timestamp. Then $r_{i,t}$ and $r'_{i,t}$ denote the representations for the same timestamp t but from two argumentations of X_i. The temporal contrastive loss for the $i-$th time series at timestamp t can be formulated as:

$$\ell_{temp}^{(i,t)} = -\log \frac{\exp(r_{i,t} \cdot r'_{i,t})}{\sum t' \in \Omega\left(\exp\left(r_{i,t} \cdot r'_{i,t'}\right) + \mathbb{I}_{[t \neq t']} \exp\left(r_{i,t} \cdot r_{i,t'}\right)\right)} \tag{1}$$

where Ω is the set of timestamps within the overlap of the two subseries, and $\mathbb{I}$ is the indicator function.

The instance-wise contrastive loss indexed with (i, t) can be formulated as:

$$\ell_{inst}^{(i,t)} = -\log \frac{\exp\left(r_{i,t} \cdot r'_{i,t}\right)}{\sum_{j=1}^{B}\left(\exp\left(r_{i,t} \cdot r'_{j,t}\right) + \mathbb{I}_{[i \neq j]} \exp\left(r_{i,t} \cdot r_{j,t}\right)\right)} \tag{2}$$

where B denotes the batch size. We use representations of other time series at timestamp t in the same batch as negative samples.

The overall loss is defined as:

$$\mathcal{L}_{dual} = \frac{1}{NT} \sum_{i} \sum_{\mathrm{t}} (\ell_{temp}^{(i,t)} + \ell_{inst}^{(i,t)}) \tag{3}$$

3.3.2. AR (AutoRegressive Component)

In the paper [38], a model called LSTNet is proposed, which enhances the robustness of nonlinear deep learning models to scale violations in time series data by introducing a traditional auto-regressive linear component alongside the nonlinear neural network component. This model also improves the accuracy of short-term forecasting. Building upon this idea, we introduce the auto-regressive component into the Transformer model. Due to the significant fluctuation in power load data, conventional deep learning models may not be sensitive enough to local extreme changes. To address this issue, we decompose the final prediction of power load into a linear component (focused on local scale issues) and the non-linear component of the Transformer. In the architecture of the load forecasting model, we employ the classical auto-regressive (AR) component as the linear component. The AR component can be represented by the following parameters:

$$h_t^L = \sum_{k=0}^{q^{ar}-1} W_k^{ar} y_{t-k} + b^{ar} \tag{4}$$

Among them, h_t^L is the predicted value of the AR component, which has a dimension of n. q^{ar} is the size of the input window on the input matrix. W^{ar} represents the weight assigned by the AR component to each linear component, with a dimension of q^{ar}, and b^{ar} is the bias value of the linear autoregressive component.

We utilize h_t^T to denote the output of the predictive component of the Transformer model. $\hat{Y}_t$ signifies the ultimate predicted electricity load value, and $\hat{Y}_t$ can be represented as:

$$\hat{Y}_t = h_t^T + h_t^L \tag{5}$$

3.3.3. Transformer Model

The Transformer model, initially developed for natural language processing, can also be effectively applied to multivariate time series prediction. By treating each element at each time step of the time series as a word embedding input, the Transformer leverages its superior ability for parallelization and modeling long-term dependencies, surpassing tradi-

tional recurrent neural networks (RNNs). This makes it particularly suitable for handling complex multivariate time series data, such as electric load data. The core structure of the Transformer accommodates this data type by employing multiple encoding layers comprising components such as multi-head attention, feed-forward fully connected, residual connections, and normalization layers. These layers collectively capture the interdependencies and interactions across different dimensions of the multivariate time series. Through the utilization of self-attention, the model identifies crucial features and relationships within the input sequence. In the decoding phase, the Transformer incorporates similar layers, including an additional multi-head attention layer, enabling it to consider both the encoded representations and past predictions, resulting in accurate forecasts for future values. By leveraging attention mechanisms, non-linear transformations, and residual connections, the Transformer effectively captures intricate dependencies and patterns within multivariate time series data, making it a powerful tool for various forecasting tasks. The component of the Self-Attention is illustrated in the following Figure 4.

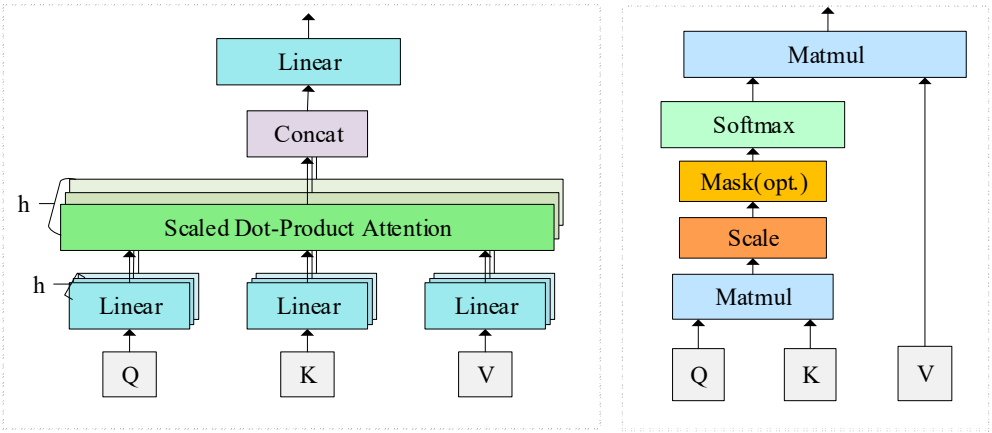

Figure 4. Multi-Head Self-Attention Component.

The Multi-Head Self-Attention in the encoding layer can be represented as follows:

$$\text{Multihead}(H) = concat(\text{head}_1, ..., \text{head}_h)W^O \tag{6}$$

$$\text{head}_i = softmax(\frac{QK^T}{\sqrt{d_k}}V) \tag{7}$$

where Q, K, and V are query, key, and value vectors, respectively, used to calculate the attention and taken from the input matrix, W^o refers to the weight matrix of the linear layer. Given that h is the number of heads and d_k represents the dimensionality of the attention heads.

3.3.4. Cross-Dimensional-Self-Attention Module

In the paper [39], a method called Cross-Shaped Self-Attention mechanism is proposed, which allows for the simultaneous calculation of attention weights in both horizontal and vertical directions. This method has shown promising performance in the field of computer vision. Motivated by this idea, we introduce the Cross-Dimensional-Self-Attention module into the Transformer model for time series forecasting.

The Cross-Dimensional-Self-Attention mechanism allows for simultaneous attention to both the positional relationships within the sequence data and the correlations across different dimensions, achieving the goal of global attention. By introducing the Cross-Dimensional-Self-Attention mechanism, we can capture complex associations between different dimensions of the multivariate data and enhance the richness of feature representations. This improvement enables the model to better understand and utilize the feature

information in multivariate time series data, thereby improving prediction accuracy. Furthermore, the Cross-Dimensional-Self-Attention mechanism helps mitigate the interference of outliers on the prediction results, enhancing the robustness of the model. Therefore, by introducing the Cross-Dimensional-Self-Attention mechanism, we can better capture the intrinsic relationships and feature representations in multivariate time series forecasting tasks, leading to improved model performance.

We have proposed an enhanced Transformer model by integrating the Cross-Dimensional-Self-Attention mechanism with the Transformer. Cross-Dimensional-Self-Attention mechanism enables Transformer to attend to both the positional relationships within the sequence data and the correlations across different dimensions, achieving a comprehensive global attention. This mechanism captures complex associations between different dimensions of the multivariate data, enriching the feature representations. Consequently, the model gains a better understanding of the feature information in multivariate time series data, resulting in improved prediction accuracy. Additionally, the Cross-Dimensional-Self-Attention mechanism helps mitigate the impact of outliers on the prediction results, enhancing the model's robustness. By incorporating the Cross-Dimensional-Self-Attention mechanism, we can effectively capture intrinsic relationships and feature representations in multivariate time series forecasting tasks, leading to superior model performance. Figure 5 provides a comparative analysis of different attention mechanisms.

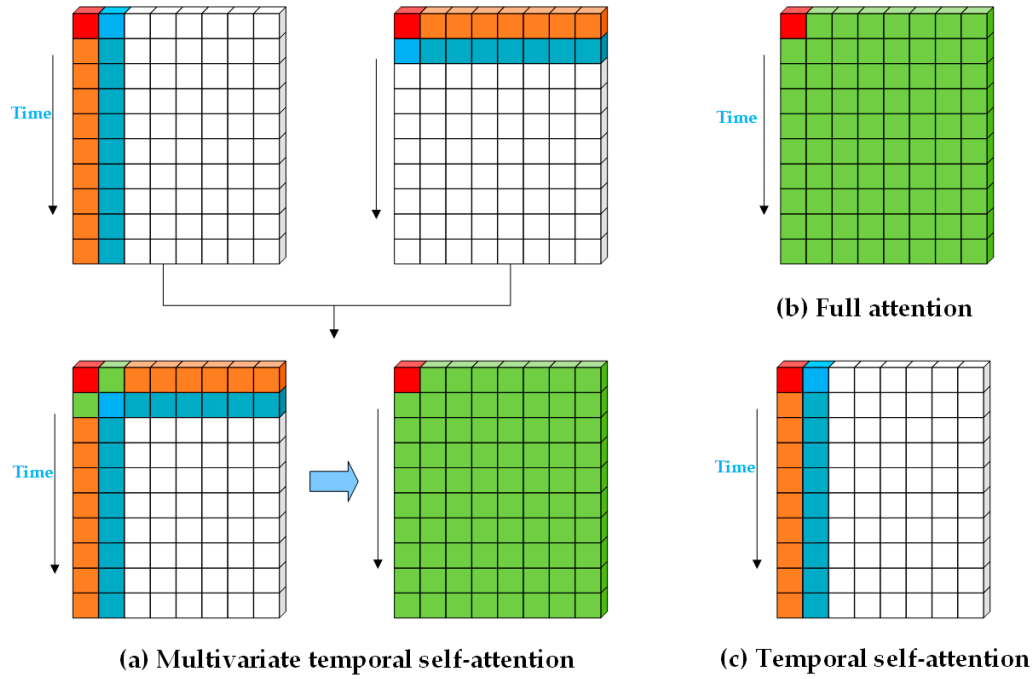

Figure 5. Comparing Various Attention Mechanisms.

Figure Description: The figure illustrates different attention mechanisms and their respective attention scopes. The red and blue dots represent the values of a specific dimension at a certain time step. The corresponding light-colored blocks represent the attention range of the current element, indicating which elements it attends to within its local context. The green color represents the global attention scope, indicating that the element attends to all elements across different dimensions and time steps. The introduction provided is as follows:

1. Temporal Self-Attention (Vertical Weight Allocation) + Multivariate Self-Attention (Horizontal Weight Allocation) ≈ Global Attention Allocation.

2. Global Self-Attention (Weight Allocation Across the Entire Sequence) (Disadvantage: The model becomes complex, especially for long forecasting tasks, its complexity becomes intolerable).
3. Temporal Self-Attention (Weight Allocation Along the Temporal Axis) (Disadvantage: Lack of vertical attention span, resulting in information loss and lower accuracy).

Let us assume that our input data is a matrix X, which represents encoded multivariate time series data. In this matrix, n represents the number of dimensions (features) horizontally (multivariate features), and t represents the number of dimensions vertically (temporal features). The process of the Cross-Dimensional-Self-Attention module can be described as follows:

$$Z_h = softmax(\frac{Q_h K_h^T}{\sqrt{d_k}})V_h \tag{8}$$

$$Z_v = softmax(\frac{Q_v K_v^T}{\sqrt{d_k}})V_v \tag{9}$$

$$h_t^T = Z_h + Z_v \tag{10}$$

In this context, $Q_h \in R^{T\times n}$, $K_h \in R^{T\times n}$ and $V_v \in R^{T\times n}$ are obtained through linear transformations of the input data X, while the $Q_h \in R^{T\times n}$, $K_h \in R^{T\times n}$ and $V_v \in R^{T\times n}$ are obtained through linear transformations of representation vectors $\{R_1, R_2, ..., R_k\}$ encoded by TS2Vec layer, representing the query, key, and value vectors for vertical self-attention. The parameter d_k represents the dimensionality of the attention heads. The $softmax$ function is used to normalize the attention weights. Finally, the output Z_h from vertical self-attention and the output Z_v from Cross-Dimensional-Self-Attention are linearly combined to obtain the final attention output h_t^T.

3.4. TS2ARCformer

TS2ARCformer is an integrated model for short-term load prediction, combining time series embedding learning layer (TS2Vec), an autoregressive component (AR), and an enhanced Transformer model. TS2Vec layer transforms historical load data into high-dimensional vector representations, capturing nonlinear features and periodic patterns. The AR component predicts current load values based on previous time steps, capturing temporal dependencies. The enhanced Transformer model incorporates a Cross-Dimensional-Self-Attention module, considering both internal dependencies and relationships with related tasks. The predictions from the AR component and Transformer model are combined, leveraging the strengths of each for improved accuracy and stability. By considering time features, temporal dependencies, and associations with related tasks, TS2ARCformer enhances efficiency and accuracy in electricity load forecasting. This hybrid approach has practical implications for power system operation and planning. The model takes the input data $X_T = \{y_1, y_2, \ldots, y_T\}$ and generates the output $\{y_{T+1}, y_{T+2}, \ldots, y_{T+h}\}$. Each y represents the historical data up to the current timestamp, and h denotes the size of the prediction window. In this paper, we propose a load forecasting model structure as shown in Figure 6, which only utilizes the encoder part of the Transformer. The model incorporates the extensive use of the Cross-Dimensional-Self-Attention mechanism. It takes historical load-related data as input and generates future multi-step load predictions as output.

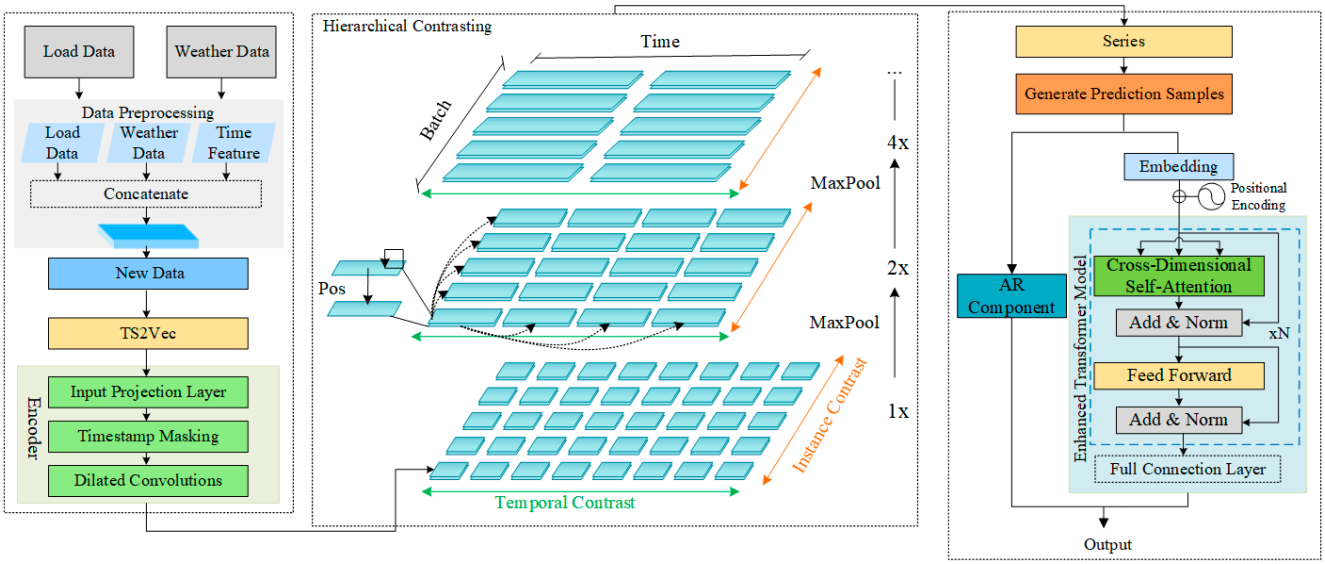

Figure 6. The structure of the TS2ARCformer Model.

4. Experimental Results and Analysis

4.1. Evaluation Metrics

This article evaluates the performance of prediction models by using four evaluation criteria: MAPE (Mean Absolute Percentage Error), RMSE (Root Mean Square Error), MAE (Mean Absolute Error), and R^2 (Coefficient of Determination). In short-term power load forecasting, a higher accuracy of the prediction model is indicated by a smaller value of the first three mentioned criteria. On the other hand, a model with good interpretability is represented by a larger value of coefficient of determination, R^2. The calculation formulas are shown in Equations (11)–(14):

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{11}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{12}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|(y_i - \hat{y}_i)| \tag{13}$$

$$R^2 = \frac{\sum_{i=1}^{n}(\overline{y}_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(\overline{y}_i - y_i)^2} \tag{14}$$

The loss function in this study applies Mean Squared Error (MSE), which measures the deviation between predicted and actual values, as demonstrated in Equation (15):

$$Loss = \frac{1}{n}\sum\nolimits_{i=0}^{n}(\mathrm{y}_i - \hat{y}_i)^2 \tag{15}$$

Explanation: The predicted load values and true load values of the i-th sampling point are represented by $\hat{y}_i$ and y_i, respectively, with n being the total number of test samples in this study.

4.2. Data Preparation

The dataset used in this study is the standard dataset provided by the National College Student Mathematical Contest in Electrical Engineering. The dataset includes electricity load data and weather data for area1 and area2 from 1 January 2009 to 10 January 2015. The electricity load data are sampled every 15 min, with 96 samples per day, and the unit is in MW. The weather data include daily maximum temperature, daily minimum temperature, daily average temperature, daily relative humidity, and daily rainfall. Missing values in the dataset are filled with the column's average value. The dataset is divided into training, testing, and validation sets. The proposed electricity load forecasting model in this study uses a sliding historical window size of 24 and a future window size of 24. This means that based on the historical 24 h load-related data, the model predicts the load for the next 24 h. The experiments were conducted on a platform equipped with NVIDIA RTX 3090, and the deep learning framework Pytorch was used to build and train the models. To facilitate model training, the data were normalized using the min-max scaling method to a range of [0, 1]. To gain a deeper understanding of the electricity load data, a specific dataset was carefully selected for analysis, as depicted in Figure 7. Figure 7A shows the trend and volatility of the electricity load data, indicating significant fluctuations and non-stationarity with some periodic patterns. Figure 7B displays the autocorrelation coefficients of the load data, revealing a high autocorrelation even at longer time lags, indicating the presence of significant long-term dependence. Therefore, the Transformer model, capable of addressing long-term dependencies, was chosen for modeling. Figure 7C illustrates the correlation between different data features, highlighting a strong correlation between weather factors and load values. Thus, in this study, the impact of weather factors on load forecasting was considered to improve prediction accuracy. Figure 7D depicts the distribution of electricity load from 2009 to 2015.

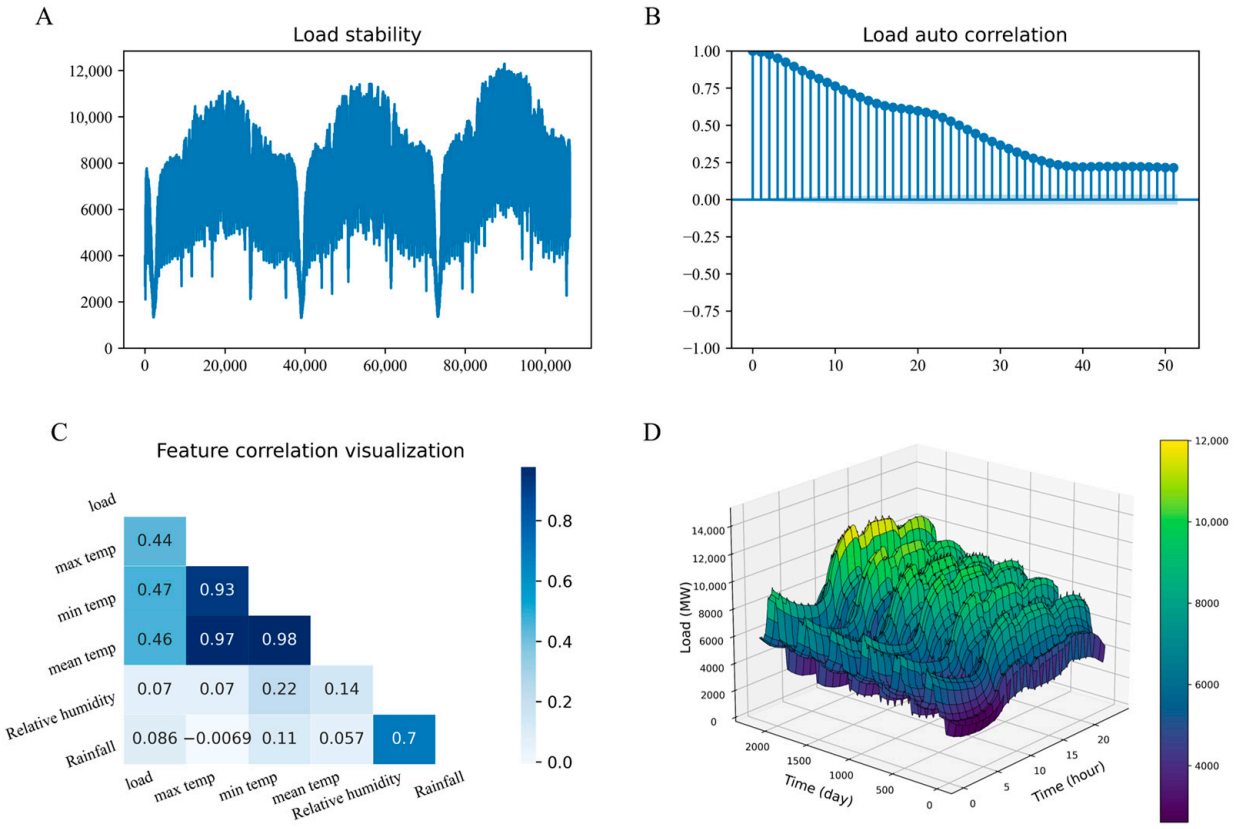

Figure 7. The analysis of Electric Power Load Data. Explanation: (**A**) represents the analysis of load data stationarity. (**B**) illustrates the analysis of load data autocorrelation. (**C**) depicts the analysis of feature correlations. (**D**) displays the three-dimensional visualization of the load data.

4.3. Experimental Setup

To validate the effectiveness of TS2ARCformer, this study compared it with five commonly used deep learning models from the RNNs and Transformer classes. The nine baseline models selected for comparison were LSTM, GRU, Transformer, TS2Vec, TS2Vec-LSTM, and TS2Vec-GRU, Seq2Seq, TCN, TCN-Transformer. The following is a brief introduction to these nine models:

LSTM (Long Short-Term Memory): LSTM is a widely used recurrent neural network (RNN) for time series modeling. It captures long-term dependencies in sequences through gated mechanisms.
GRU (Gated Recurrent Unit): GRU is another type of gated recurrent neural network that simplifies the gating mechanism while capturing sequence dependencies effectively.
Transformer: Transformer is a model with self-attention mechanism, initially used in natural language processing. It captures dependencies in a sequence and processes long sequences efficiently.
TS2Vec: TS2Vec is a representation learning method that encodes multi-dimensional data into fixed-dimensional vectors. It extracts features for prediction tasks. In this case, TS2Vec is used for encoding, followed by a fully connected layer for prediction.
TS2Vec-LSTM: TS2Vec-LSTM combines TS2Vec with LSTM for sequence modeling and prediction, capturing multi-dimensional features and time dependencies.
TS2Vec-GRU: TS2Vec-GRU is similar to TS2Vec-LSTM but uses GRU for sequence modeling, with fewer parameters and higher learning efficiency.
Seq2Seq: The Seq2Seq model, widely employed for sequence-to-sequence tasks, consists of an encoder and a decoder. Both the encoder and the decoder are built using LSTM networks.

TCN is a type of neural network architecture designed specifically for processing time series data. It utilizes 1D convolutions to capture temporal patterns in the data. TCN can efficiently model long-range dependencies in time series, making it suitable for tasks such as sequence-to-sequence prediction and forecasting.

TCN-Transformer is a hybrid model that combines the Temporal Convolutional Network (TCN) and the Transformer architecture. TCN is used to capture local temporal patterns in the time series data, while the Transformer handles global dependencies and long-range interactions. The table below presents detailed information on the compared models and the proposed method, including their parameter configurations. Models were evaluated on the same dataset, and grid search was performed for parameter selection. Cross-validation was used to estimate performance on the validation set. For more detailed parameter information, please refer to Tables 1–6.

Table 1. Parameter setting of TS2Vec.

Item	Hyper-Parameter
represent_dimension	320
lr	0.001
batch_size	256
epochs	200

Table 2. Parameter setting of LSTM and GRU.

Item	Hyper-Parameter
batch_size	16
hidden_size	256
num_layers	1
gamma	0.9
weight_decay	1×10^{-5}
step_size	8
loss_function	MSE
epochs	200

Table 3. Parameter setting of Transformer.

Item	Hyper-Parameter
batch_size	16
d_model	256
optimization	Adam
n_head	8
num_layers	10
step size	8
lr	0.001
dropout	0.1
epochs	200
gamma	0.9
weight_decay	1×10^{-5}
loss_function	MSE

Table 4. Parameter setting of Seq2Seq.

Item	Hyper-Parameter
batch_size	32
step_size	8
lr	0.001
epochs	200
node in hidden layer	256
optimization	Adam
loss_function	MSE

Table 5. Parameter setting of TCN.

Item	Hyper-Parameter
batch_size	32
step_size	8
lr	0.001
num_channels	[6]
kernal_size	7
epochs	200
optimization	Adam
loss_function	MSE

Table 6. Parameter setting of TS2ARCformer.

Item	Hyper-Parameter
batch_size	16
step size	8
lr	0.001
gamma	0.9
weight_decay	1×10^{-5}
epochs	200
optimization	Adam
loss_function	MSE
d_model_trvs	256
ar_window	48
ar_output	48

Explanation: Since the TS2Vec model shares the same parameters with the compared models TS2Vec-LSTM, TS2Vec-GRU, and TS2Vec-Transformer, the parameters of the TS2Vec model are separately listed to avoid redundancy in the parameter table.

4.4. Comparative Experiments

In the experiment, we utilize the TS2ARCformer model to predict the short-term electricity load in area1 and area2. We then compare the results with several other models, such as LSTM, GRU, TS2Vec, TS2Vec-LSTM, TS2Vec-GRU, Transformer, Seq2Seq, TCN, TCN-Transformer. The temporal scale is represented on the horizontal axis, while the load data values are represented on the vertical axis. The obtained results are shown below:

The performance of the LSTM, GRU, Transformer, TS2Vec, TS2Vec-LSTM, TS2Vec-GRU, Seq2Seq, TCN, TCN-Transformer and TS2ARCformer models in predicting load data on the area1 test dataset is shown in Figure 8 of this paper. The x-axis represents the time scale, while the y-axis represents the load values. In the dataset, the load data exhibit clear periodic variations. It can be observed that LSTM, GRU, TS2Vec, Transformer, Seq2Seq and TCN models have poor fit to the data's changing trends. However, the prediction models using TS2Vec encoding achieve better fit than individual models. Furthermore, we notice that utilizing Temporal Convolutional Network (TCN) for encoding the data on dataset area1, and then employing Transformer for prediction, resulted in better outcomes in comparison to solely using Transformer. Table 7 presents the experimental results of various metrics for LSTM, GRU, Transformer, TS2Vec, TS2Vec-LSTM, TS2Vec-GRU, Seq2Seq, TCN, TCN-Transformer and TS2ARCformer models. From the table, it can be observed that among the individual models, the Transformer model has the worst fit compared to LSTM, GRU, Seq2Seq and TCN models, with LSTM performing the best. A more visual comparison is shown in Figure 9. The combination model with TS2Vec layer encoding achieves better results than individual models, indicating that TS2Vec layer effectively encodes the data, enhances the model's information extraction capability, and improves prediction accuracy. It is worth noting that the single TS2Vec model with a fully connected layer does not achieve high prediction accuracy compared to TS2Vec-LSTM and TS2Vec-GRU models, suggesting that the role of the prediction model is also crucial after data representation learning. Additionally, our proposed TS2ARCformer model exhibits the best prediction performance on dataset area1 compared to the baseline Transformer model, showing significant improvements across multiple metrics. It reduces the MAPE metric by 43.2% and the MSE metric by 60.7%. As depicted in Figure 8, TS2ARCformer demonstrates more accurate peak predictions of power peaks, which could be attributed to the Cross-Dimensional-Self-Attention mechanism learning more interdependencies, enabling the model to better capture the growing trend of load data. Moreover, we find that TS2ARCformer not only accurately captures the details of load data changes, possibly due to the AR component within TS2ARCformer enhancing the short-term prediction capability of the overall model. To validate the generalization of TS2ARCformer for multi-to-multi prediction tasks, we further conducted comparative experiments on the power load dataset area2. As shown in the diagram, Table 8 (A more visual comparison is shown in Figures 10 and 11) demonstrate that TS2ARCformer achieves a 37.8% reduction in MAPE metric and a 57.9% reduction in MSE metric compared to the baseline model. Overall, our proposed TS2ARCformer model exhibits strong generalization ability, achieving state-of-the-art results on both dataset area1 and dataset area2.

Table 7. Performance Comparison of Different Models on Load Testing Dataset area1.

Models	MAPE%	MSE/MW	MAE/MW	R^2
LSTM	6.42	0.00419	0.03813	0.8733
GRU	7.21	0.00495	0.04632	0.8413
Transformer	7.43	0.00512	0.04666	0.8312
TS2Vec	6.99	0.00448	0.04363	0.8652
TS2Vec-LSTM	5.03	0.00254	0.03121	0.9222
TS2Vec-GRU	4.76	0.00252	0.03037	0.9211
Seq2Seq	7.10	0.00473	0.04566	0.8491
TCN	7.01	0.00489	0.04401	0.8405
TCN-Transformer	6.75	0.00467	0.04277	0.8586
Ours	4.22	0.00201	0.02623	0.9412

Table 8. Performance Comparison of Different Models on Load Testing Dataset area2.

Models	MAPE%	MSE/MW	MAE/MW	R^2
LSTM	5.79	0.00319	0.04017	0.8929
GRU	5.55	0.00268	0.03771	0.9182
Transformer	5.74	0.00314	0.03813	0.9031
TS2Vec	5.30	0.00232	0.03494	0.9307
TS2Vec-LSTM	4.41	0.00186	0.03012	0.9395
TS2Vec-GRU	4.08	0.00179	0.02859	0.9401
Seq2Seq	5.63	0.00283	0.03891	0.9119
TCN	5.26	0.00275	0.03533	0.9143
TCN-Transformer	5.56	0.00279	0.03724	0.9169
Ours	3.57	0.00132	0.02376	0.9622

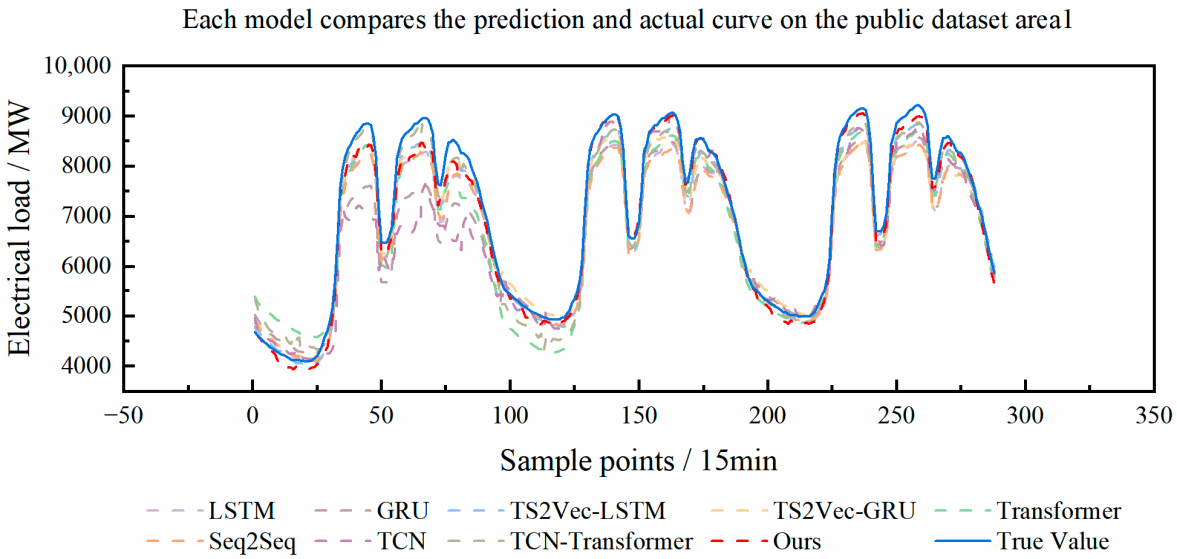

Figure 8. The plot of the forecasting results of all models on the public dataset area1.

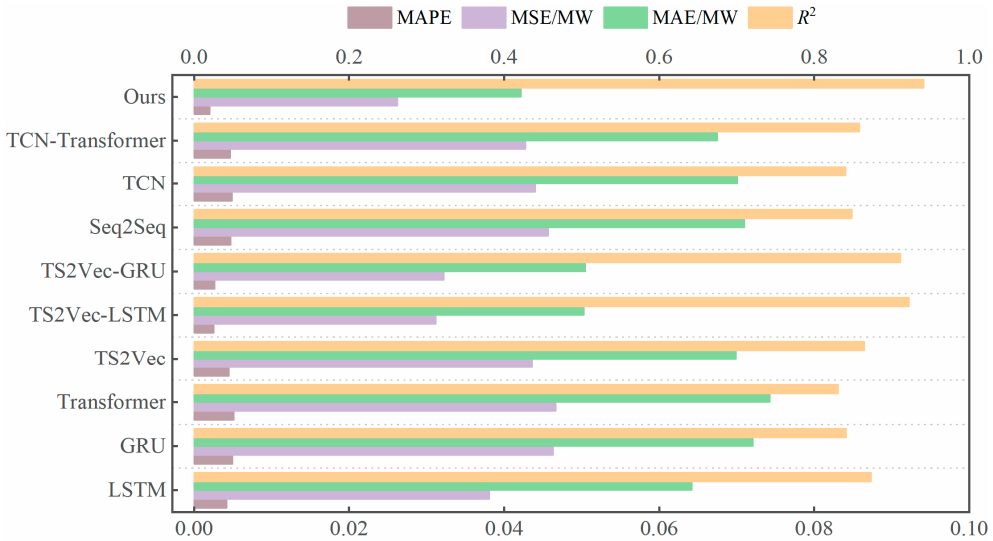

Figure 9. Comparison experiment results of 10 models on dataset area1.

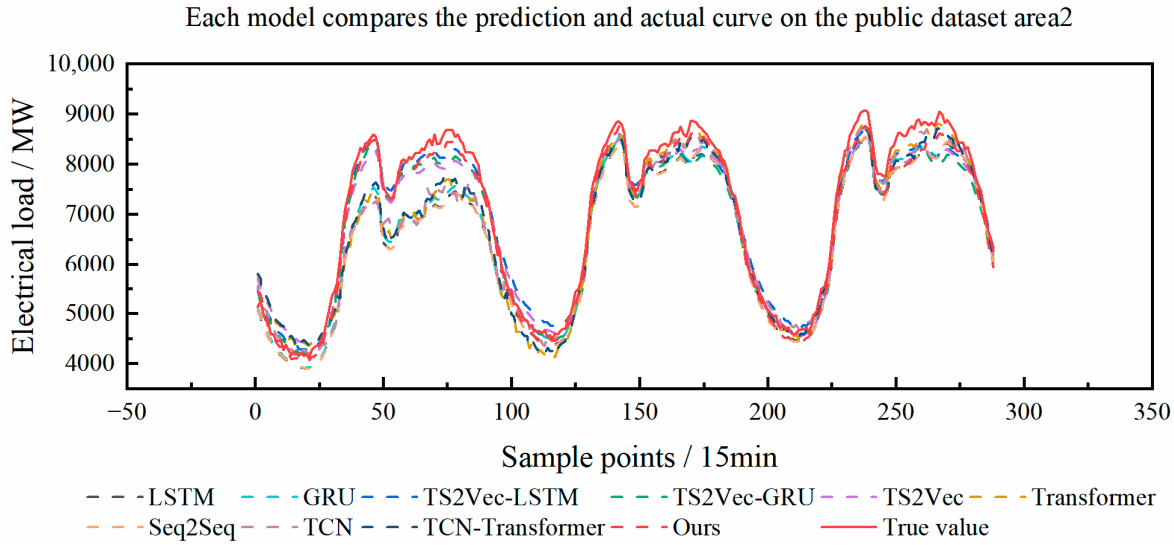

Figure 10. The plot of the forecasting results of all models on the public dataset area2.

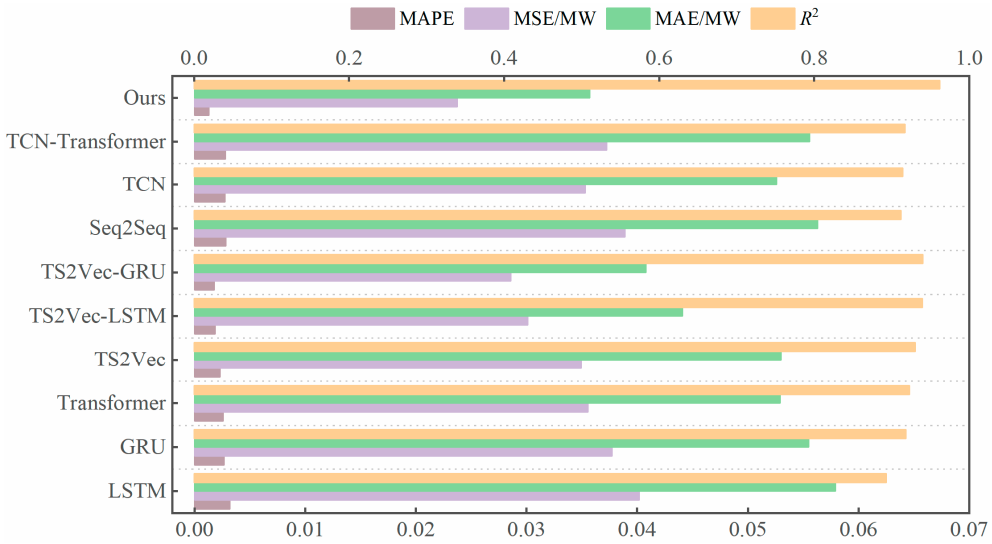

Figure 11. Comparison experiment results of 10 models on dataset area2.

In addition, we compared the computational resources of various short-term electricity load forecasting models. These models include LSTM, GRU, Transformer, TS2Vec, TS2Vec-LSTM, TS2Vec-GRU, Seq2Seq, TCN, TCN-Transformer, and our proposed model (referred to as "Ours"). As shown in Table 9, from the perspective of Flops, Training Time (in seconds), and Params (Size of Parameters of each model), the following observations can be made:

1. In terms of computational resource consumption, TCN (206.21 K Flops) is one of the most efficient models, while Transformer (305.82 M Flops) and our model "Ours" (406.84 M Flops) require higher computational resources.
2. Regarding training time, TCN (236 s) and TS2Vec (185 s) are the quickest to train, while our model "Ours" (2250 s) and TCN-Transformer (1805 s) take longer to complete training.
3. In the number of model parameters, TCN (2.063 K Params) and Seq2Seq (565.344 K Params) have the fewest parameters, while Transformer (5.523 M Params) and our model "Ours" (6.621 M Params) have more parameters.

Table 9. Comparison of Computational Resources for Different Models.

Models	Flops	Training Time	Params
LSTM	28.51 M	470 s	295.01 K
GRU	21.99 M	390 s	227.424 K
Transformer	305.82 M	1449 s	5.523 M
TS2Vec	61.02 M	185 s	637.95 K
TS2Vec-LSTM	89.53 M	635 s	932.96 K
TS2Vec-GRU	83.01 M	565 s	865.37 K
Seq2Seq	54.65 M	1629 s	565.344 K
TCN	206.21 K	236 s	2.063 K
TCN-Transformer	308.69 M	1805 s	5.885 M
Ours	406.84 M	2250 s	6.621 M

Although our proposed model "Ours" exhibits relatively higher computational resource consumption compared to some other models in the comparison, it demonstrates significantly improved predictive performance. This higher resource consumption is a trade-off that we willingly accept to achieve better forecasting accuracy. However, it is crucial to note that the nature of the electricity forecasting task allows for relatively small resource consumption across all models. In this context, our model's resource consumption falls within an acceptable range. In practice, the focus should be on the predictive accuracy, which is the more important metric for electricity load forecasting tasks. The improved accuracy of our model can lead to more reliable and efficient decision-making, making the higher resource consumption worthwhile. As such, the trade-off is justified, as the predictive performance gains outweigh the incremental resource cost.

4.5. Ablation Experiment

To validate the effectiveness of incorporating the TS2Vec layer, AutoRegressive (AR) component, and Cross-Dimensional-Self-Attention module in enhancing the performance of the Transformer model for long sequence prediction, we conduct ablation experiments on two datasets under the same experimental settings. The dataset is divided into a ratio of 6:2:2 for training, testing, and validation, respectively. We comprehensively evaluate the impact of these three modules on the experiments using various evaluation metrics, including MAPE, MSE, MAE, etc. (where MSE and MAE are normalized). Based on the results of these metrics, we assess the effectiveness of this approach.

By conducting these ablation experiments and analyzing the results, we gain insights into the impact of the AutoRegressive (AR) Component, TS2Vec Layer, and Cross-Dimensional-Self-Attention module on the performance of the Transformer prediction model in different prediction scenarios. The results of the ablation experiments are presented in Tables 10 and 11. To display the results in Tables 10 and 11 more clearly, a more visual comparison is depicted in Figures 12 and 13. The results are analyzed as follows:

1. Data Set Comparison: In the initial step, we evaluate the performance of the baseline model on two distinct datasets. It is observed that the Transformer model obtains a MAPE of 7.43% on dataset area1, whereas the baseline model achieves a MAPE of 5.74% on dataset area2. This suggests that there are variations between the two datasets, which establishes a reference point for future ablation experiments.
2. Impact of TS2Vec Layer: In our study, we conduct representation learning on the power load data using the TS2Vec layer and feed the learned representations into the Transformer model for prediction. We perform experiments on two different datasets and observe the following results. When we introduce the TS2Vec layer, we achieve a MAPE of 6.01% on dataset area1. This results in a substantial reduction of 19.11% compared to the baseline. Similarly, on dataset area2, the MAPE decreases to 4.84%, showing a reduction of 15.6%. These results clearly demonstrate that the inclusion of Module 1 positively impacts the performance of the prediction model on both datasets. Notably, dataset area1 experiences a more pronounced improvement.

Additionally, we observe improvements in three other performance metrics when compared to the baseline Transformer model without the TS2Vec layer. This suggests that the utilization of the TS2Vec layer for representing temporal data enhances the performance of the prediction model in downstream tasks. The analysis of these results indicates that training the prediction model using the TS2Vec layer yields a significant improvement in performance compared to the Transformer model trained without the TS2Vec layer.

3. Impact of AR Component and Cross-Dimensional-Self-Attention Module: We conduct an evaluation to assess the impact of incorporating the Autoregressive (AR) component and Cross-Dimensional-Self-Attention module into the Transformer architecture of our forecasting model. The results show that these modules improve the predictive accuracy of the model. Specifically, when the AR component is included, the MAPE on dataset area1 decreases from 7.43% to 6.97%, representing a reduction of 6.19%. Similarly, on dataset area2, the MAPE decreases from 5.74% to 4.77%, indicating a reduction of 16.89%. These findings demonstrate the positive effect of the AR component on both datasets, with a slightly more significant impact on dataset area2. Furthermore, the integration of the Cross-Dimensional-Self-Attention module further enhances the model's performance. On dataset area1, the MAPE decreases to 5.55%, resulting in a reduction of 25.30%. On dataset area2, the MAPE decreases to 4.74%, showing a reduction of 17.42%. The results demonstrate that the Cross-Dimensional-Self-Attention module successfully captures relationships between different time steps, leading to improved forecasting accuracy on both datasets. Furthermore, we integrated these two modules to evaluate their combined impact on the model's performance. The results demonstrate that when the AR component and Cross-Dimensional-Self-Attention module are used together, the MAPE on dataset area1 further decreases to 5.22%, representing a relative reduction of 29.74%. Similarly, on dataset area2, the MAPE decreases to 4.11%, showing a relative reduction of 28.40%. These findings highlight the further enhancement of the model's performance on both datasets through the integration of the Cross-Dimensional-Self-Attention module with the AR component. In summary, the experimental findings indicate that incorporating the AR component and Cross-Dimensional-Self-Attention module positively affects the performance of the forecasting model. The incorporation of these modules results in significant decreases in MAPE and three other metrics for both datasets, highlighting their effectiveness in capturing temporal dependencies and enhancing the model's predictive abilities.

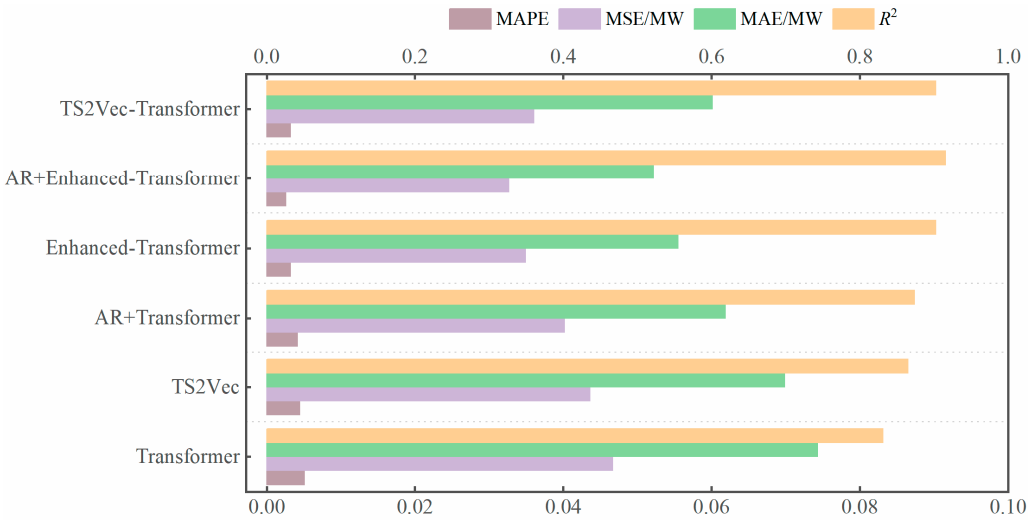

Figure 12. Ablation experiment results of 6 models on dataset area1.

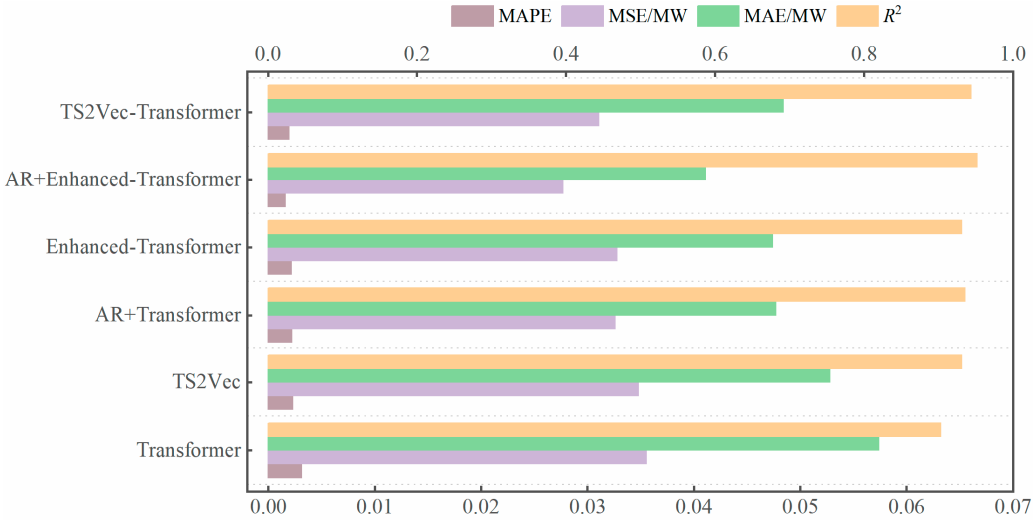

Figure 13. Ablation experiment results of 6 models on dataset area2.

Table 10. Ablation Experiment Results on dataset area1.

Models	MAPE%	MSE/MW	MAE/MW	R^2
Transformer	7.43	0.00512	0.04666	0.8312
TS2Vec	6.99	0.00448	0.04363	0.8652
AR + Transformer	6.97	0.00424	0.04382	0.8700
Enhanced Transformer	5.55	0.00320	0.03490	0.9024
AR + Enhanced Transformer	5.22	0.00262	0.03267	0.9157
TS2Vec-Transformer	6.01	0.00325	0.03606	0.9029

Table 11. Ablation Experiment Results on dataset area2.

Models	MAPE%	MSE/MW	MAE/MW	R^2
Transformer	5.74	0.00314	0.03556	0.9031
TS2Vec	5.28	0.00230	0.0348	0.9313
AR + Transformer	4.77	0.00221	0.0326	0.9358
Enhanced Transformer	4.74	0.00218	0.0328	0.9311
AR + Enhanced Transformer	4.11	0.00161	0.0277	0.9524
TS2Vec-Transformer	4.84	0.00198	0.0311	0.9439

5. Conclusions

This paper introduces a novel framework called TS2ARCformer for multivariate time series forecasting. The framework combines the TS2Vec layer for encoding multi-dimensional features with an enhanced Transformer model and an autoregressive component (AR) for predicting future data. The enhanced Transformer model incorporates Cross-Dimensional-Self-Attention mechanism to improve the model's ability to extract information from multi-dimensional features. Through extensive experiments, our proposed method achieves state-of-the-art performance on multiple datasets in multivariate time series forecasting. Ablation experiments were conducted to validate the effectiveness of each component of TS2ARCformer.

In conclusion, TS2ARCformer offers a promising framework for multivariate time series forecasting, with the potential to make significant contributions to the field. In the future, we aim to apply TS2ARCformer to forecast other multivariate datasets and further explore its generalizability. We will also focus on optimizing model training time and computational resources to enhance overall efficiency while maintaining high predictive accuracy.

Author Contributions: Writing—original draft preparation, W.Z.; writing—review and editing, S.L.; supervision, P.W.; project administration, P.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the grants of National Key Research and Development Program of China, grant number 2021YFB1714400, and the Jilin Provincial Science and Technology Innovation Center for Network Database Application, grant number YDZJ202302CXJD027.

Data Availability Statement: Not applicable.

Acknowledgments: The author expresses gratitude to the National College Student Electrical Engineering Mathematical Modeling Competition for providing the dataset on electric power load. I would like to extend my gratitude to the Joint Key Laboratory of Big Data Science and Engineering in Jilin Province for their assistance.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

STLF	Short-term Load Forecasting.
AR	AutoRegressive.
TCN	Temporal Convolutional Network.
CNN	Convolutional Neural Network.
RNN	Recurrent Neural Network.
LSTM	Long Short-term Memory.
GRU	Gated Recurrent Unit.
Seq2Seq	Sequence-to-Sequence.
MAPE	Mean Absolute Percentage Error.
RMSE	Root Mean Square Error.
MAE	Mean Absolute Error.
R^2	Coefficient of Determination.

References

1. Kim, N.; Park, H.; Lee, J.; Choi, J.K. Short-term electrical load forecasting with multidimensional feature extraction. *IEEE Trans. Smart Grid* **2022**, *13*, 2999–3013. [CrossRef]
2. Sharma, A.; Jain, S.K. A novel seasonal segmentation approach for day-ahead load forecasting. *Energy* **2022**, *257*, 124752. [CrossRef]
3. Kong, W.; Dong, Z.Y.; Jia, Y.; Hill, D.J.; Xu, Y.; Zhang, Y. Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Trans. Smart Grid* **2017**, *10*, 841–851. [CrossRef]
4. Lu, C.; Li, J.; Zhang, G.; Zhao, Z.; Bamisile, O.; Huang, Q. A GRU-based short-term multi-energy loads forecast approach for integrated energy system. In Proceedings of the 2022 4th Asia Energy and Electrical Engineering Symposium (AEEES), Chengdu, China, 25–28 March 2022; pp. 209–213.
5. Liu, R.; Chen, L.; Hu, W.; Huang, Q. Short-term load forecasting based on LSTNet in power system. *Int. Trans. Electr. Energy Syst.* **2021**, *31*, e13164. [CrossRef]
6. Zhang, Y.; Tang, S.; Yu, G. An interpretable hybrid predictive model of COVID-19 cases using autoregressive model and LSTM. *Sci. Rep.* **2023**, *13*, 6708. [CrossRef]
7. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271.
8. Guo, C.; Kang, X.; Xiong, J.; Wu, J. A new time series forecasting model based on complete ensemble empirical mode decomposition with adaptive noise and temporal convolutional network. *Neural Process. Lett.* **2022**, *55*, 4397–4417. [CrossRef]
9. Staffell, I.; Pfenninger, S. The increasing impact of weather on electricity supply and demand. *Energy* **2018**, *145*, 65–78. [CrossRef]
10. Abideen, Z.U.; Sun, H.; Yang, Z.; Ahmad, R.Z.; Iftekhar, A.; Ali, A. Deep wide spatial-temporal based transformer networks modeling for the next destination according to the taxi driver behavior prediction. *Appl. Sci.* **2020**, *11*, 17. [CrossRef]
11. Xiang, Y.; Chen, J.; Yu, W.; Wu, R.; Liu, B.; Wang, B.; Li, Z. A Two-Phase Approach for Predicting Highway Passenger Volume. *Appl. Sci.* **2021**, *11*, 6248. [CrossRef]
12. Hernández-Callejo, L.; Baladrón, C.; Aguiar, J.M.; Calavia, L.; Carro, B.; Sánchez-Esguevillas, A.; Cook, D.J.; Chinarro, D.; Gomez, J. A study of the relationship between weather variables and electric power demand inside a smart grid/smart world framework. *Sensors* **2012**, *12*, 11571–11591. [CrossRef]

13. Yue, Z.; Wang, Y.; Duan, J.; Yang, T.; Huang, C.; Tong, Y.; Xu, B. Ts2vec: Towards universal representation of time series. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 22 February 22–1 March 2022; Volume 36, pp. 8980–8987.
14. Saber, A.Y.; Alam, A.K.M.R. Short term load forecasting using multiple linear regression for big data. In Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, 27 November–1 December 2017; pp. 1–6.
15. Sharma, S.; Majumdar, A.; Elvira, V.; Chouzenoux, E. Blind Kalman filtering for short-term load forecasting. *IEEE Trans. Power Syst.* **2020**, *35*, 4916–4919. [CrossRef]
16. Rendon-Sanchez, J.F.; de Menezes, L.M. Structural combination of seasonal exponential smoothing forecasts applied to load forecasting. *Eur. J. Oper. Res.* **2019**, *275*, 916–924. [CrossRef]
17. Singh, U.; Vadhera, S. Random Forest and Xgboost Technique for Short-Term Load Forecasting. In Proceedings of the 2022 1st International Conference on Sustainable Technology for Power and Energy Systems (STPES), Srinagar, India, 4–6 July 2022; pp. 1–6.
18. Zhang, J.; Zhang, Q.; Li, G.; Ma, Y.; Wang, C. Application of HIMVO-SVM in short-term load forecasting. In Proceedings of the 2020 Chinese Control And Decision Conference (CCDC), Hefei, China, 22–24 August 2020; pp. 768–772.
19. Janković, Z.; Selakov, A.; Bekut, D.; Đorđević, M. Day similarity metric model for short-term load forecasting supported by PSO and artificial neural network. *Electr. Eng.* **2021**, *103*, 2973–2988. [CrossRef]
20. Xuan, Y.; Si, W.; Zhu, J.; Sun, Z.; Zhao, J.; Xu, M.; Xu, S. Multi-model fusion short-term load forecasting based on random forest feature selection and hybrid neural network. *IEEE Access* **2021**, *9*, 69002–69009. [CrossRef]
21. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
22. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
23. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
24. Dong, X.; Qian, L.; Huang, L. Short-term load forecasting in smart grid: A combined CNN and K-means clustering approach. In Proceedings of the 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju, Republic of Korea, 13–16 February 2017; pp. 119–125.
25. Park, K.; Yoon, S.; Hwang, E. Hybrid load forecasting for mixed-use complex based on the characteristic load decomposition by pilot signals. *IEEE Access* **2019**, *7*, 12297–12306. [CrossRef]
26. Rafi, S.H.; Deeba, S.R.; Hossain, E. A short-term load forecasting method using integrated CNN and LSTM network. *IEEE Access* **2021**, *9*, 32436–32448. [CrossRef]
27. Gong, G.; An, X.; Mahato, N.K.; Sun, S.; Chen, S.; Wen, Y. Research on short-term load prediction based on Seq2seq model. *Energies* **2019**, *12*, 3199. [CrossRef]
28. Wu, L.; Kong, C.; Hao, X.; Chen, W. A short-term load forecasting method based on GRU-CNN hybrid neural network model. *Math. Probl. Eng.* **2020**, *2020*, 1–10. [CrossRef]
29. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; p. 30.
30. Wang, C.; Wang, Y.; Ding, Z.; Zheng, T.; Hu, J.; Zhang, K. A transformer-based method of multienergy load forecasting in integrated energy system. *IEEE Trans. Smart Grid* **2022**, *13*, 2703–2714. [CrossRef]
31. Tay, Y.; Dehghani, M.; Bahri, D.; Metzler, D. Efficient transformers: A survey. *ACM Comput. Surv.* **2022**, *55*, 1–28. [CrossRef]
32. Guo, S.; Lin, Y.; Wan, H.; Li, X.; Cong, G. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Trans. Knowl. Data Eng.* **2021**, *34*, 5415–5428. [CrossRef]
33. L'heureux, A.; Grolinger, K.; Capretz, M.A.M. Transformer-Based Model for Electrical Load Forecasting. *Energies* **2022**, *15*, 4993. [CrossRef]
34. Zhao, Z.; Xia, C.; Chi, L.; Chang, X.; Li, W.; Yang, T.; Zomaya, A.Y. Short-term load forecasting based on the transformer model. *Information* **2021**, *12*, 516. [CrossRef]
35. Koohfar, S.; Woldemariam, W.; Kumar, A. Prediction of Electric Vehicles Charging Demand: A Transformer-Based Deep Learning Approach. *Sustainability* **2023**, *15*, 2105. [CrossRef]
36. Li, C.; Qian, G. Stock Price Prediction Using a Frequency Decomposition Based GRU Transformer Neural Network. *Appl. Sci.* **2022**, *13*, 222. [CrossRef]
37. Ran, P.; Dong, K.; Liu, X.; Wang, J. Short-term load forecasting based on ceemdan and transformer. *Electr. Power Syst. Res.* **2023**, *214*, 108885. [CrossRef]
38. Lai, G.; Chang, W.C.; Yang, Y.; Liu, H. 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, Ann Arbor, MI, USA, 8–12 July 2018; pp. 95–104.
39. Dong, X.; Bao, J.; Chen, D.; Zhang, W.; Yu, N.; Yuan, L.; Chen, D.; Guo, B. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 12124–12134.