frontiers | Frontiers in Energy Research

TYPE Original Research
PUBLISHED 1 August 2024
DOI 10.3389/fenrg.2024.1429746

# Enhanced short-term flow prediction in power dispatching network using a transfer learning approach with GRU-XGBoost module ding

Zhe Ding[1]*, Tian Li[2], Xi'an Li[3] and Zhesen Cui[4]

[1]School of Computer Science, Queensland University of Technology, Brisbane, Australia, [2]Faculty of Engineering, Architecture and Information Technology, The University of Queensland, St Lucia, Australia, [3]Ceyear Technologies Co. Ltd, Qingdao, China, [4]Department of Computer Science, Changzhi University, Changzhi, China

OPEN ACCESS

EDITED BY
Shangce Gao,
University of Toyama, Japan

REVIEWED BY
Chaonan Zhang,
Zhejiang University, China
Vikram Kulkarni,
SVKM's Narsee Monjee Institute of Management Studies, India
Kenneth E. Okedu,
Melbourne Institute of Technology, Australia

*CORRESPONDENCE
Zhe Ding,
✉ zhe.ding@hdr.qut.edu.au

RECEIVED 08 May 2024
ACCEPTED 15 July 2024
PUBLISHED 1 August 2024

CITATION
Ding Z, Li T, Li X and Cui Z (2024), Enhanced short-term flow prediction in power dispatching network using a transfer learning approach with GRU-XGBoost module ding. *Front. Energy Res.* 12:1429746.
doi: 10.3389/fenrg.2024.1429746

The power dispatching network forms the backbone of efforts to automate and modernize power grid dispatching, rendering it an indispensable infrastructure element within the power system. However, accurately forecasting future flows remains a formidable challenge due to the network's intricate nature, variability, and extended periods of missing data resulting from equipment maintenance and anomalies. Vital to enhancing prediction precision is the interpolation of missing values aligned with the data distribution across other time points, facilitating the effective capture of nonlinear patterns within historical flow sequences. To address this, we propose a transfer learning approach leveraging the gated recurrent unit (GRU) for interpolating missing values within the power dispatching network's flow sequence. Subsequently, we decompose the generation of future flow predictions into two stages: first, extracting historical features using the GRU, and then generating robust predictions via eXtreme Gradient Boosting (XGBoost). This integrated process termed the GRU-XGBoost module, is applied in experiments on four flow sequences obtained from a power grid company in southern China. Our experimental findings illustrate that the proposed flow prediction model outperforms both machine learning and neural network models, underscoring its superiority in short-term flow prediction for power-dispatching networks.

KEYWORDS

**power dispatching network, transfer learning, neural networks, short-term flow forecasting, forecast**

## 1 Introduction

The power dispatching network (Wang et al., 2019) serves as the cornerstone for automating and modernizing power grid dispatching, ensuring safe, stable, and economical operation. It plays a pivotal role in coordinating the joint functioning of power generation, transmission, transformation, distribution, consumption, and other power system components, thereby guaranteeing the reliability and stability of the power grid (Peyghami et al., 2020). Given its vast scale, extensive data, and complex structure, effective management of the data flow within the power dispatching network is imperative (Yan et al., 2017; Li Z. et al., 2022). This study proposes modeling and predicting the power dispatching network flow to proactively identify evolving characteristics and trends. Such predictive capabilities facilitate the development of practical and efficient strategies for managing power dispatching network flow, mitigating network congestion, optimizing resource utilization, and meeting users' quality of service (QoS) requirements.

The processing and prediction of large-scale and complex network flow data in power-dispatching networks are paramount. However, the methods for forecasting flow data in power dispatching networks are scant compared to other forecasting fields within power systems, such as load forecasting (Wang et al., 2022) and wind energy forecasting (Zhao E. et al., 2022). This study scrutinizes existing time series forecasting methods and proposes enhancements to these methods (Tian et al., 2015; Cui et al., 2021), focusing on improving the dataset and refining the model perspective on time series forecasting, including machine learning and neural network models.

A primary constraint on existing time series forecasting methods is the dataset quality, particularly the flow sequence within the power dispatching network under scrutiny. In practical scenarios, unresolved power grid services or abnormal situations can lead to prolonged missing data in the flow time series (Wang et al., 2020), resulting in limited training samples and potential overfitting. Moreover, neural networks struggle to accurately model numerous consecutive missing values (Weerakody et al., 2021). Thus, addressing missing values in the flow sequence of power-dispatching networks is crucial to enhancing flow prediction. Various methods exist for handling missing values, including direct deletion, statistics-based filling methods, machine learning-based filling methods, and imputation using transfer learning (Adhikari et al., 2022). Direct deletion involves removing samples with missing values, risking the loss of crucial information. Meanwhile, statistics-based filling methods like mean filling, median filling, and common value filling use statistical metrics to replace missing data but disregard time series information. Machine learning-based methods encompass techniques such as K-nearest neighbors (KNN) (Chen, 2022; Keerin and Boongoen, 2022), Expectation-Maximization (EM) (Rahman and Islam, 2016; Deng et al., 2021), and Matrix Factorization (Rivera-Muñoz et al., 2022), while neural network-based approaches like Recurrent Neural Networks (RNN) (Turabieh et al., 2018; Kim and Chung, 2022) are also employed for filling missing values. Transfer learning imputes data by transferring annotated data or knowledge structures from related domains to enhance learning efficacy and fill missing values. For instance, Li et al. (2021) introduces a method called Transferring Long Short-Term Memory-based Iterative Estimation (TLSTM-IE) to address continuous missing values. Additionally, Yao and Zhao (2022) proposes a residual life prediction method based on deep timing feature migration.

Moreover, current research predominantly focuses on enhancing time series prediction through various model perspectives. A prevalent approach to improving prediction accuracy is through machine learning algorithm models, which effectively handle non-linearity, high dimensionality, and local minima (Cui et al., 2023; Wu et al., 2023). Common machine learning algorithms for time series prediction include linear regression, the gray model, SVR, and the XGBoost model. Numerous scholars have proposed combined models based on these algorithms (Wang et al., 2019; Chen et al., 2021; Wang and Bao, 2021; Li B. et al., 2022) to enhance predictive performance. For instance, SVR constructs a regression prediction model by minimizing the regression interval to fit a dataset. Zhao Z. et al. (2022) suggests an intelligent prediction method that optimizes support vector machine regression using a particle swarm algorithm, yielding improved prediction outcomes. Moreover, Han et al. (2014) employs a firefly algorithm to optimize the parameters of the SVR model, establishing a combined prediction model based on SVR for sample data prediction. Another widely used machine learning model for time series forecasting is the XGBoost model, known for regularization and reduction functions that mitigate model complexity, prevent overfitting, and handle missing values. Many researchers have applied the XGBoost model to enhance predictive accuracy in time series flow prediction (Hu and Li, 2021; Deng et al., 2022; Wen and Wang, 2022).

While machine learning models demonstrate efficacy in load forecasting and improving time series forecasting, they still face challenges in processing large volumes of network flow data efficiently. These models often lack robust handling of features over time scales. In contrast, neural networks excel at approximating complex nonlinear mappings and exhibit adaptability to dynamic characteristics and fault tolerance in uncertain systems, making them pivotal for constructing predictive models in nonlinear systems. Neural network models excel in learning intricate data distributions from vast datasets, serving as benchmark models for researchers seeking to enhance prediction accuracy and model fusion. For example, RNN, LSTM, and GRU are commonly employed in load forecasting. GRU, a variant of RNN, captures significant patterns in historical data and encodes them into a hidden state vector, elevating the quality of input feature vectors. Researchers have devised hybrid models based on GRU to achieve superior predictive performance (Jin et al., 2022; Lu et al., 2022; Yue and Liu, 2022). However, the majority of GRU-based power load forecasting models directly employ historical data for regressive future forecasting, performing feature extraction and regression modeling simultaneously. On the other hand, dividing the process of feature extraction and modeling into two steps may represent a potential direction to enhance forecasting effectiveness.

To address the aforementioned challenges, our study introduces a novel algorithm for power-dispatching network flow prediction, based on the GRU-XGBoost module, aimed at enhancing time series prediction. Unlike conventional approaches, our methodology employs transfer learning utilizing the GRU to interpolate flow sequences with missing values, facilitating training and prediction on complete datasets for imputation and forecasting. Moreover, the fusion of GRU and XGBoost modules represents a departure from sole reliance on GRU for sequence prediction. In our algorithm, GRU reconstructs features from the flow series, while XGBoost predicts post-reconstruction results, generating prediction sequences. Ultimately, this paper proposes a pioneering flow prediction algorithm for power dispatching networks based on the GRU-XGBoost module, presenting several innovative contributions:

1. A continuous missing value filling algorithm leveraging GRU-based transfer learning to establish a source domain module based on highly correlated flow sequences with the target sequence. This module is subsequently applied to the target

domain to fill continuous missing values, compensating for information loss resulting from continuous deletion in the original flow sequence.

2. The developed GRU-XGBoost block enhances the quality of input feature vectors, leading to more precise network flow predictions. Departing from conventional GRU forecasting approaches, this algorithm employs GRU for flow series feature reconstruction and XGBoost for time series prediction on reconstructed data, resulting in more accurate predictions and heightened forecasting precision.
3. Experimental validation on a dataset comprising power dispatching flow data (inflow and outflow flow of provincial access network and information region) from a southern Chinese power grid company demonstrates the superior prediction efficacy of the proposed algorithm compared to alternative models.

The remainder of this paper is organized as follows: In Section 2, we provide an exhaustive exposition of the foundational elements of our proposed algorithm, delineating the techniques of missing value imputation via transfer learning and the integration of GRU-XGBoost. Section 3 elaborates on the algorithm implementation procedures. Section 4 delineates the experimental design and the criteria employed for evaluation, and the proposed framework undergoes training and testing using flow datasets. Finally, Section 5 encapsulates the paper's denouement, presenting conclusive remarks and insights gleaned from the study.

# 2 Algorithmic model principles

In this section, we offer a comprehensive overview of the two pivotal components of our proposed algorithm: missing value imputation based on transfer learning, and GRU-XGBoost.

## 2.1 Missing value imputation based on transfer learning

Given the persistent presence of numerous missing values within the flow sequence of power dispatching networks, traditional methods such as linear interpolation and mean interpolation have yielded suboptimal results. Additionally, the reliance on boundary values for forecasting missing values across the entire time series has significantly affected performance, owing to the influence of neighboring values. In instances characterized by a high proportion of enduring missing values, resultant forecasts often demonstrate similar trends, leading to issues of flat prediction. The process of GRU-based transfer learning entails the selection and acquisition of the highly correlated flow sequence from the source domain, which is subsequently integrated into the target domain to address missing data. Specifically, the flow sequences for the source and target domains are represented as $\{x_j^{origin}|j = 1, 2, \ldots, N\}$ and $\{x_i^{target}|i = 1, 2, \ldots, N\}$, respectively.

The transfer learning process is illustrated in Figure 1. Initially, the GRU module is utilized to train the complete flow sequence, which exhibits a high correlation with the missing value sequence, resulting in the establishment of the source domain module $M_o$. Following the freezing of the neural network, the source domain module $M_o$ is employed to predict the missing value sequence. When migrating $M_o$ to the target domain for missing value imputation, it is essential to normalize the flow sequence of the source domain and the target domain to mitigate the impact caused by their inconsistent orders of magnitude. Additionally, there are often differences between the flow sequences of the source domain and the target domain to a certain degree after using the neural network model to build the source domain module $M_o$. Therefore, the tail fully connected layer of the source domain module $M_o$ needs fine-tuning based on the flow sequence of the target domain to obtain the module $M_t$ corresponding to the target domain. Given that the flow sequence of the power dispatching network is characterized by continuous missing values, a sliding window situation cycle should be employed to impute missing moments when utilizing the target domain module $M_t$ for missing value imputation. The training set part of the final imputation result represents the outcome of the transfer input, while the test set part denotes the prediction result of transfer learning.

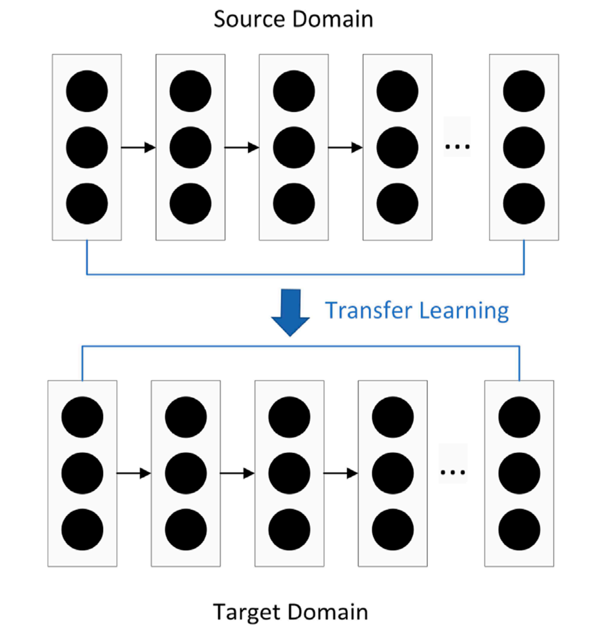

FIGURE 1
The process of transfer learning.

## 2.2 GRU-XGboost

The process of historical feature extraction from the flow sequence and subsequent future prediction generation is delineated into two distinct steps, facilitated by a GRU-XGBoost module tailored for each task.

For the flow sequence of the target domain $\{x_i^{target}|i = 1, 2, \ldots, N\}$, the inputs and outputs of feature reconstruction need to be constructed as follows:

$$\{(X_i, y_i)\,|\,i = 1, 2, 3, \ldots, N - p + 1\}. \quad (1)$$

where $X_i = \{x_i, x_{i+1}, \ldots, x_{i+p}\}$ is a historical input sequence for a set of feature reconstructions, $y_i = x_{i+p}$ is the corresponding label, $p$ is the length of the historical moment entered here. Eq. 1 indicates that the target prediction sample $x_{i+p+1}$ is only relevant to the historical sample of the previous $p$ moment, and the sample before the $p$ moment is approximately zero or uncorrelated with the target prediction sample $x_{i+p+1}$. The selection of $p$ can be determined by the partial auto-correlation coefficient and the number of moments with a confidence greater than 0.95 is selected as the value of $p$.

The GRU network is leveraged to extract crucial features from the historical sequence, encoding them into an implicit state vector. Within the GRU framework, a "gate" mechanism regulates the flow of information at each input moment, aiming to incorporate pertinent information into the state vector while filtering out less relevant data. For the input of feature reconstruction $X_i = \{x_i, x_{i+1}, \ldots, x_{i+p}\}$ and label $y_i$, the process of GRU reconstructing features is as follows:

$$z = \sigma\left(W_z x_i + U_z h_{i-1}\right),$$

$$r = \sigma\left(W_r + U_z h_{i-1}\right),$$

$$\widehat{h_i} = \tan h\left(W x_i + U(\mathrm{r} * h_{i-1})\right),$$

and

$$h_i = z h_{i-1} + (1-z)\widehat{h_i},$$

where $x_i$ represents the data sample of $i$-time power dispatching network flow, $h_{i-1}$and $h_i$is the previous and current hidden state respectively, $W_z$, $U_z$, $W_r$, $U_r$, $W$ and $U$ are the learned weight matrices, and $\sigma$ is the logistic sigmoid function. Now, for each group of input vectors $X_i$ and label $y_i$, we can obtain an improved feature vector $h$.

Upon obtaining the reconstructed features, this step involves utilizing these features for future prediction generation. Our approach integrates XGBoost, a widely adopted algorithm in both research and industrial settings, as the predictive module. XGBoost is employed to further refine feature importance assessment and model the reconstructed hidden features. Comprising $k$ regression tree models, XGBoost executes the prediction task through these models, culminating in the aggregation of predictions for the final output. The prediction process is expressed by the following formula:

$$\hat{y}_i^{(t)} = \sum_{k=1}^{K} f_k(x_i) = \hat{y}_i^{(k-1)} + f_k(x_i).$$

where $f_k(x_i)$ is the prediction of the $k$th regression tree model, $\hat{y}_i^{(k-1)}$ is the sum of predictions for the first $k-1$ regression trees.

To sum up, Figure 2 shows the overall algorithm flowchart of our proposed model. The GRU-XGBoost model is an integrated approach that combines the strengths of GRU for handling sequential data and XGBoost for robust prediction. Below is an elaborate step-by-step guide through the entire procedure of the model, from imputation of missing values in time series to the prediction phase.

**Step 1:** Data Preprocessing. The model begins with data preprocessing, where the time series is examined for missing values and normalized to ensure consistent scaling.

**Step 2:** Transfer Learning for Source Domain Establishment. A source domain module is created using GRU, trained on complete or less incomplete data segments, establishing a baseline for imputation.

**Step 3:** Target Domain Imputation. The trained GRU model is then adapted to the target domain, which contains the missing values. A fine-tuning process adjusts the model to fit the specific characteristics of the target domain, ensuring the imputed values are contextually appropriate.

**Step 4:** Feature Reconstruction. Post-imputation, the reconstructed time series is fed into the GRU-XGBoost module for feature extraction. The GRU component of the module processes the sequence to extract and reconstruct features that are indicative of future trends.

**Step 5:** XGBoost for Prediction. The extracted features are then passed to the XGBoost model, which is adept at capturing non-linear relationships and interactions among features. XGBoost builds an ensemble of decision trees, where each tree is added sequentially to minimize the loss function, enhancing the predictive accuracy.

**Step 6:** Evaluation and Iteration. The predictions are assessed using standard metrics such as MAE, MSE, and RMSE. The model's performance is iteratively improved by refining the imputation and prediction steps based on evaluation results.

## 3 Algorithm implementation

In addressing the processing and prediction challenges inherent in large-scale complex network flow data within a power dispatching network, we introduce an algorithm for flow prediction grounded in neural network principles and bolstered by boosting techniques. This approach aims to proactively discern the evolving characteristics and trends within the power-dispatching network flow. Consequently, we devise a pragmatic and efficient strategy for managing power-dispatching network flow, thereby meeting the quality of service (QoS) expectations of network users.

The prediction algorithm encompasses two key modules: the GRU-BIT module for missing value reconstruction via transfer learning, and the GRU-XGboost prediction module, which seamlessly integrates GRU and XGboost methodologies. Utilizing the GRU-BIT module for transfer learning facilitates the reconstruction, imputation, and prediction of randomly scattered missing values across the continuous spectrum of power-dispatching network flow sequences. This process significantly enhances the integrity of the power network traffic data. Furthermore, the GRU-XGboost module is tailored for feature reconstruction and subsequent prediction. In this framework, the GRU network undertakes the vital role of feature reconstruction within the flow sequence, while the XGboost module forecasts the future flow sequence of the power-dispatching network based on the reconstructed features. The sequential procedures involved in this algorithmic approach are elaborated below:

**Step 1:** Reconstructing missing values. Our approach initiates with the utilization of a continuous missing value-filling algorithm empowered by GRU-based transfer learning. This intricate process entails the reconstruction of a source domain module leveraging

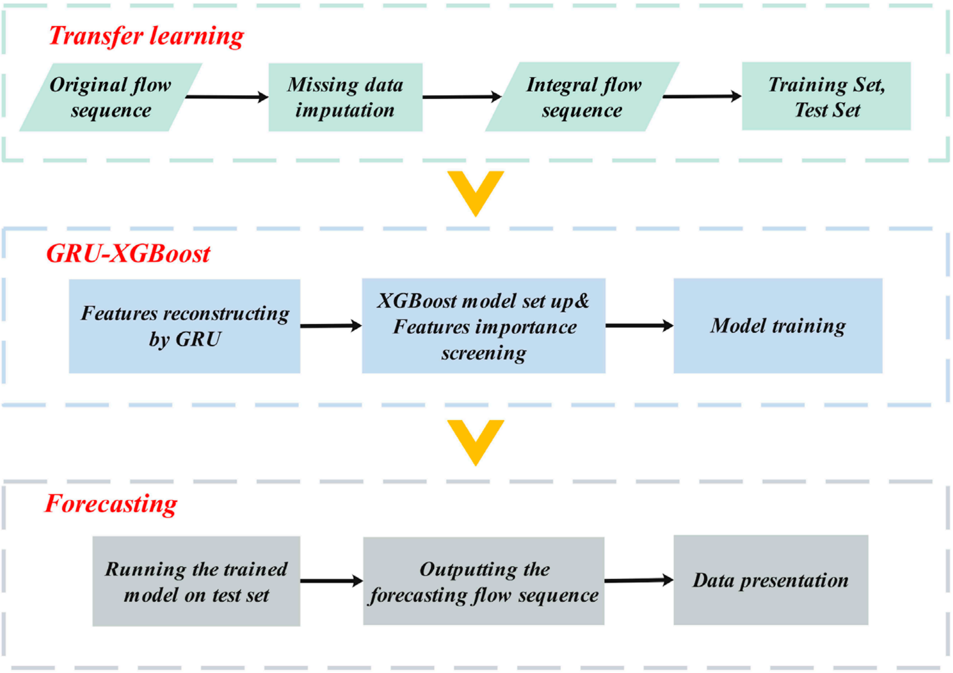

FIGURE 2
The flow chart of the proposed approach.

highly correlated flow sequences extracted from the target sequence. Subsequently, this meticulously crafted module is transposed into the target domain to rectify continuous missing values. Through this strategic maneuver, we effectively mitigate the information loss stemming from the continuous deletion inherent in the original flow sequence.

**Step 2:** Development of the GRU-XGboost module. To refine the prediction process, we introduce the GRU-XGboost module, designed to bifurcate prediction into two pivotal stages: feature extraction and prediction. Initially, GRU is employed to meticulously reconstruct features derived from historically related sequences. Following this, XGboost assumes the mantle to conduct an in-depth assessment of the significance of these features in the generation of future flow predictions.

**Step 3:** Deriving the ultimate power Dispatching Network Flow Prediction. Building upon the foundational steps above, we amalgamate the insights gleaned to derive the ultimate prediction for power-dispatching network flow.

# 4 Experimental analysis

The experiment endeavors to undertake short-term predictions of power dispatch network flow via a time series prediction algorithm. By augmenting the model's efficacy through the refinement of traditional time series prediction algorithms, optimization of network flow prediction accuracy is achieved. Subsequent to the collection and meticulous processing of power dispatch flow data, the model undergoes rigorous training, following which the trained model is deployed to forecast the test data. The predictive performance is then subjected to scrutiny and evaluation utilizing statistical metrics.

## 4.1 Experimental data

The dataset used for evaluation comprises power dispatch flow data obtained from a prominent power grid company in southern China. This dataset provides a higher-level perspective of the power flow, as it combines multiple sub-port flows. The target domain data focuses on the inflow and outflow flows of the Provincial Access Network and Information Region. Specifically, the dataset consists of four subsets named x_in, x_out, s_in, and s_out. To ensure the dataset's suitability for analysis, several preprocessing steps were undertaken. The temporal scope of the dataset spans from 28 December 2020, to 22 April 2021, with a granularity of 24 h. This time frame allows for a comprehensive examination of the power flow patterns. Our dataset covers the transition from winter to spring, a period marked by significant changes in temperature and daylight, which are known to influence power consumption patterns. The inclusion of data across weekdays, weekends, and public holidays further adds to the diversity, allowing our model to account for the variability in energy usage during regular and atypical days. This dataset provides a solid foundation for assessing the model's performance under diverse real-world conditions. During the dataset partitioning process, considerations such as computational resource consumption, time constraints, and model effectiveness were taken into account. As a result, the dataset was split into a training set, comprising the initial 75% of the data, and a test set, consisting of the remaining 25%. To gain a better

TABLE 1 Descriptive statistics for datasets.

| Dataset name | Dataset division | Sample size | Minimum | Maximum | Mean | Variance | Standard deviation |
|---|---|---|---|---|---|---|---|
| x_in | Training set | 2070 | 3.361E+06 | 6.658E+08 | 4.565E+08 | 4.745E+16 | 2.178E+08 |
| | Test set | 690 | 4.313E+08 | 6.008E+08 | 5.296E+08 | 4.825E+14 | 2.197E+07 |
| | Total dataset | 2,760 | 3.361E+06 | 6.658E+08 | 4.748E+08 | 3.670E+16 | 1.916E+08 |
| x_out | Training set | 2070 | 3.268E+06 | 8.607E+08 | 5.953E+08 | 7.638E+16 | 2.764E+08 |
| | Test set | 690 | 5.409E+08 | 8.562E+08 | 6.885E+08 | 2.194E+15 | 4.684E+07 |
| | Total dataset | 2,760 | 3.268E+06 | 8.607E+08 | 6.186E+08 | 5.945E+16 | 2.438E+08 |
| s_in | Training set | 2070 | 1.834E+05 | 3.635E+08 | 1.581E+08 | 2.593E+16 | 1.610E+08 |
| | Test set | 690 | 3.324E+06 | 3.714E+08 | 3.363E+08 | 5.371E+14 | 2.317E+07 |
| | Total dataset | 2,760 | 1.834E+05 | 3.714E+08 | 2.026E+08 | 2.553E+16 | 1.598E+08 |
| s_out | Training set | 1,632 | 2.400E+08 | 3.285E+08 | 2.989E+08 | 1.187E+14 | 1.089E+07 |
| | Test set | 1,224 | 2.400E+08 | 3.215E+08 | 2.983E+08 | 1.292E+14 | 1.137E+07 |
| | Total dataset | 408 | 2.466E+08 | 3.285E+08 | 3.006E+08 | 8.318E+13 | 9.120E+06 |

understanding of the dataset's characteristics, descriptive statistics were conducted. These statistics provided insights into the dataset's distribution, central tendency, and variability. Key data metrics extracted from the analysis are presented in Table 1, which aids in comprehensively assessing the dataset's features.

## 4.2 Evaluation criteria

To evaluate the prediction effect of each model, we introduce three evaluation indicators: Mean Absolute Error (MAE), Mean Squared Error (MSE), and Mean Absolute Percentage Error (MAPE).

MAE is calculated as:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|,$$

where $\hat{y}_i$ denotes the predicted value and $y_i$ denotes the true value. According to this equation, the MAE ranges from $[0,+\infty)$, which is equal to 0 when the predicted value exactly matches the true value, i.e., the perfect model; the larger the error, the larger the value.

MSE is calculated as:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2,$$

where $\widehat{y_i}$ denotes the predicted value and $y_i$ denotes the true value. The range is $[0,+\infty)$, which is equal to 0 when the predicted value exactly matches the true value, i.e., the perfect model; the larger the error, the larger the value.

MAPE is calculated as:

$$\mathrm{MAPE} = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{\widehat{y_i} - y_i}{y_i}\right|,$$

where $\widehat{y_i}$ denotes the predicted value and $y_i$ denotes the true value. The range is $[0,+\infty)$, MAPE of 0% indicates a perfect model, and MAPE greater than 100% indicates a poor model. a smaller MAPE value indicates a better model prediction.

## 4.3 Experimental settings

In this experiment, we established five comparison models, selecting typical models from various fields as control experiments to be compared with the optimized models, namely, multilayer perceptron (MLP), GRU, XGBoost, support vector regression (SVR), and Informer. These models span a broad spectrum of areas, including simple neural networks, recurrent neural networks, and machine learning, thereby showcasing the superiority of our models.

Following numerous adjustments and iterative validations, we identified the optimal parameters for each model to achieve relatively improved results. The parameters selected for optimal prediction performance in each model are as follows: For MLP, the activation function "relu" $(f(x) = \max(0,x))$ outperformed "identity" and "logistic" in actual predictions. Additionally, the default regularization parameter "alpha" was increased from 0.0001 to 0.01. In GRU, setting the time step "look_back" to 2 did not yield significant predictions, while values greater than 3 did not notably enhance prediction effectiveness. Consequently, we set "look_back" to 3, indicating that the first 3 inputs are used to predict the 4th. Furthermore, the 'epochs' parameter of GRU was set to 200 to minimize loss. For XGBoost, the booster parameter utilized the tree model gbtree as the base classifier. The L1 regularization parameter for weights was adjusted to 0.01 to enhance algorithm speed in high dimensions, and "look_back" exhibited optimal performance when set to 5. The SVR model employed a radial-basis function,

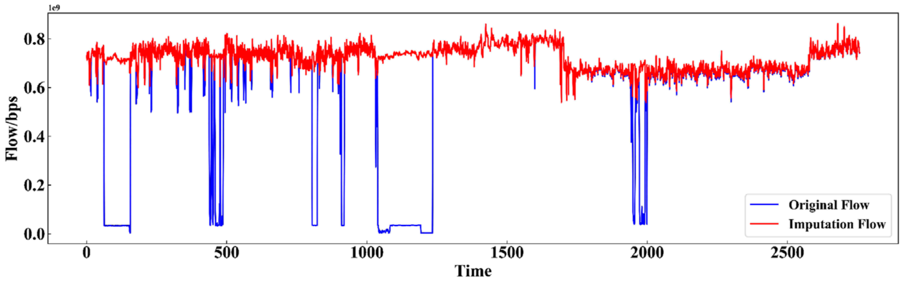

FIGURE 3
Transfer learning result of the x_out dataset.

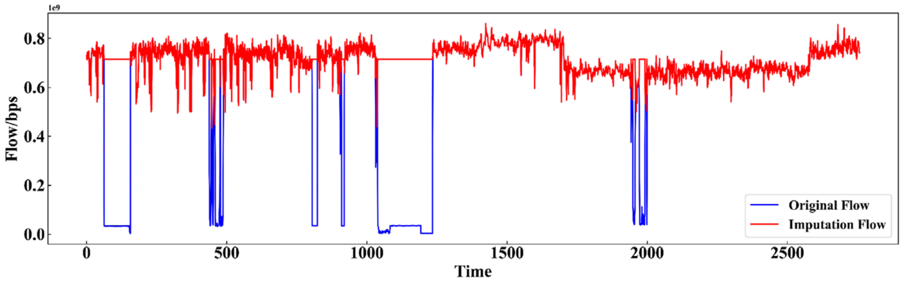

FIGURE 4
Matrix completion result of the x_out dataset.

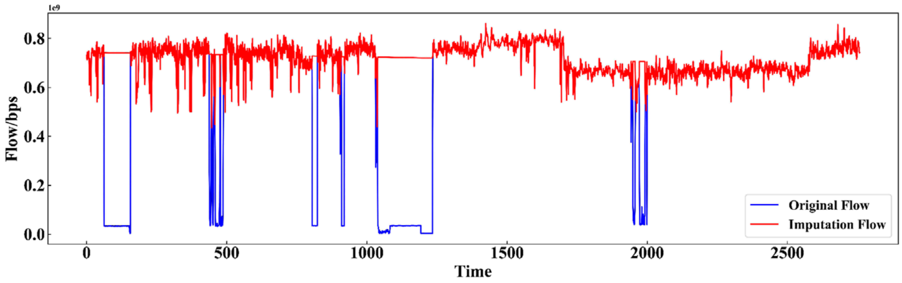

FIGURE 5
Gaussian process regressor result of the x_out dataset.

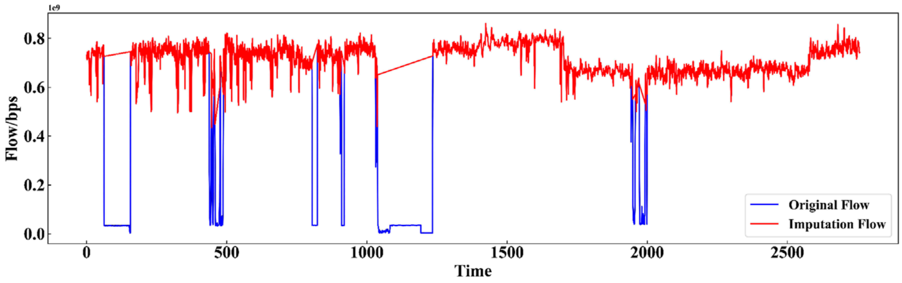

FIGURE 6
Linear imputation result of the x_out dataset.

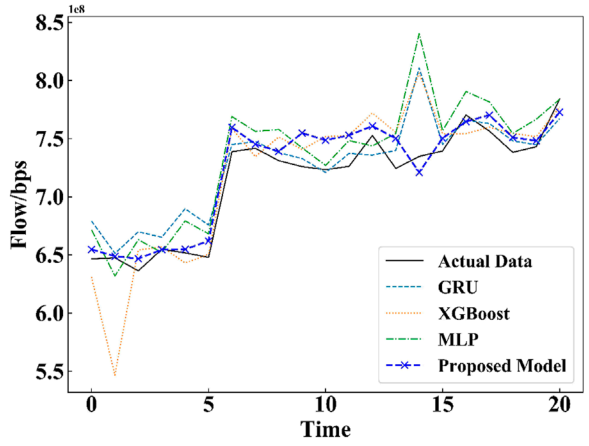

FIGURE 7
Prediction effect of each model.

which maps the original space to an infinite-dimensional space and demonstrates superior performance across large and small sample sizes. Compared to "linear", the "sigmoid" function exhibited significantly enhanced predictive effects on the flow series. The relevant parameters for the Informer model align with the settings described in the original paper.

## 4.4 Analysis of missing value filling

We initially visualized the original dataset and observed a significant drop in some data points from the images. This drop was attributed to anomalous data during flow monitoring, prompting us to conduct a quantitative assessment of this phenomenon.

To address the eliminated missing data, we employed transfer learning to interpolate them, presenting a comparison between the original data and the differentially complemented data (exemplified by x_out) in Figure 3. The original data is denoted in blue, while the interpolated data is highlighted in red. Observing the original flow sequence reveals a substantial presence of missing values persisting over a considerable duration. Leveraging transfer learning via the GRU model entails identifying relevant source domain flow sequences from the original flow data. By assimilating insights from these sequences, the model adeptly fills in missing segments within the target domain's flow sequence, resulting in the generation of a comprehensive interpolated flow series. The interpolated flow series illustrated in Figure 3 exhibits a coherent pattern, thereby enhancing the precision of network flow prediction endeavors.

As shown in Figures 3–6, we have now included a direct comparison between our proposed GRU-based transfer learning method, matrix completion method, probabilistic model, and linear interpolation on the same set of time series data with missing values. The superiority of our GRU-based approach can be attributed to its ability to capture the underlying patterns and dependencies within the data that linear interpolation, being a simpler method, does not take into account.

TABLE 2 The evaluation indicators of each mode.

| Dataset | Algorithm | MAE | MAPE | RMSE |
|---|---|---|---|---|
| x_in | Optimized algorithm | 1.49E+07 | 2.84E-02 | 1.96E+07 |
| | MLP | 1.52E+07 | 2.89E-02 | 2.02E+07 |
| | GRU | 1.52E+07 | 2.94E-02 | 1.96E+07 |
| | XGBoost | 1.52E+07 | 2.88E-02 | 1.97E+07 |
| | SVR | 1.48E+07 | 2.85E-02 | 1.91E+07 |
| | Informer | 1.50E+07 | 2.86E-02 | 1.98E+07 |
| x_out | Optimized algorithm | 1.86E+07 | 2.71E-02 | 2.48E+07 |
| | MLP | 1.99E+07 | 2.88E-02 | 2.61E+07 |
| | GRU | 1.88E+07 | 2.77E-02 | 2.45E+07 |
| | XGBoost | 1.95E+07 | 2.85E-02 | 2.63E+07 |
| | SVR | 2.19E+07 | 3.24E-02 | 2.77E+07 |
| | Informer | 1.85E+07 | 2.69E-02 | 2.46E+07 |
| s_in | Optimized algorithm | 3.80E+06 | 1.13E-02 | 5.54E+06 |
| | MLP | 4.02E+06 | 1.21E-02 | 6.72E+06 |
| | GRU | 5.10E+06 | 1.52E-02 | 7.62E+06 |
| | XGBoost | 3.95E+06 | 1.19E-02 | 6.69E+06 |
| | SVR | 1.11E+07 | 3.28E-02 | 1.23E+07 |
| | Informer | 4.41E+06 | 1.32E-02 | 8.99E+06 |
| s_out | Optimized algorithm | 4.52E+06 | 1.51E-02 | 6.87E+06 |
| | MLP | 4.34E+06 | 1.46E-02 | 6.79E+06 |
| | GRU | 5.83E+06 | 1.94E-02 | 7.74E+06 |
| | XGBoost | 5.13E+06 | 1.72E-02 | 8.30E+06 |
| | SVR | 1.05E+07 | 3.12E-02 | 1.19E+07 |
| | Informer | 5.15E+06 | 1.72E-02 | 9.16E+06 |

## 4.5 Experimental results

Figure 7 illustrates the prediction performance of each model, utilizing the x_out dataset as a demonstration. The test set comprises approximately 700 data points. For clarity, Figure 7 presents the prediction performance for a selected subset of 21 data points. A comparison between the prediction results of each model and the actual values reveals the superior flow prediction performance of the optimized model proposed in this study.

Table 2 exhibits the three evaluation metrics for the prediction performance of power network flow for each model. To facilitate a comprehensive comparison, four sets of flow data are selected

TABLE 3 The R-squared and standard deviation evaluation results.

| | GRU | MLP | SVR | XGBoost | Optimized algorithm |
|---|---|---|---|---|---|
| R-squared | 3.48E+07 | 4.58E+07 | 3.38E+07 | 4.46E+07 | 4.12E+07 |
| standard deviation | 0.6986 | 0.7524 | 0.7366 | 0.7159 | 0.7547 |

TABLE 4 With and without transfer learning.

| Dataset | Algorithm | MAE | MAPE | RMSE |
|---|---|---|---|---|
| x_in | Without Transfer Learning | 1.52E+07 | 2.85E-02 | 1.94E+07 |
| | With Transfer Learning | 1.49E+07 | 2.84E-02 | 1.96E+07 |
| x_out | Without Transfer Learning | 1.94E+07 | 2.91E-02 | 3.25E+07 |
| | With Transfer Learning | 1.86E+07 | 2.71E-02 | 2.48E+07 |
| s_in | Without Transfer Learning | 5.54E+06 | 1.63E-02 | 1.02E+07 |
| | With Transfer Learning | 3.80E+06 | 1.13E-02 | 5.54E+06 |
| s_out | Without Transfer Learning | 8.62E+06 | 2.87E-02 | 2.34E+07 |
| | With Transfer Learning | 4.52E+06 | 1.51E-02 | 6.87E+06 |

TABLE 5 Before and after GRU Reconfiguration.

| Dataset | Algorithm | MAE | MAPE | RMSE |
|---|---|---|---|---|
| x_in | Without GRU | 1.50E+07 | 2.88E-02 | 1.87E+07 |
| | With GRU | 1.49E+07 | 2.84E-02 | 1.96E+07 |
| x_out | Without GRU | 2.13E+07 | 3.11E-02 | 2.63E+07 |
| | With GRU | 1.86E+07 | 2.71E-02 | 2.48E+07 |
| s_in | Without GRU | 3.95E+06 | 1.17E-02 | 5.39E+06 |
| | With GRU | 3.80E+06 | 1.13E-02 | 5.54E+06 |
| s_out | Without GRU | 4.35E+06 | 1.46E-02 | 6.66E+06 |
| | With GRU | 4.52E+06 | 1.51E-02 | 6.87E+06 |

to assess the prediction effects in the result analysis of this paper. The evaluation metrics indicate that the proposed optimization model demonstrates the best prediction performance, followed by the MLP and Informer models. Among the remaining five comparison models, the Informer model stands out due to its ability to capture relevant temporal features. Despite being a relatively recent prediction model, it performs admirably. Conversely, while the SVR model exhibits lower computational complexity during the prediction process, its prediction effect in this experiment is subpar. This suggests that the SVR model is not well-suited for direct power dispatch network flow prediction.

Table 3 exhibits the two evaluation metrics for the prediction performance of power network flow in x_out dataset. These results indicate the proposed model offers a balance between a high R-squared value and a lower standard deviation, indicating a more consistent prediction performance with less variability around the mean.

Expanding on conventional prediction methodologies, this study introduces a model integrating a transfer learning interpolation module and GRU feature reconstruction to augment prediction efficacy. To assess the efficacy of these components, the model's prediction performance is evaluated with and without the inclusion of each module. Table 4 juxtaposes the evaluation metrics pre- and post-incorporation of the transfer learning module. The findings underscore a notable decline in prediction accuracy in the absence of the transfer learning module, underscoring its constructive influence in processing consecutive missing values within original power network flow sequences to bolster predictive outcomes.

Likewise, Table 5 delineates the distinction between direct GRU-based prediction and prediction subsequent to GRU-based feature reconstruction across identical datasets. Performance differentials manifest contingent upon the datasets analyzed. Nonetheless, across x_in, x_out, and s_in datasets, employing GRU-based feature reconstruction consistently yields superior predictive outcomes compared to direct GRU-based prediction. This corroborates the pivotal role of the GRU feature reconstruction module in enhancing prediction accuracy.

## 4.6 Further discussions

This experiment aims to enhance prediction accuracy by optimizing and integrating existing time-series prediction models to develop a high-performance model for power network flow prediction. The proposed algorithm in this study has several strengths that make it an ideal model for power network flow prediction. By optimizing and integrating existing time-series prediction models, the algorithm achieves high prediction accuracy and improved fidelity of experimental data. The use of neural networks and machine learning models offers a comprehensive comparative analysis that evaluates the performance of the optimized model using robust metrics. The removal of outliers from the original dataset ensures that the experimental data is reliable and accurate. The implementation of transfer learning for data interpolation further enhances the prediction performance of the model, and the impact of integrating this module is thoroughly examined.

Furthermore, the algorithm incorporates GRU feature reconstruction, which contributes to the refinement of prediction accuracy. This technique allows the model to capture and utilize long-term dependencies in the data, leading to more accurate predictions. The comparison of actual and predicted values demonstrates the superior performance of our optimized model compared to existing models. This showcases the algorithm's refined predictive capabilities and highlights its potential for real-world applications.

Our GRU-XGBoost model, while more sophisticated, does entail higher computational demands due to the complexity of processing sequential data and the ensemble learning approach of XGBoost. Specifically, the model's training phase requires additional computational time and memory to handle the dynamic nature of GRU units and the optimization process of XGBoost. While our GRU-XGBoost model demands more computational resources compared to simpler alternatives, the benefits in terms of predictive accuracy and the ability to capture complex temporal patterns outweigh these costs.

Power dispatching companies can harness the GRU-XGBoost model to significantly upgrade their operational forecasting capabilities. By integrating this model into their existing systems, these companies can achieve more accurate short-term flow predictions. This enhanced foresight allows for proactive grid management, better resource allocation, and the preemptive mitigation of potential congestions or supply deficiencies. The model's ability to handle missing data and capture complex patterns ensures that predictions are reliable even with incomplete information, which is a common challenge in real-time data from power networks.

Despite its strengths, the proposed algorithm also has some limitations to consider. One limitation is that the performance of the algorithm may be influenced by the quality and availability of input data. Factors such as data completeness, missing values, or noisy data can affect the accuracy of predictions. Additionally, the computational requirements of the algorithm may be high due to the complexity of the model, which can limit its practicality in certain settings.

The deployment of the GRU-XGBoost model in real-world scenarios, while promising, is not without challenges. One of the primary challenges is the requirement for high-quality, consistent data to train and refine the model effectively. Data scarcity or irregularities can impede the model's predictive accuracy. Additionally, the computational intensity of training the model, particularly for large-scale grids, may demand substantial resources. There is also the challenge of model interpretability; given the complexity of the GRU-XGBoost architecture, translating the model's predictions into actionable insights could be non-trivial for practitioners. Lastly, the model's sensitivity to hyperparameters necessitates careful tuning to adapt to the specific characteristics of different power networks.

## 5 Conclusion

In the context of power dispatching network flow prediction, this paper presents a novel algorithm for power dispatching network flow prediction that leverages neural networks and boosting techniques. By applying transfer learning to interpolate the original flow sequence and constructing a GRU module for feature reconstruction, we have substantially improved the accuracy of power network flow predictions. We conducted a comprehensive comparative analysis, which included MLP, SVR, XGBoost, GRU, and Informer models, and evaluated their performance on four datasets obtained from the Information Region and Provincial Access Network. Our proposed model proved highly effective in generating precise power network flow sequence predictions, as validated by the findings of this study. Moving forward, further strategies are required to mitigate the residuals of the GRU-XGBoost module and reduce the impact of random noise on prediction outcomes. Possible approaches include exploring alternative network architectures or optimizing the hyperparameters of the existing model to enhance its robustness. Additionally, incorporating more diverse and complex data inputs may lead to more accurate predictions and improve the generalizability of the model. By continuing to refine and advance our approach, we can contribute to the ongoing progress in power dispatching network flow prediction.

## Data availability statement

The data analyzed in this study is subject to the following licenses/restrictions: This data can be obtained with a reasonable request. Requests to access these datasets should be directed to cuizhesen@163.com.

## Author contributions

ZD: Conceptualization, Formal Analysis, Methodology, Writing–original draft. TL: Data curation, Formal Analysis, Writing–review and editing. XL: Writing–review and editing. ZC: Data curation, Methodology, Resources, Writing–original draft.

## Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This work was supported by the Fundamental Research Program of Shanxi Province (202303021222271), Scientific and Technological Innovation Programs of Higher Education Institutions in Shanxi, PR China (2022L517).

## Conflict of interest

Author XL was employed by Ceyear Technologies Co. Ltd.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## References

Adhikari, D., Jiang, W., Zhan, J., He, Z., Rawat, D. B., Aickelin, U., et al. (2022). A comprehensive survey on imputation of missing data in internet of things. *ACM Comput. Surv.* 55, 1–38. doi:10.1145/3533381

Chen, J., Mao, W., Liu, J., and Zhang, X. (2021). Remaining useful life prediction of bearing based on deep temporal feature transfer. *Control Decis.* 36, 1699–1706.

Chen, X. (2022). Research on an algorithm for filling missing values by k-nearest neighbors with optimized weights. *Wirel. Internet Technol.* 19, 5.

Cui, Z., Wu, J., Ding, Z., Duan, Q., Lian, W., Yang, Y., et al. (2021). A hybrid rolling grey framework for short time series modelling. *Neural Comput. Appl.* 33, 11339–11353. doi:10.1007/s00521-020-05658-0

Cui, Z., Wu, J., Lian, W., and Wang, Y.-G. (2023). A novel deep learning framework with a covid-19 adjustment for electricity demand forecasting. *Energy Rep.* 9, 1887–1895. doi:10.1016/j.egyr.2023.01.019

Deng, X., Ye, A., Zhong, J., Xu, D., Yang, W., Song, Z., et al. (2022). Bagging–xgboost algorithm based extreme weather identification and short-term load forecasting model. *Energy Rep.* 8, 8661–8674. doi:10.1016/j.egyr.2022.06.072

Deng, Z., Tang, Z., Zhu, H., and Zhao, Y. (2021). An improved expectation maximization algorithm for missing data management of concrete pump truck. *J. Central South Univ.* 318, 443.

Han, M., Xu, M., and Mu, D. (2014). Relevance vector machine with reservoir for time series prediction. *Chin. J. Comput.* 37, 2427–2432.

Hu, Y., and Li, Y. (2021). Power load forecasting based on pso-svr. *J. Jiamusi Univ.* 39, 54–57.

Jin, B., Zeng, G., Lu, Z., Peng, H., Luo, S., Yang, X., et al. (2022). Hybrid lstm–bpnn-to-bpnn model considering multi-source information for forecasting medium- and long-term electricity peak load. *Energies* 15, 7584. doi:10.3390/en15207584

Keerin, P., and Boongoen, T. (2022). Improved knn imputation for missing values in gene expression data. *Comput. Mater. Continua* 70, 4009–4025. doi:10.32604/cmc.2022.020261

Kim, J., and Chung, K. (2022). Recurrent neural network-based multimodal deep learning for estimating missing values in healthcare. *Appl. Sci.* 12, 7477. doi:10.3390/app12157477

Li, B., Zhang, S., Zheng, S., and Zhao, Z. (2022a). Research on hydropower power prediction based on multiple linear regression and arima combination model. *Sci. Technol. Innovation* 33, 71–74.

Li, Y., Bao, T., Chen, H., Zhang, K., Shu, X., Chen, Z., et al. (2021). A large-scale sensor missing data imputation framework for dams using deep learning and transfer learning strategy. *Measurement* 178, 109377. doi:10.1016/j.measurement.2021.109377

Li, Z., Xu, Z., Yu, S., Chen, J., Yang, M., and Liu, Y. (2022b). Reliability evaluation power dispatching data network. *Front. Data Domputing* 4, 87.

Lu, W., Zheng, R., Zhao, W., and Tang, J. (2022). Short-term power load forecasting method based on emd-gru. *Electr. Drive* 52, 74–80.

Peyghami, S., Palensky, P., and Blaabjerg, F. (2020). An overview on the reliability of modern power electronic based power systems. *IEEE Open J. Power Electron.* 1, 34–50. doi:10.1109/ojpel.2020.2973926

Rahman, G., and Islam, M. Z. (2016). Missing value imputation using a fuzzy clustering-based em approach. *Knowl. Inf. Syst.* 46, 389–422. doi:10.1007/s10115-015-0822-y

Rivera-Muñoz, L., Giraldo-Forero, A., and Martinez-Vargas, J. (2022). Deep matrix factorization models for estimation of missing data in a low-cost sensor network to measure air quality. *Ecol. Inf.* 71, 101775. doi:10.1016/j.ecoinf.2022.101775

Tian, Z., Li, S., Wang, Y., and Gao, X. (2015). Network traffic prediction based on empirical mode decomposition and time series analysis. *Control Decis.* 30, 905.

Turabieh, H., Salem, A. A., and Abu-El-Rub, N. (2018). Dynamic l-rnn recovery of missing data in iomt applications. *Future Gener. Comput. Syst.* 89, 575–583. doi:10.1016/j.future.2018.07.006

Wang, H., Alattas, K. A., Mohammadzadeh, A., Sabzalian, M. H., Aly, A. A., and Mosavi, A. (2022). Comprehensive review of load forecasting with emphasis on intelligent computing approaches. *Energy Rep.* 8, 13189–13198. doi:10.1016/j.egyr.2022.10.016

Wang, J. Q., Du, Y., and Wang, J. (2020). Lstm based long-term energy consumption prediction with periodicity. *energy* 197, 117197. doi:10.1016/j.energy.2020.117197

Wang, R., Zhang, Y., Wang, D., Zhang, T., and Liu, Y. (2019). Optimization and scheduling of power system stochastic model predictive control based optimization and scheduling for power system with large scale wind integrated. *Control Decis.* 34, 1616–1625.

Wang, S., and Bao, C. (2021). Research on the application of intelligent algorithms in power grid load forecasting. *J. Anhui Polytech. Univ.* 36, 8.

Weerakody, P. B., Wong, K. W., Wang, G., and Ela, W. (2021). A review of irregular time series data handling with gated recurrent neural networks. *Neurocomputing* 441, 161–178. doi:10.1016/j.neucom.2021.02.046

Wen, Y., and Wang, W. (2022). Short-term power load forecasting based on the combined fa-svr-lstm model. *Electron. Technol.* 1. doi:10.16180/j.cnki.issn1007-7820.2023.09.001

Wu, Q., Li, J., Liu, Z., Li, Y., and Cucuringu, M. (2023). Symphony in the latent space: provably integrating high-dimensional techniques with non-linear machine learning models. *Proc. AAAI Conf. Artif. Intell.* 37, 10361–10369. doi:10.1609/aaai.v37i9.26233

Yan, Z., Zhu, G., Liu, X., and Sun, P. (2017). "Research on application of sdn in dispatching system of power grid," in Proceedings of the 2017 6th International Conference on Energy, Environment and Sustainable Development (ICEESD 2017) (Academic Press), 1003–1010.

Yao, Z., and Zhao, C. (2022). Fedtmi: knowledge aided federated transfer learning for industrial missing data imputation. *J. Process Control* 117, 206–215. doi:10.1016/j.jprocont.2022.08.004

Yue, W., and Liu, Q. (2022). Multi-energy load forecasting for integrated energy systems based on pcclstmmtl. *J. Shanghai Univ. Electr. Power* 38, 483–487.

Zhao, E., Sun, S., and Wang, S. (2022a). New developments in wind energy forecasting with artificial intelligence and big data: a scientometric insight. *Data Sci. Manag.* 5, 84–95. doi:10.1016/j.dsm.2022.05.002

Zhao, Z., Zhang, Y., Yang, Y., and Yuan, S. (2022b). Load forecasting via grey model-least squares support vector machine model and spatial-temporal distribution of electric consumption intensity. *Energy* 255, 124468. doi:10.1016/j.energy.2022.124468