A Unified Model for Smart Meter Data Applications

Zhenyi Wang, *Graduate Student Member, IEEE,* Hongcai Zhang, *Senior Member, IEEE,* Geert Deconinck, *Senior Member, IEEE,* and Yonghua Song, *Fellow, IEEE*

***Abstract*—Making adequate utilization of smart meter data is conducive to improving the energy efficiency of the power system from demand side, especially with booming artificial intelligence (AI) technology. However, most existing AI-based methods are highly incompatible to each other due to unique designs based on their respective tasks. Low compatibility will lead to duplicate modeling among similar tasks and skyrocketing implementation costs, which is not suitable for diverse and changing demand-side tasks. Although large language models provide a promising way to build the general-purpose models, they either need substantial resources for pre-training or case-by-case design for fine-tuning. Hence, there are practically rare task-generic models available for power systems. In this paper, we propose a novel unified model for smart meter data applications. Specifically, we first propose a unified model with mixture-of-expert layers to ensure sufficient model capacity in a cost-effective manner, which makes the training from scratch affordable. Then, we design an information bottleneck-based training scheme to facilitate the unified model to efficiently learn the generic knowledge. Moreover, we develop a general framework based on pre-training paradigm to formulate a uniform objective function and provide a consistent workflow for different tasks. Finally, the effectiveness and superiority of our proposed method are validated on public datasets, where the proposed unified model can be applied to load forecasting, data imputation as well as anomaly detection, and realizes comparable performance to state-of-the-art task-specific methods.**

***Index Terms*—Demand-side task, information bottleneck, mixture of expert, smart meter data, unified model**

I. Introduction

TOWARDS the low-carbon power systems, the advanced metering infrastructure represented by smart meters plays an indispensable role [1]. Smart meters facilitate the two-way communication between electric utilities and customers, which improves the efficiency of system operation and control. By the end of 2023, the smart meter installations in the United States, Europe, and China have exceeded 128 million, 186 million, and 650 million, respectively [2]. The widely deployed smart meters produce the sheer amount of fine-grained electricity consumption data, which encompasses a wealth of information on electricity consumption behaviors of electric customers. With the deregulation of the power industry, analyzing smart meter data can provide valuable insights for electric utilities and customers to reduce energy costs. Meanwhile, in response to the penetration of distributed energy resources, smart meter data analytics can also promote the consumption of renewable generation and maintain the system balance through demand response. Therefore, how to apply smart meter data to improve energy efficiency and grid sustainability from the demand side is becoming an important and promising topic.

In the past decade, with the boom in artificial intelligence (AI), research on AI-based smart meter data applications has emerged explosively. AI technology, especially deep learning, is able to effectively process and model the complex temporal dependencies of smart meter data, and therefore has achieved significant success in smart meter data applications [3]. For example, Ruan et al. [4] proposed a spatiotemporal graph deep learning-based method to detect cyberattacks using electricity load data. Powell et al. [5] presented a charging demand model based on hybrid methods to advise policymakers on adjusting utility rates and charging infrastructure, by historical charging load data. Wang et al. [6] developed a federated learning-based framework to evaluate regulation capacity using realistic baseline load data from smart meters. In a nutshell, massive AI technologies have been applied to various demand-side tasks, such as load forecasting, customer categorization, and market bidding, and have realized performance improvements [7].

However, existing studies on AI-based smart meter data applications face a serious challenge, i.e., incompatibility [8]. To be specific, although these studies are all oriented towards smart meter data applications, their proposed models can only be applicable to their respective tasks, because each method is specially designed for the corresponding task. For instance, the AI-based methods proposed for load forecasting usually can not be used to accomplish the baseline load estimation task. To make matters worse, even for different tasks in the same type, the proposed models also suffer from mutual incompatibility, such as day-ahead and hour-ahead load forecasting models. This may be owing to the different formats of model input and output, as well as the distinct focuses of different tasks. In this way, the incompatibility of AI-based methods will lead to the high implementation cost of smart meter data applications, especially for electric utilities. This is because that there are plentiful tasks with diverse types in the demand side that require electric utilities to analyze and apply smart meter data, e.g., providing quality service to customers or making profits from the electricity markets [7]. In this way, electric utilities need to build separate application models for each task, and undertake high costs in model training due to the large number of tasks. In addition, with the deregulation of the demand side, there is a foreseeable growth in the type and number of demand-side tasks [3], which will further exacerbate the costs of electric utilities for building AI-based methods.

To resolve the above problem, building a task-generic model

This paper is funded in part by the Science and Technology Development Fund, Macau SAR (File no. 001/2024/SKL, and File no. 0053/2022/AMJ) (Corresponding author: *Hongcai Zhang*.)

Z. Wang, H. Zhang and Y. Song are with the State Key Laboratory of Internet of Things for Smart City and Department of Electrical and Computer Engineering, University of Macau, Macao, 999078 China (email: hczhang@um.edu.mo).

G. Deconinck is with the Department of Electrical Engineering (ESAT), KU Leuven, 3001 Leuven, Belgium (email: geert.deconinck@kuleuven.be).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSG.2024.3452490.

rather than multiple task-specific models may be a feasible approach. The AI community has pursued the development of unified models capable of fulfilling multiple tasks. Generally, unified models can be applied to diverse tasks with little or no additional training [9], which is ideal for electric utilities to conduct smart meter data applications. On the one hand, the unified models eliminate the need to train several task-specific models, thus effectively reducing implementation costs. Moreover, the general purpose of unified models is also conducive to handling new tasks that may arise in the future. On the other hand, unified models avoid preparing high-quality training data for each task separately, thus alleviating the expense of dataset construction. Therefore, the unified model of smart meter data applications is very necessary for electric utilities.

However, general-purpose unified models for smart meter data applications remain unexplored. Considering that smart meter data are typical time series, we expect to gain inspiration from research on unified models for time series data, which is in full swing. Generally speaking, there are two main streams in existing research, depending on whether large language models (LLMs) [10] are involved. For the LLM-based method, the intuitive idea is to build unified models like ChatGPT from scratch for smart meter data applications. Although the model capability of GPT family has been well proven to accomplish different tasks, the resources required for model training are too many to meet in practice. According to statistics [11], the training cost of GPT-4 in 2023 is estimated to be 78 million USD, which is beyond the affordability of common electric utilities. In addition, some studies point out that the benefits of LLMs may not be as great as the burdens they impose from the energy perspective [12]. Therefore, this start-from-scratch approach is unwise and unrealistic for electric utilities.

To avoid training LLM-based models from zero, researchers intend to reprogram existing LLMs in other domains to build unified models for time series data. With the power of LLMs, such unified models can still cope with different tasks, and the implementation costs are also reduced to an affordable level [10]. Recently, there has been an influx of work utilizing LLMs in power systems. For example, Huang et al. [13] developed the systematic pipelines based on GPT-4 to fulfill the optimal power flow and electric vehicle scheduling tasks. Mongaillard et al. [14] designed a novel user-centric architecture for power resource scheduling tasks by constructing three LLM-based agents. Jia et al. [15] proposed a modular framework based on expertise from power systems and LLM domains, to promote LLMs to perform power system simulations. However, these methods require either adding model structure or constructing prompts for LLMs, which still need expertise and experience to design according to specific tasks. This will impose a heavy burden on practical implementation. Furthermore, applying the existing LLMs to power systems may pose potential security threats [16], e.g., privacy leakage and cyber attacks. Therefore, this LLM-reprogramming manner is still not appropriate for developing unified models for electric utilities that have high risk-awareness and need to deal with numerous tasks.

In addition to LLM-based methods, there have been efforts to build unified models by developing novel neural network architectures. For instance, Wu et al. [17] proposed a new task-general model for time series data analysis, which exploits multiple levels of frequency-based features obtained through Fourier transform to capture complex temporal signals. Zhao et al. [18] proposed a novel generic model based on multitask reinforcement learning for large distribution network operation to perform distinct tasks separately. Even though these studies can be valid for multiple tasks, there are still some limitations: 1) need to build individual models for each task [17]; 2) need under an invariant environment [18]. However, electric utilities usually face a wide range of different tasks in a complex and changing environment [19]. This leads to the need of a separate model for each environment and task, which deviates from the intention of building unified models. Therefore, the existing non-LLM methods are also not suitable for electric utilities.

In summary, existing general methods on time series data do not fit electric utilities to build unified models for smart meter data applications. In particular, electric utilities usually only have limited resources and capabilities. Hence, they face two main challenges when developing AI-based unified models as:

- Challenge 1: To sufficiently learn generic knowledge, the unified model needs to have a high model capacity, which will correspond to a heavy computational cost in model training. Meanwhile, generic knowledge is often learned through the extensive training to avoid omitting valuable information, thus requiring abundant computing and data resources. However, electric utilities usually have limited model training budget, so they might not be able to afford the training expense of conventional AI-based models.
- Challenge 2: To accomplish multiple tasks of different types, the unified model needs to be compatible with different input and output formats, which ordinarily requires additional adaptations to meet the specific requirements of diverse tasks. However, electric utilities lack adequate experience and capability to realize the task-by-task design for different and growing tasks, because their main business is the power system field rather than AI.

To address these aforesaid challenges, we propose a general-purpose unified model for smart meter data applications without relying on LLMs, which can accomplish multifarious tasks on the demand side for electric utilities. Specifically, we first propose a unified model based on the mixture-of-expert layers, which is capable of boosting the model capacity to capture the temporal dependencies of smart meter data with a small computational cost. Then, we design a new training scheme with information bottleneck theory, which enables the unified model to learn sufficient generic knowledge in an efficient manner. Finally, we develop a pre-training-based framework to make the proposed unified model able to carry out different applications with a consistent workflow. To the best of our knowledge, we are the first to develop the task-generic model for smart meter data applications. Compared to the published literature, our key contributions are summarized in threefold:

1) We propose a unified model based on mixture-of-expert layers, which can be applicable to multiple smart meter data applications, rather than being task-specific. Unlike most existing models with a linear relation between the capacity and cost, the proposed unified model leverages

mixture-of-expert layers to acquire high model capacity at a lower computational cost. This makes the expense of training the unified model from scratch affordable for electric utilities in practice.

2) We develop a general framework based on pre-training paradigm, which offers a consistent workflow for diverse smart meter data applications. Different from the existing methods, this framework enables the unified models to perform a variety of applications without the task-by-task design, which reduces the experience required by electric utilities to adapt unified models for different applications.
3) We design a training scheme based on the information bottleneck, which can improve the training efficiency and the learning effect of the unified model. The information bottleneck is able to ensure that the unified model learns generic knowledge instead of task-specific ones or irrelevant information, thus reducing the model training costs.

The rest of this paper is organized as follows. Section II introduces smart meter data applications, followed by the exposure of the developed general framework. The proposed unified model and training scheme are elaborated in Section III. Section IV validates the effectiveness and superiority of the proposed method, and Section V presents the conclusion and future work of this paper.

II. General Pre-Training Framework

In this section, we first introduce and classify smart meter data applications. Then, we expound on the general framework developed for the proposed unified model.

A. Taxonomy of Smart Meter Data Applications

Typically, smart meters can measure and record information about electric customers, e.g., load consumption and voltage levels [20], in near real-time. In this paper, since we focus on applying smart meter data for demand-side tasks, we mainly consider load data [3]. In this way, we broadly divide smart meter data applications into the following 3 categories:

1) Load analysis: It includes dissecting what the data is like and providing the description and analysis of load data, such as customer categorization, anomaly detection, and data imputation. This is also convenient for subsequent stages.

2) Load forecasting: It refers to predicting what will happen to load data, i.e., providing future information on the data. Notably, this is the most essential task category of smart meter data, including point and probabilistic load forecasting.

3) Load application: It involves what decisions can be made from load data, which reflects the practical value of the data in the real world. Particularly, this can facilitate electric utilities with demand-side tasks, such as energy management, market bidding, and demand response [21].

B. General Framework Based on Pre-Training Paradigm

According to the aforesaid introduction, the task goals of smart meter data applications vary from each other, especially in different categories. This makes it more hard to determine the objective function of unified models as the traditional AI-based methods, even the common LLM methods, because the unified model needs to adapt to multiple tasks with different targets [22]. Inspired by the popular LLMs that have subverted deep learning, we adopt the pre-training paradigm [23] to develop a general framework among different tasks for unified models, which is illustrated in Fig. 1. Specifically, the general framework first trains unified models on a uniform task in the pre-training stage, and then adapts the pre-trained unified models at the fine-tuning stage, thereby accomplishing diverse tasks with only a single model. The success of this paradigm is mainly thanks to knowledge learned from massive data during pre-training, which can contribute to completing various tasks by offering the *priori* information [10] for the unified model.

To learn sufficient useful knowledge, the key is to design a suitable pre-training task for smart meter data. Particularly, we expect the unified model to learn generic rather than task-specific knowledge during pre-training so that it can be applied to multiple smart meter data applications. Moreover, the high cost of data labeling makes it difficult to construct high-quality training data for every application in practice, so pre-training cannot be conducted in a supervised learning way. Considering that electric utilities possess massive available raw meter data, we adopt unsupervised learning to acquire generic knowledge from smart meter data, which is the prevailing manner [23]. However, the lack of explicit labels as learning targets prevents unsupervised learning from being as good as supervised ones regarding training effect and efficiency. Hence, a well-designed pre-training task is crucial to ensure the pre-learning effect.

As described in Section II-A, we focus on load consumption data recorded by smart meters, which reflects customers' load characteristics, including electricity consumption behavior and habits. In other words, for smart meter data applications, the essence of using load data as input is actually to utilize the behind information to realize load analysis, forecasting, and application. In this way, load characteristics become a common concern for all applications, even though they have their own task targets. Therefore, the pre-training task in this paper is to extract load characteristics of electric customers. Specifically, we anticipate unified models to learn the generic knowledge by extracting load characteristics from load data during pre-training. Moreover, we transform the designed task into a self-supervised manner to better extract load characteristics, which has been widely proven to distill beneficial representation from unlabelled data [24]. To be specific, we randomly mask out a portion of input data, and then use unified models to extract load characteristics from masked data. Our goal is to use the extracted load characteristics to recover complete original load data, so the pre-trained unified model can be represented as:

$$f_s = \arg\min_{f} \frac{1}{N_s} \sum_{i=1}^{N_s} \|x_i - g(f(M(x_i + \epsilon)))\|_2, \quad (1)$$

where N_s is the number of samples in the pre-training dataset; $M(\cdot)$ denotes the random mask operator; ϵ is Gaussian noise; f and g denote the unified model and load data recovery model (see details in Section III-A), respectively.

Since the input and output formats of different tasks are varied, pre-trained unified models will still fall into the dilemma of task-by-task design during fine-tuning. This will bring about

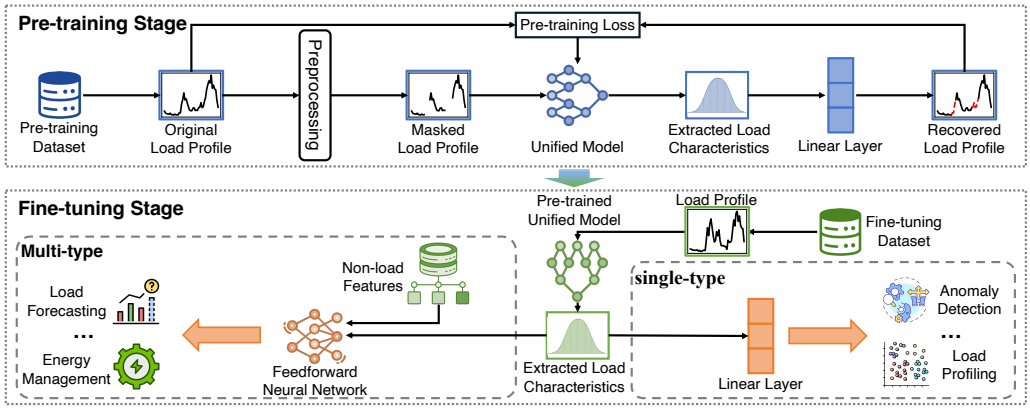


Fig. 1. The illustration of the proposed general framework, including the pre-training stage and the fine-tuning stage.

high labor expenditure in implementation. In order to prevent the cumbersome task-specific designs, we create a consistent workflow for the unified models to efficiently and conveniently perform different smart meter data applications, as illustrated in Fig. 1. Specifically, according to the type of input data, smart meter data applications are separated into the following two categories: 1) *single-type*: only involves smart meter data, mainly referring to the load analysis tasks of smart meter data applications; 2) *multi-type*: contains other inputs in addition to smart meter data (e.g., weather and location), including the load forecasting and load application tasks of smart meter data applications. For category 1, since the input data is the same as the pre-training task, we only need to add an output layer behind the unified model to adjust the output format. For category 2, we establish a new model to perform applications using the same inputs, but replace the original load data with the extracted load characteristics. Since the complex temporal relationships of load data have been processed and distilled by unified models, the new model does not need to be complicated and can be realized by common neural networks. In this paper, we uniformly use the feed-forward neural network to enhance the consistency of fine-tuning. To sum up, the unified model can accomplish multiple smart meter data applications by fine-tuning on the basis of pre-training, which is formulated as:

$$f^* = \arg\min_f \frac{1}{N'} \sum_{i=1}^{N'} \ell\left(f(x_i), y_i\right) + \lambda R(f; f_s), \qquad (2)$$

where f^* denote the fine-tuned unified model; N' represents the number of samples in fine-tuning dataset, and (x_i, y_i) is the i-th data sample; ℓ denotes the loss function; and $R(\cdot;\cdot)$ is the regularization operator with weight λ to avoid overfitting. It should be noted that although some operations are needed in the framework, we provide a consistent workflow that can be followed directly, which can still greatly reduce the complexity of fine-tuning compared to the case-by-case design approach.

In summary, we exploit the pre-training paradigm to furnish an identical training way for smart meter data applications. Furthermore, we also design a fine-tuning workflow for the different applications in a consistent process. As a result, this general framework allows electric utilities to carry out multiple tasks without additional design, as summarized in Algorithm 1, which reduces the required expertise and experience.

Algorithm 1: General Framework Based on the Pre-training Paradigm for Smart Meter Data Applications

Input : The unified model f, pre-training dataset D_s, fine-tuning dataset D_d.
Output : The pre-trained and fine-tuned unified models f_s, f^*.
Procedure:
Pre-training stage:
  Perform load characteristics extraction with the unified model f and then recover original load data as Algorithm 2;
  Obtain the pre-trained unified model f_s;
Fine-tuning stage:
if *the type of input data in* D_s *is single* **then**
  Add an output layer behind f_s;
else
  Perform load characteristics extraction for D_s with f_s;
  Establish a new model to use load characteristics and D_s;
end
  Train the adapted model and get the fine-tuned unified model f^*;
return f_s, f^*

III. Unified Model and Training Scheme

In this section, we unveil the proposed unified model for smart meter data applications, followed by the elaboration of the designed training scheme.

A. Unified Model with Mixture-of-Expert Layers

Considering that we aspire for unified model to learn generic knowledge to handle different tasks, a large amount of data is required for adequate training, which comes from different customers. These heterogeneous smart meter data elevate the training difficulty, leading to the necessity for more powerful models. Traditionally, model capacity is increased by directly expanding the number of model layers. However, this approach brings about a roughly quadratic increase in training costs [25], and thus is not suitable for electric utilities. To this end, we propose a novel unified model with mixture-of-expert layers, which can dramatically increase the model capacity without a proportional increase in computation. Specifically, the unified model consists of two primary modules: preprocessing module and encoding module, each of which comprises several layers. Fig. 2 reveals the architecture of the unified model, and we will then present each layer in turn.

1) Patching layer: Because the model input (i.e., load data) is typical time-series data, the unified model aims to learn correlations between data in each time steps. However, most existing studies use point-wise data as input, which often

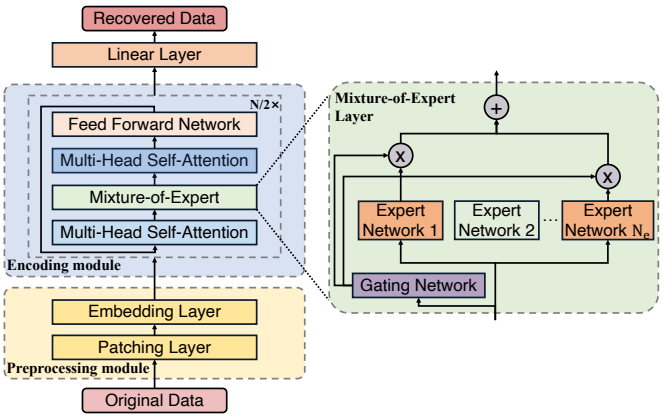


Fig. 2. The architecture of the proposed unified model, mainly including the preprocessing module and the encoding module.

results in inadequate extraction of valuable information. Thus, we convert the model input into patch-level to make it possible to capture local information from load data, thereby enhancing the comprehensiveness of extracted information. Specifically, we first divide the original input x into patches and combine them into a new sequence x_p. In other words, denote the length of load profile as L and the patch number as P, then the model input is transformed from $x \in \mathbb{R}^{L\text{x}1}$ to $x_p \in \mathbb{R}^{P\text{x}d_P}$. Moreover, the model input is normalized to alleviate the distribution shift between training and test data. In particular, we apply instance normalization to each load profile separately, which can avoid the problem of dealing with input data in inconsistent shapes.

Through the use of patches, the number of timesteps in input data reduces from L to P, which indicates the memory and computing resources required by unified models will decrease quadratically by a factor of approximately L/P [26]. Besides, the data normalization also helps to improve the convergence speed of model training. In this way, the patching layer enables the unified model to efficiently process load profiles even with limited resources, thus improving model performance.

2) Embedding layer: Although the input data is no longer point-wise after the patching layer, we still perform embedding transformation on x_p to improve its representation capability. This will facilitate the unified model to better release its high computational capacity, and accordingly enable it extract more comprehensive information from load data [27]. Specifically, considering each patch contains several load data, we apply linear transformation to enhance the amount of information. Besides, we adopt the positional encoding [28] to enhance position information of each patch within the input sequence. In particular, the patch-level input x_p is embedded as follows:

$$\boldsymbol{x}_{\text{emb}} = \mathit{FC}(\boldsymbol{x}_p) + \mathit{PE}\big(\mathit{FC}(\boldsymbol{x}_p)\big), \tag{3}$$

where $\mathit{FC}(\cdot)$ and $\mathit{PE}(\cdot)$ denote fully-connected neural network and positional encoding function, respectively. It should be noted that $\boldsymbol{x}_{\text{emb}} \in \mathbb{R}^{P\text{x}d_{\text{model}}}$ is the output of the preprocessing module, which will then be fed into the encoding module.

3) Attention layer: Following the preprocessing module, the input data has been fully prepared to facilitate the unified model to learn generic knowledge. Since we mainly consider load data as input, the complex temporal dependencies of load profiles need to be distilled delicately. For this purpose, we utilize the self-attention mechanism to extract comprehensive information from the input $\boldsymbol{x}_{\text{emb}}$. Specifically, the self-attention mechanism [28] builds the similarities between each patch and other patches in the sequence to capture temporal relationships, without being restricted by the sequence length. Furthermore, the self-attention mechanism can focus on important parts and ignore irrelevant content based on similarities, thus improving the modeling efficiency. In particular, this mechanism employs dot product to compute similarity, which can be formulated as:

$$\mathit{Atten}(\boldsymbol{x}_{\text{emb}}) = \mathit{Softmax}\left(\frac{\boldsymbol{Q}\cdot\boldsymbol{K}^{\text{T}}}{\sqrt{d_k}}\right)\cdot\boldsymbol{V}, \tag{4}$$

where $\boldsymbol{Q} = \boldsymbol{x}_{\text{emb}}\boldsymbol{W}_q$, $\boldsymbol{K} = \boldsymbol{x}_{\text{emb}}\boldsymbol{W}_k$, $\boldsymbol{V} = \boldsymbol{x}_{\text{emb}}\boldsymbol{W}_v$; $\boldsymbol{W}_q$, $\boldsymbol{W}_k$ and $\boldsymbol{W}_v$ denote three transformation matrices with the same dimension $\mathbb{R}^{d_{\text{model}}\times d_k}$; $\sqrt{d_k}$ is the scale factor to avoid vanishing gradients. Besides, in order to extract comprehensive information, we also apply the multi-head skill to enhance the attention mechanism [28]. Specifically, we first project $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ into several subspaces, followed by performing Eq. (4) on each subspace separately. Finally, we aggregate the attention results from all subspaces and then reproject into the original space. According to Eq. (4), the output $\boldsymbol{x}_{\text{atten}}$ of the multi-head self-attention mechanism can be written as:

$$\boldsymbol{x}_{\text{atten}} = \mathit{Concat}(\mathit{Atten}_1(\boldsymbol{x}_{\text{emb}}),\ldots,\mathit{Atten}_h(\boldsymbol{x}_{\text{emb}}))\cdot\boldsymbol{W}_o, \tag{5}$$

where $\mathit{Concat}(\cdot)$ denotes the concatenation operator; Atten_i is the attention result of the i-th head (subspace) and h is the number of heads; $\boldsymbol{W}_o$ denotes the projection matrix for output.

4) Mixture-of-Expert layer: In the vanilla Transformer, the attention layer is followed by a feed-forward layer to intensify the non-linear fitting ability of the model [28]. However, this is not sufficient for the unified model to process complex correlations of load data and learn adequate generic knowledge for different tasks. In the conventional way [25], it is common to augment the number of attention and feed-forward layers, which has been shown effectively improve model performance. Nevertheless, this approach will lead to a roughly quadratic boost in training costs, making it hard for electric utilities to meet the demand for model training. For this reason, we draw on the conditional computation theory to remarkably enhance model capacity in a cost-effective way [29]. Specifically, we design a mixture-of-expert layer that is composed of a number of expert networks and a gating network, to replace the naive feed-forward layer. In other words, the output of our designed layer $\boldsymbol{x}_{\text{moe}}$ is no longer determined by a single network, but the combination of all expert network outputs, where the weight of each expert network is decided by the gating network:

$$\boldsymbol{x}_{\text{moe}} = \sum_{i=1}^{N_e} G(\boldsymbol{x}_{\text{atten}})_i \cdot E_i(\boldsymbol{x}_{\text{atten}}), \tag{6}$$

$$G(x) = \mathit{Softmax}(x\cdot\boldsymbol{W}_g), \tag{7}$$

where $E_i(\cdot)$ and $G(\cdot)$ are the i-th expert network and gating network, respectively; $G(\cdot)_i$ represents the weight of $E_i(\cdot)$; $\boldsymbol{W}_g$ is the trainable weight matrix of $G(\cdot)$; N_e is the number of expert networks. It is worth mentioning that the combination of expert networks can be regarded as a manifestation of swarm

intelligence, which is different from directly increasing model layers. In this paper, each expert network is implemented by feed-forward networks with identical model architectures but holds separate parameters. In this way, if all expert networks are activated in each calculation to produce the output, this will raise a problem: the training cost of the mixture-of-expert layer is not different from directly adding model layer [25].

In response to the aforementioned issue, we develop a sparse version of the gating network, where only a handful of expert networks are allowed to contribute to generating the output at a time. Specifically, before applying $\mathit{Softmax}(\cdot)$ in Eq. (7), we only retain the top k values and adjust the remaining values to $-\infty$, which will make the weight of non-top k values become 0. Moreover, we also insert Gaussian noise before determining the top k values to enhance randomness, so that each expert network has an opportunity to be activated [29]. According to Eq. (7), the sparse gating network can be formulated as:

$$G(x) = \mathit{Softmax}(\mathit{TopK}(\mathit{Noise}(x), k)), \tag{8}$$

$$\mathit{Noise}(x) = x \cdot \boldsymbol{W}_g + \epsilon_\Phi \cdot \mathit{Softmax}(x \cdot \boldsymbol{W}_\epsilon), \tag{9}$$

$$\mathit{TopK}(x, k)_i = \begin{cases} x_i, & \text{if } x_i \text{ is the top } k \text{ of } x \\ -\infty, & \text{otherwise} \end{cases}, \tag{10}$$

where ϵ_Φ is the noise from standard Gaussian distribution; $\boldsymbol{W}_\epsilon$ denotes the trainable noise matrix that further increases the stochasticity of the process of selecting top k experts. When $G(x)_i$ is 0, the model will not need to compute $E_i(x)$, thereby significantly reducing training costs to improve efficiency. It is important to note that although only a few expert networks are involved in computing $\boldsymbol{x}_{\text{moe}}$, this sparse manner will not degrade the model capacity, because all expert networks are considered and selected by the gating network. Moreover, the combination of expert networks can further enhance the non-linear ability and also avoid the mode collapse problem [29].

However, the sparsity of the gating network may cause it to converge to an eccentric state where $G(x)$ always assigns larger weights to certain expert networks rather than equally. This makes sense because some experts will be favored at the start and thus train faster, leading to being more likely to be selected as the top k experts in the follow-up. It goes without saying that this vicious circle will render most expert networks useless and reduce model capacity. Regarding this undesirable phenomenon, we adopt the soft constrain strategy to avoid the imbalance issue among expert networks [29]. Specifically, we define the value of an expert network as the sum of its weights from $G(\cdot)$ over the data, which can be formulated as:

$$\mathit{Value}(X)_i = \sum_{x \in X} G(x)_i, \tag{11}$$

where X is a batch of training samples and $\mathit{Value}(X)_i$ denotes the value of the i-th expert network on X. In addition, we also portray the participation degree of expert networks in decision-making from a probabilistic perspective. Similarly, we define the expert's payload as the sum of the probabilities that its weights are non-zero over the data, as follows:

$$\mathit{Payload}(X)_i = \sum_{x \in X} \Phi\left(\frac{(x \cdot \boldsymbol{W}_g)_i - \mathit{Except}(\mathit{Noise}(x), k, i)}{\mathit{Softmax}((x \cdot \boldsymbol{W}_\epsilon)_i)}\right), \tag{12}$$

where Φ is the cumulative distribution function of the standard Gaussian distribution; $\mathit{Payload}(X)_i$ denotes the payload of the i-th expert network on X; $\mathit{Except}(x, k, i)$ represents the k-th top element of x except element i. It should be mentioned that $G(x)_i \neq 0$ if and only if $\mathit{Noise}(x)_i > \mathit{Except}(\mathit{Noise}(x), k, i)$.

Considering that $\mathit{Value}(X)$ and $\mathit{Payload}(X)$ describe the contribution of each expert network to the output of X, we can use their coefficient of variation to quantify the dispersion of balance levels among expert networks. In particular, a smaller coefficient of variation implies greater equality. Therefore, we add the coefficient of variation of $\mathit{Value}(X)$ and $\mathit{Payload}(X)$ into the loss function, thus ensuring equal contributions.

It should be noted that since the goal of the pre-training task is to recover load profiles using extracted load characteristics, we add a linear layer followed by the unified model for output, which corresponds to the model g in Eq. (1). This is because most temporal features have been extracted, there is no need to require too much model capacity to produce recovered data. In this way, we choose the linear layer for load data recovery, which is also the popular approach in many pre-training task [10]. As for the fine-tuning stage, we adopt the feed-forward neural network or linear layer according to the type of tasks, i.e., following the consistent workflow in Section II-B.

B. Model Training Scheme Based on Information Bottleneck

According to Eq. (1), the output of unified models (i.e., load characteristics) needs to contain as much information about load data as possible so that recovery error can be minimized. However, due to the high model capability of neural networks, the proposed unified model will inevitably learn the irrelevant information during pre-training [30], which results in the need for more training to ensure it can learn generic knowledge.

To improve the pre-training efficiency, we design a new training scheme based on information bottleneck from the view of information theory [30]. To be specific, we apply a constraint to restrict the irrelevant information about input data X in load characteristics Z. Typically, the mutual information is used to measure the relevance between two random variables:

$$I(X, Y) = \int dxdy\, p(x, y) \log \frac{p(x, y)}{p(x)p(y)}, \tag{13}$$

where $I(X, Y)$ denotes the mutual information of X and Y; and $p(x, y)$ is the joint probability density of X and Y. In this way, the objective of pre-training can be reformulated as:

$$\max I(Z, Y) \text{ s.t. } I(X, Z) \leq I_c, \tag{14}$$

where I_c is the information constraint. By introducing the Lagrange multiplier β, the objective function is rewritten as:

$$R_{\text{IB}} = I(Z, Y) - \beta I(X, Z), \tag{15}$$

where R_{IB} denotes the information bottleneck [30]. Hence, the training goal is to make Z that is maximally informative about Y while being maximally compressive about X. Consequently,

the unified model can learn generic knowledge from load data with high quality and efficiency by maximizing R_{IB}.

Following the information theoretic view, there is an information flow $X \rightarrow Z \rightarrow Y$ in unified models. Specifically, in the pre-training task, the proposed unified model extracts load characteristics Z from original load data X, and produces recovered load data Y based on Z, which means that valid information is transmitted from X to Y via Z. Moreover, since Z is inherently determined by X and does not depend on Y, the joint distribution $p(X, Y, Z)$ can be represented as:

$$p(X, Y, Z) = p(X)p(Y|X)P(Z|X), \tag{16}$$

where $p(Z|X)$ is the conditional distribution, which represents the unified model f with X as input and Z as output. Similarly, $p(Y|Z)$ corresponds to the load data recovery model g. Thus, according to the Bayes' theorem and Eq. (13), the two mutual information terms in Eq. (15) can be rewritten as:

$$\begin{aligned} I(Z, Y) &= \int dzdy\, p(z, y) \log \frac{p(y|z)}{p(y)}, \\ I(X, Z) &= \int dxdz\, p(x, z) \log \frac{p(z|x)}{p(z)}, \end{aligned} \tag{17}$$

where $p(y|z) = \int dx \frac{p(y|x)p(z|x)p(x)}{p(z)}$, $p(z) = \int dxp(z|x)p(x)$. However, since $p(x)$ is difficult to compute in practice, $p(y|z)$ and $p(z)$ are intractable, which leads to the maximization of R_{IB} being computationally infeasible during pre-training.

To this end, we utilize the variational inference [31] to solve the intractability problem. Specifically, variational inference refers to using a simple distribution as an approximation of the target distribution, and achieving approximate replacement by continuously narrowing the distance between them. Here, we will apply variational inference to the two intractable terms in turn. First, we let $q(y|z)$ be the variational approximation of $p(y|z)$. In particular, we can get the following lower bound:

$$I(Z, Y) \geq \int dxdzdy\, p(x)p(y|x)p(z|x) \log q(y|z). \tag{18}$$

Then, we consider another term $p(z)$, and use $m(z)$ to be its variational approximation. Similarly, we gain the upper bound:

$$I(X, Z) \leq \int dxdz\, p(x)p(z|x) \log \frac{p(z|x)}{m(z)}. \tag{19}$$

Substituting both upper and lower bounds into Eq. (15), we can obtain the lower bound of R_{IB} (i.e., its variational form):

$$\begin{aligned} R_{\text{IB}} \geq &\int dxdzdy\, p(x)p(z|x)p(y|x) \log q(y|z) \\ &- \beta \int dxdz\, p(x)p(z|x) \log \frac{p(z|x)}{m(z)} = R_{\text{VIB}}, \end{aligned} \tag{20}$$

where R_{VIB} is the variational information bottleneck [31]. Now, the maximization of R_{IB} can be achieved by maximizing R_{VIB}. To compute the lower bound R_{VIB}, we acquire the empirical distribution by sampling on the pre-training dataset, which can approximate $p(x, y) = p(x)p(y|x)$. In this way, R_{VIB} can be estimated in the following approach:

$$\begin{aligned} R_{\text{VIB}} &\approx \frac{1}{N'} \sum_{i=1}^{N'} \left[\int dz \big(p(z|x_i) \log q(y_i|z) - \beta p(z|x_i) \log \frac{p(z|x_i)}{m(z)} \big) \right] \\ &= \frac{1}{N'} \sum_{i=1}^{N'} \left[\mathbb{E}_{z \sim p(z|x_i)} \log q(y_i|z) - \beta \text{KL}(p(z|x_i) \| m(z)) \right], \end{aligned} \tag{21}$$

where KL denotes the Kullback–Leibler divergence. In consequence, maximizing R_{IB} can be transformed into minimizing the following objective during pre-training:

$$\min \frac{1}{N'} \sum_{i=1}^{N'} \left[\mathbb{E}_{z \sim p(z|x_i)} - \log q(y_i|z) + \beta \text{KL}(p(z|x_i) \| m(z)) \right]. \tag{22}$$

Because $p(Z|X)$ corresponds to the unified model f, we can sample multiple z from the output of f, and then calculate the approximation of the mathematical expectation $\mathbb{E}_{z \sim p(z|x_i)}$:

$$\mathbb{E}_{z \sim p(z|x_i)} \log q(y_i|z) = \frac{1}{N_z} \sum_{s=1}^{N_z} \log q(y_i|z_s), \tag{23}$$

where N_z is the total sampling amount of z from $p(z|x_i)$. Furthermore, since $q(y|z)$ is the variational approximation, we assume that its distribution is the multivariate Gaussian, i.e., $q(y_i|z_s) = \mathcal{N}(\boldsymbol{\mu}'_s, \sum_{\sigma'})$. By this means, $q(y_i|z_s)$ can be represented in analytical form, as follows:

$$\begin{aligned} \log q(y_i|z_s) &= -\frac{1}{2} \sum_{k=1}^{K} \frac{(y_i^k - \mu_s'^{(k)})^2}{\sigma'^{(k)}} - \log \sqrt{(2\pi)^K \prod_{k=1}^{K} \sigma'^{(k)}} \\ &\overset{(1)}{=} -\|y_i - \mu'_s\|_2^2, \end{aligned} \tag{24}$$

where K denotes the dimension of $\mathcal{N}(\boldsymbol{\mu}'_s, \sum_{\sigma'})$. In particular, the equality (1) holds because we set the value of all elements in $\sum_{\sigma'}$ to $\frac{1}{2}$ for ease of calculation, i.e., $\sigma'^{(k)} = \frac{1}{2}$.

Similarly, we also assume that $m(z)$ belongs to standard multivariate Gaussian distribution, i.e., $m(z) \sim \mathcal{N}(\mathbf{1}, \mathbf{0})$. In addition, we suppose $Z|X$ obeys the multivariate Gaussian distribution, i.e., $p(z|x) \sim \mathcal{N}(\boldsymbol{\mu}, \sum_{\sigma})$. This makes sense because the multivariate Gaussian distribution has strong fitting ability, and can thus fit the distribution of the outputs of model f. In this way, the KL term in Eq. (22) can be written as:

$$\text{KL}(p(z|x_i) \| m(z)) = \sum_{j=1}^{d} \frac{1}{2} (-1 + \mu^{(j)^2} + \sigma^{(j)^2} - 2 \log \sigma^{(j)}), \tag{25}$$

where $\mu^{(j)}$ and $\sigma^{(j)}$ are the j-th element of mean vector $\boldsymbol{\mu}$ and covariance matrix $\sum_{\sigma}$ in d-dimensional Gaussian distribution.

In conclusion, combining Eqs. (22)–(25), we can acquire the information bottleneck-based objective function of unified models during pre-training. Meanwhile, since the pre-training task is performed by the proposed unified model f, based on Eqs. (1) and (11)–(12), the loss function can be formulated as:

$$\begin{aligned} L = &\frac{1}{N'} \sum_{i=1}^{N'} \|x_i - g(f(M(x_i) + \epsilon))\|_2^2 \\ &+ \beta \frac{1}{N'} \sum_{i=1}^{N'} \sum_{j=1}^{d} (-1 + \mu^{(j)^2} + \sigma^{(j)^2} - 2 \log \sigma^{(j)}) \end{aligned} \tag{26}$$

Algorithm 2: Unified Model Pre-training Scheme

Input : The unified model f, pre-training dataset D_s, batch size B, training epoch E, Adam parameters α, β_1, β_2.
Output : The pre-trained unified model f_s.

Procedure:
for $e = 1, \ldots, E$ **do**
  for *each batch of training data* **do**
    Sample B data $x \sim D_s$ and add Gaussian noise ϵ to x;
    for $i = 1, \ldots, B$ **do**
      Convert $x^{(i)} \in \mathbb{R}^{Lx1}$ into batch-wise $x_p^{(i)} \in \mathbb{R}^{Px d_P}$ and perform instance normalization to $x_p^{(i)}$;
      Randomly mask $x_p^{(i)}$ and then embed $x_p^{(i)}$ into $\boldsymbol{x}_{\text{emb}}^{(i)}$;
      Calculate gate weights $G(x^{(i)})$ according to Eqs. (8)–(10), and two constraint items $\mathit{Value}(x^{(i)})$ and $\mathit{Payload}(x^{(i)})$ according to Eqs. (11) and (12);
      Extract load characteristics $z^{(i)}$ based on Eqs. (4) and (6), and obtain recovered data $\hat{x}^{(i)}$ from $z^{(i)}$;
      Compute the pre-training loss $L^{(i)}$ based on Eq. (26);
    end
    Update model's parameters based on Adam algorithm $f \leftarrow \mathit{Adam}(\nabla_f \frac{1}{B} \sum_{i=1}^{B} L^{(i)}, \alpha_{\mathit{Adam}}, \beta_1, \beta_2)$;
  end
end
return f

$$+ \omega_{\mathit{value}} \frac{\mathit{Var}(\mathit{Value}(X))}{\mathit{Mean}(\mathit{Value}(X))^2} + \omega_{\mathit{load}} \frac{\mathit{Var}(\mathit{Payload}(X))}{\mathit{Mean}(\mathit{Payload}(X))^2},$$

where μ and σ are the mean and standard deviation of unified model's output (i.e., $f(M(x_i) + \epsilon)$), respectively; $\mathit{Mean}(\cdot)$ and $\mathit{Var}(\cdot)$ denote the mean and variance functions; ω_{value} and ω_{load} represent two factors for balancing expert networks. In addition, we adopt the Adam algorithm for model update, and the details of pre-training are summarized in Algorithm 2. It should be noted that the fine-tuning stage uses the conventional model training ways, instead of the designed training scheme.

IV. Case Studies

A. Experiment Settings

1) Dataset: To satisfy the amount of training data required to learn generic knowledge, we choose two public datasets for our experiments, which are both composed of load profiles recorded by smart meters. Specifically, [32] contains historical load profiles of nearly 6,500 customers in Ireland from July 2009 to December 2010. Meanwhile, [33] includes electricity consumption data for over 5,000 UK residents from November 2011 to February 2014. To ensure data availability, we perform data preprocessing on raw datasets, and ultimately screen and retain approximately 3 million complete daily load profiles with a granularity of 30 minutes. Considering that there are two stages in our general framework, we use 80% of overall data for pre-training and the remaining for fine-tuning. In particular, the fine-tuning dataset is divided for the model training, validation, and testing in specific tasks in the ratio of 6:2:2, where the dataset split in different tasks is randomized.

2) Task & Metric: Considering the proposed unified model is supposed to be able to accomplish multiple smart meter data applications, we select three different tasks to comprehensively verify its effectiveness, as follows:

- *Load forecasting*: It refers to predicting future electricity consumption based on historical load, weather information, and other factors. Since this is a regression task, we choose RMSE, MAE, and MAPE as metrics. Here, we consider day-ahead and hour-ahead load forecasting, which are the most common tasks on the demand side.
- *Anomaly detection*: It refers to the identification of data points that deviate from the standard or expected. Based on the confusion matrix, we select three common classification metrics, i.e., accuracy, precision, and recall. Here, we consider data anomalies caused by scaling and ramping attacks, which are simulated according to [34].
- *Data imputation*: It refers to replacing missing items with substituted values based on available data. To evaluate the performance, we also adopt RMSE, MAE, and MAPE to compute imputation effects of missing parts. In this paper, we consider the missing range from 10% to 50%, which are the regular missing ratios in reality except for attacks.

In particular, these selected tasks encompass every category described in Section II-B, and are also classical tasks for time series data. Therefore, we believe that they are representative enough to validate our proposed method.

TABLE I
Implementation Details of The Proposed Method

Parameter	Definition	Value
P	the number of data patch	12
d_{model}	the dimension of embedding vector	16
d_k	the dimension of Q, K, and V parts	4
N_e	the number of expert networks	8
k	the expert value of top selection	2
N_{model}	the layer number of encoding module	8
β	the value of Lagrange penalty	0.001
ω_{value}, ω_{load}	balancing factors of expert networks	0.1, 0.1
E	the number of training epochs	100
B	the batch size of training data	16
α	the learning rate of Adam	0.001

3) Implementation: The proposed method is implemented by the open-source machine learning framework PyTorch, and the implementation details are summarized in Table I. Besides, we conduct all experiments on an Ubuntu 18.04 LTS platform, which is equipped with the Intel Core i9-10980XE CPU and NVIDIA GeForce RTX 3090 GPU.

B. Performance Comparison with Task-specific Methods

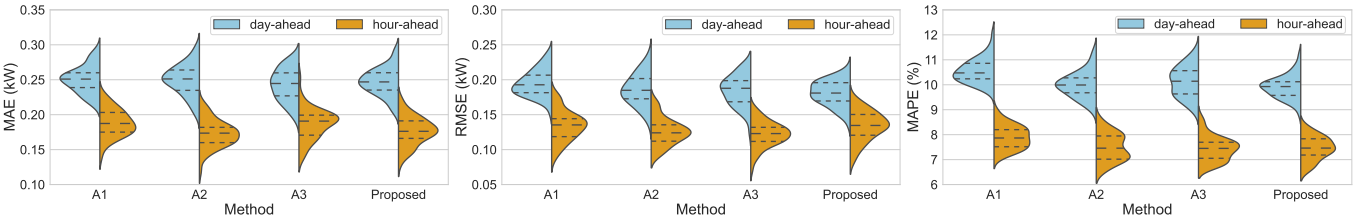
In this section, we validate the effectiveness of our proposed method on different tasks by comparing it with state-of-the-art studies. In particular, we choose three benchmarks from each task for performance comparison, as follows:

- A1: A hybrid load forecasting model combining LSTM and self-attention proposed in 2021 by Zang et al. [35].
- A2: A robust framework for individual load forecasting under concept drift proposed in 2022 by Yang et al. [36].
- A3: A short-term load forecasting model based on causal inference proposed in 2024 by Wang et al. [37].
- B1: A machine learning anomaly detection model with statistical features proposed in 2019 by Cui et al. [38].
- B2: A non-intrusive load monitoring method for anomaly detection proposed in 2021 by Azizi et al. [39].
- B3: A unsupervised anomaly detection framework for load data proposed in 2023 by Wang et al. [40].

TABLE II
NUMERICAL RESULTS OF PERFORMANCE COMPARISON FOR LOAD FORECASTING AND ANOMALY DETECTION

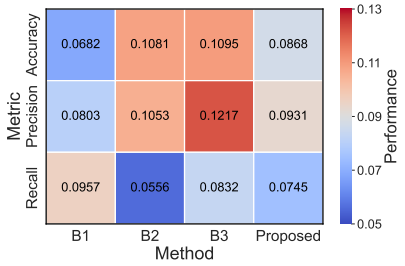
Load Forecasting							Anomaly Detection						
Method	Day-ahead			Hour-ahead			Method	Scaling attack			Ramping attack		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE		Acc.	Prec.	Recall	Acc.	Prec.	Recall
A1	0.2524	0.1938	10.4691	0.1918	0.1353	7.8619	B1	0.8121	0.9173	**0.7774**	0.8219	0.8869	0.7483
A2	0.2487	0.1842	9.9956	**0.1741**	0.1251	7.4185	B2	0.8239	0.8992	0.7465	0.8277	0.8950	0.7566
A3	**0.2426**	0.1857	10.1279	0.1877	**0.1192**	**7.2394**	B3	0.8328	0.8917	0.7566	**0.8343**	**0.9032**	0.7548
Proposed	0.2443	**0.1839**	**9.9329**	0.1796	0.1275	7.3848	Proposed	**0.8444**	**0.9234**	0.7729	0.8329	0.8918	**0.7567**

*MAE and RMSE are in kW and MAPE is in percentage. Bold term indicates the best performance, while underlining term represents second best.

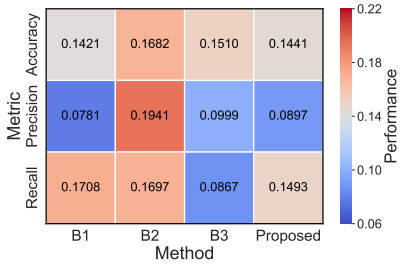


(a) (b) (c)

Fig. 3. The performance comparison of the load forecasting task in three metrics. The three dashed lines in each violin plot represent the first quartile (25%), median (50%), and third quartile (75%) from top to bottom, respectively. (a) MAE, (b) RMSE, (c) MAPE.



(a)



(b)

Fig. 4. The performance comparison of the anomaly detection task under two scenarios. The central number in each cell represents the standard deviation of the corresponding evaluation metric. (a) Scaling attack, (b) Ramping attack.

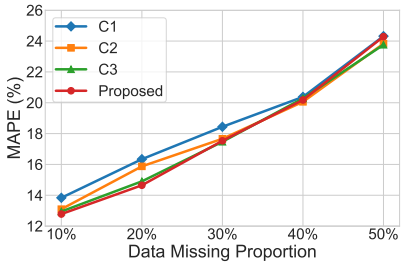
- C1: A missing load data imputation method considering neighbor information proposed in 2021 by [41].
- C2: A building load data imputation model with mixture factor analysis proposed in 2021 by Jeong et al. [42].
- C3: A missing load data restoration method based on the bidirectional encoder proposed in 2024 by Hu et al. [43].

Note that notation A, B, and C correspond to load forecasting, anomaly detection, and data imputation tasks, respectively. In addition, we repeat all experiments five times and calculate the average value as results, to avoid human interference.

Table II summarizes performance comparison results of load forecasting and anomaly detection tasks. It can be observed that the proposed unified model achieves results that are not inferior to these task-specific benchmarks in load forecasting. Specifically, the RMSE and MAE of our proposed method are the lowest in the day-ahead scenario, with values of only 0.18 kW and 9.9%, respectively. Moreover, although its MAE is larger than benchmarks, the gap is almost negligible. Similarly, in the hour-ahead scenario, the proposed method is very close to benchmarks in all evaluation metrics, where the difference of MAPE is within 0.15%. Besides, the quantile comparison in Fig. 3 comprehensively demonstrates that our proposed model has comparable performance to benchmarks. It is worth noting that our proposed method shows a performance degradation as temporal resolution increases, where its results are not the best in the hour-ahead scenario. This is reasonable since the unified model will inevitably sacrifice some accuracy for broad applicability, while this degradation is acceptable owing to the small performance discrepancy. As for the anomaly detection task, our proposed method realizes competitive performance under scaling attack, with an accuracy of 0.84 and a precision of 0.92. Moreover, Fig. 4 shows that the performance fluctuation of the proposed method is also not the largest, which further signifies it is not inferior to benchmarks. Similarly, there is a slight performance decline when switching to ramping attack, where its accuracy and precision drop by 0.16% and 1.28% compared to the best benchmark, respectively. In summary, the subtle performance gaps in practical tasks demonstrate that our proposed method can be applied to multiple applications.

In order to intuitively evaluate data imputation effects under different missing ratios, we provide a visual demonstration of performance comparison for the proposed unified model with benchmarks, as presented in Fig. 5. It can be seen that our proposed method ranks at the top in almost all five scenarios. Specifically, its RMSE is controlled within 0.3 kW and 0.4 kW at missing rates of 10% and 20%, and the corresponding MAPEs are both the lowest at 12.8% and 14.6%. Furthermore, even though the proposed method lags behind some benchmarks in other scenarios, the performance gaps are tolerable,

(a)



(b)

Fig. 5. The performance comparison of the data imputation task in two evaluation metrics. (a) RMSE, (b) MAPE.

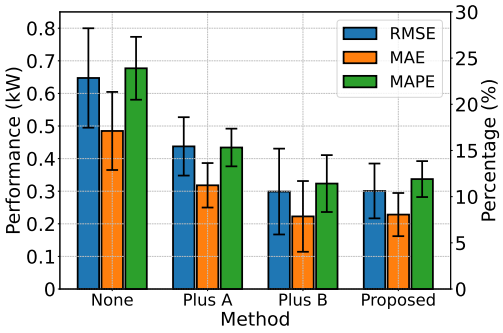


Fig. 6. The ablation study of the mixture-of-expert layer and information bottleneck in our proposed method under the data imputation task. None: the proposed method without mixture-of-expert layer and information bottleneck; Plus A: the proposed method without mixture-of-expert layer; Plus B: the proposed method without information bottleneck.

where the largest discrepancies in RMSE and MAPE are only 0.02 kW and 0.5%, respectively. In other words, although these benchmarks are task-specific, their performance is not significantly better than our proposed method, which can carry out multiple tasks. Therefore, the wide applicability of the unified model is further verified, and the effectiveness and superiority of our proposed method are also demonstrated.

C. Ablation Study

In this part, we investigate the model performance on tasks after removing certain components to explore their contributions. Specifically, we inspect the mixture-of-expert layer and information bottleneck, which are at the heart of the proposed method. For clarity, we take data imputation as an example. Similarly, we present the average values of five independent experiments as the results and illustrated in Fig. 6.

It is clear to see that there is a significant increase in model performance with the support of the information bottleneck and mixture-of-expert layer, where the RMSE is reduced from 0.65 kW by 32.4% and 52.8%. The different reduction rates reflect that these two core components contribute differently. In particular, the mixture-of-expert layer provides a remarkable enhancement in accuracy with the MAE of 0.22 kW, while the information bottleneck focuses more on the variance but its MAE still exceeds 0.32 kW. Furthermore, when we continue to add another component on this basis, the model performance changes quite differently. Specifically, after introducing the mixture-of-expert layer, the RMSE and MAE are greatly reduced by 8.3% and 6.2%, and the error variance is kept small, where the standard deviation of MAPE is controlled at 1.95%. On the other hand, by adding the information bottleneck, there is even a slight degradation in model performance, e.g., the RMSE goes from 0.30 kW to 0.40 kW. However, the variance of all metrics has been significantly diminished, with a decline of 36.4%, 36.5%, 23.1%, respectively. We believe that this is reasonable because the role of the information bottleneck is to enhance the generalization ability, and there is a trade-off between accuracy and generalization. In summary, both the mixture-of-expert layer and information bottleneck contribute to model performance, with the former taking the lead, which further demonstrates the effectiveness of our proposed method.

D. Hyperparameter Sensitivity

In this part, we explore the effectiveness of our proposed method from the perspective of methodology hyperparameters. Specifically, we focus on four hyperparameters, including the the number of expert network N_e, sparse selection k, patch dimension d_P, and Lagrange multiplier β. Here, we consider both the model performance and training time for evaluation. Similarly, we choose the data imputation task and present the average values of repeated experiments, as shown in Fig. 7.

It can be seen that as N_e increases, the model performance fluctuates slightly, with a maximum discrepancy of only 0.054 kW in RMSE. At the same time, the training time continues to rise. This phenomenon is more evident in the study of k. In particular, there is almost no gap in model performance under different values of k, while the training time is multiplying at a constant rate. This indicates that our proposed unified model does not need to increase model parameters to improve the performance, which will reduce the model training cost. In addition, with the growth of d_P, the variation of RMSE is not obvious, but the training time continues to significantly reduce from 4.5 seconds to less than 2 seconds. This demonstrates the proposed model can improve efficiency without compromising performance. In contrast, as β boots, the model's training time hardly changes, but the RMSE climbs significantly due to the pursuit of generalization. In other words, in order to achieve the same performance, different values of β will bring different training costs. This means the unified model can learn generic knowledge with high efficiency via the information bottleneck, thus further proving the effectiveness of our proposed method.

V. Conclusion

In this paper, we focus on building data-driven models for smart meter data applications. Owing to specialized properties of each application, existing AI-based methods are uniquely

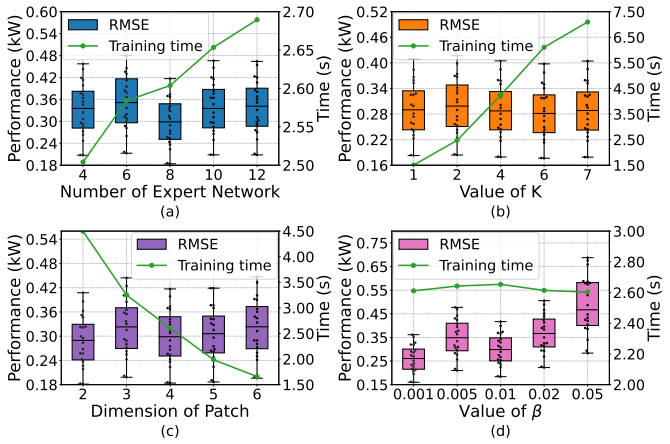


Fig. 7. The data imputation results under different hyperparameter settings of the proposed method in terms of RMSE and the training time. The training time represents the total duration to train one round with every 100 samples.

designed, leading to their low compatibility for different tasks. In addition, because of the astronomical cost of pre-training and elaborate design of fine-tuning, the vogue large language models are also not suitable. To address this issue, we propose a unified model for smart meter data applications. Specifically, we develop a general framework to unify training objective and build consistent workflow for various tasks. Then, we propose a unified model to learn generic knowledge in low costs with mixture-of-expert layers. Moreover, we design an information bottleneck-based training scheme for the efficient knowledge learning. Case studies verify the effectiveness of our proposed method, where its performance differences to state-of-the-art methods in three tasks are within 2.9%, 1.2%, and 1.6%. In addition, we inspect the contribution of core components via the ablation study and sensitivity analysis, where the impact of their hyperparameter selection on performance are controlled within 10%. Besides, the training time variation is within 0.05 seconds per sample, which further verifies the effectiveness.

Considering that some operations are still required in the fine-tuning stage (i.e., add linear layer or new model), how to further improve the universality and automation of the general framework is very necessary, and will be considered in our future work. Furthermore, we also intend to apply the unified model to power system tasks in other fields in the future, which are beyond the scope of smart meter data applications.

REFERENCES

[1] M. L. Di Silvestre, S. Favuzza, E. R. Sanseverino, and G. Zizzo, "How decarbonization, digitalization and decentralization are changing key power infrastructures," *Renew. Sustain. Energy Rev.*, vol. 93, pp. 483–498, 2018.

[2] Adarsh Krishnan, "Smart electricity meter market 2024: Global adoption landscape." https://iot-analytics.com/smart-meter-adoption/.

[3] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 3125–3148, 2018.

[4] J. Ruan, G. Fan, Y. Zhu, G. Liang, J. Zhao, F. Wen, and Z. Y. Dong, "Super-resolution perception assisted spatiotemporal graph deep learning against false data injection attacks in smart grid," *IEEE Trans. Smart Grid*, vol. 14, no. 5, pp. 4035–4046, 2023.

[5] S. Powell, G. V. Cezar, L. Min, I. M. Azevedo, and R. Rajagopal, "Charging infrastructure access and operation to reduce the grid impacts of deep electric vehicle adoption," *Nat. Energy*, vol. 7, no. 10, pp. 932–945, 2022.

[6] Z. Wang, P. Yu, and H. Zhang, "Privacy-preserving regulation capacity evaluation for hvac systems in heterogeneous buildings based on federated learning and transfer learning," *IEEE Trans. Smart Grid*, vol. 14, no. 5, pp. 3535–3549, 2023.

[7] L. Xie, X. Zheng, Y. Sun, T. Huang, and T. Bruton, "Massively digitized power grid: opportunities and challenges of use-inspired ai," *Proc. IEEE*, vol. 111, no. 7, pp. 762–787, 2022.

[8] S. Tu, Y. Zhang, J. Zhang, and Y. Yang, "Powerpm: Foundation model for power systems," *arXiv preprint arXiv:2408.04057*, 2024.

[9] S. Gao, T. Koker, O. Queen, T. Hartvigsen, T. Tsiligkaridis, and M. Zitnik, "Units: A unified multi-task time series model," *arXiv preprint arXiv:2403.00131*, 2024.

[10] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[11] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[12] A. de Vries, "The growing energy footprint of artificial intelligence," *Joule*, vol. 7, no. 10, pp. 2191–2194, 2023.

[13] C. Huang, S. Li, R. Liu, H. Wang, and Y. Chen, "Large foundation models for power systems," *arXiv preprint arXiv:2312.07044*, 2023.

[14] T. Mongaillard, S. Lasaulce, O. Hicheur, C. Zhang, L. Bariah, V. S. Varma, H. Zou, Q. Zhao, and M. Debbah, "Large language models for power scheduling: A user-centric approach," *arXiv preprint arXiv:2407.00476*, 2024.

[15] M. Jia, Z. Cui, and G. Hug, "Enabling large language models to perform power system simulations with previously unseen tools: A case of daline," *arXiv preprint arXiv:2406.17215*, 2024.

[16] J. Ruan, G. Liang, H. Zhao, G. Liu, X. Sun, J. Qiu, Z. Xu, F. Wen, and Z. Y. Dong, "Applying large language models to power systems: Potential security threats," *IEEE Trans. Smart Grid*, vol. 15, no. 3, pp. 3333–3336, 2024.

[17] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," in *The Eleventh International Conference on Learning Representations*, 2023.

[18] Y. Zhao, J. Liu, X. Liu, Y. Nie, J. Liu, and C. Chen, "Ldm: A generic data-driven large distribution network operation model," *IEEE Trans. Smart Grid*, vol. 15, no. 4, pp. 4284–4287, 2024.

[19] Z. Zhang, H. Hui, and Y. Song, "Response capacity allocation of air conditioners for peak-valley regulation considering interaction with surrounding microclimate," *IEEE Trans. Smart Grid*, vol. 16, no. 2, pp. 1155–1167, 2025.

[20] H. Li, H. Zhang, J. Zhang, Q. Wu, and C.-K. Wong, "A frequency-secured planning method for integrated electricity-heat microgrids with virtual inertia suppliers," *Appl. Energy*, vol. 377, p. 124540, 2025.

[21] P. Yu, Z. Wang, H. Zhang, and Y. Song, "Safe reinforcement learning for power system control: A review," *arXiv preprint arXiv:2407.00681*, 2024.

[22] Z. Zhu and H. Zhang, "Real-time coordinated operation of electric vehicle fast charging stations with energy storage: An efficient spatiotemporal decomposition approach," *IEEE Trans. Smart Grid*, 2025.

[23] K. He, R. Girshick, and P. Dollár, "Rethinking imagenet pre-training," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4918–4927, 2019.

[24] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, "Self-supervised learning: Generative or contrastive," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 857–876, 2021.

[25] F. Xue, Z. Shi, F. Wei, Y. Lou, Y. Liu, and Y. You, "Go wider instead of deeper," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 8779–8787, 2022.

[26] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730*, 2022.

[27] Z. Wang and H. Zhang, "Customized load profiles synthesis for electricity customers based on conditional diffusion models," *IEEE Trans. Smart Grid*, vol. 15, no. 4, pp. 4259–4270, 2024.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[29] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *arXiv preprint arXiv:1701.06538*, 2017.

[30] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *2015 IEEE information theory workshop*, pp. 1–5, IEEE, 2015.

[31] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," in *International Conference on Learning Representations*, pp. 1–19, 2017.

[32] Commission for Energy Regulation (CER). (2012), "CER Smart Metering Project - Electricity Customer Behaviour Trial." 2009-2010 [dataset]. 1st Edition. Irish Social Science Data Archive. SN: 0012-00. https://www.ucd.ie/issda/data/commissionforenergyregulationcer/.

[33] J. R. Schofield, R. Carmichael, S. Tindemans, M. Bilton, M. Woolf, G. Strbac, *et al.*, "Low carbon london project: Data from the dynamic time-of-use electricity pricing trial, 2013," *UK Data Service, SN*, vol. 7857, no. 2015, pp. 1–5, 2015.

[34] M. Cui, J. Wang, and M. Yue, "Machine learning-based anomaly detection for load forecasting under cyberattacks," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5724–5734, 2019.

[35] H. Zang, R. Xu, L. Cheng, T. Ding, L. Liu, Z. Wei, and G. Sun, "Residential load forecasting based on lstm fusing self-attention mechanism with pooling," *Energy*, vol. 229, p. 120682, 2021.

[36] E. Yang and C.-H. Youn, "Temporal data pooling with meta-initialization for individual short-term load forecasting," *IEEE Trans. Smart Grid*, vol. 14, no. 4, pp. 3246–3258, 2022.

[37] Z. Wang, H. Zhang, R. Yang, and Y. Chen, "Improving model generalization for short-term customer load forecasting with causal inference," *IEEE Transactions on Smart Grid*, vol. 16, no. 1, pp. 424–436, 2025.

[38] M. Cui, J. Wang, and M. Yue, "Machine learning-based anomaly detection for load forecasting under cyberattacks," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5724–5734, 2019.

[39] E. Azizi, M. T. Beheshti, and S. Bolouki, "Appliance-level anomaly detection in nonintrusive load monitoring via power consumption-based feature analysis," *IEEE Trans. Consum. Electron.*, vol. 67, no. 4, pp. 363–371, 2021.

[40] X. Wang, Z. Yao, and M. Papaefthymiou, "A real-time electrical load forecasting and unsupervised anomaly detection framework," *Appl. Energy*, vol. 330, p. 120279, 2023.

[41] C. E. Borges, O. Kamara-Esteban, T. Castillo-Calzadilla, C. M. Andonegui, and A. Alonso-Vicario, "Enhancing the missing data imputation of primary substation load demand records," *Sustain. Energy Grids Netw.*, vol. 23, p. 100369, 2020.

[42] D. Jeong, C. Park, and Y. M. Ko, "Missing data imputation using mixture factor analysis for building electric load data," *Appl. Energy*, vol. 304, p. 117655, 2021.

[43] Y. Hu, K. Ye, H. Kim, and N. Lu, "Bert-pin: A bert-based framework for recovering missing data segments in time-series load profiles," *IEEE Trans. Ind. Inform.*, vol. 20, no. 10, pp. 12241–12251, 2024.

Zhenyi Wang (S'22) received the B.E. degree in cybersecurity from Sichuan University, Chengdu, China, in 2021. He is currently pursuing the Ph.D. degree in electrical and computer engineering at University of Macau, Macao, China. From Aug. 2024 to Feb. 2025, he was a visiting student researcher with the Electrical Energy Systems and Applications Group at KU Leuven, Belgium.

His research interests include data analytics in demand response and electricity market, trustworthy and data-centric AI for power systems, and the intersection of causal inference with AI.

Hongcai Zhang (S'14–M'18–SM'23) received the B.S. and Ph.D. degree in electrical engineering from Tsinghua University, Beijing, China, in 2013 and 2018, respectively. He is currently an Assistant Professor with the State Key Laboratory of Internet of Things for Smart City and Department of Electrical and Computer Engineering, University of Macau, Macao, China. In 2018-2019, he was a postdoctoral scholar with the Energy, Controls, and Applications Lab at University of California, Berkeley, where he also worked as a visiting student researcher in 2016. He is an associate editor of IEEE Transactions on Power Systems, associate editor of Journal of Modern Power Systems and Clean Energy, and associate editor of iEnergy.

His current research interests include urban energy systems, transportation electrification, and distributed energy resources.

Geert Deconinck (S'88–M'96–SM'00) received the M.Sc. degree in electrical engineering and the Ph.D. degree in engineering sciences from KU Leuven, Belgium, in 1991 and 1996, respectively, where he is currently a Full Professor. He has been heading the Research Group on Electrical Energy Systems and Application for more than 12 years, and has been serving twice as division head of the KU Leuven division in the EnergyVille Research Center.

His research focuses on robust distributed coordination and control, specifically in the context of cyber-physical energy systems. He has been a Fellow of the Institute of Engineering and Technology since 2013.

Yonghua Song (F'08) received the B.E. and Ph.D. degrees from the Chengdu University of Science and Technology, Chengdu, China, and the China Electric Power Research Institute, Beijing, China, in 1984 and 1989, respectively, all in electrical engineering. He was awarded DSc by Brunel University in 2002, Honorary DEng by University of Bath in 2014 and Honorary DSc by University of Edinburgh in 2019. From 1989 to 1991, he was a Post-Doctoral Fellow at Tsinghua University, Beijing. He then held various positions at Bristol University, Bristol, U.K.; Bath University, Bath, U.K.; and John Moores University, Liverpool, U.K., from 1991 to 1996. In 1997, he was a Professor of Power Systems at Brunel University, where he was a Pro-Vice Chancellor for Graduate Studies since 2004. In 2007, he took up a Pro-Vice Chancellorship and Professorship of Electrical Engineering at the University of Liverpool, Liverpool. In 2009, he joined Tsinghua University as a Professor of Electrical Engineering and an Assistant President and the Deputy Director of the Laboratory of Low-Carbon Energy. During 2012 to 2017, he worked as the Executive Vice President of Zhejiang University, as well as Founding Dean of the International Campus and Professor of Electrical Engineering and Higher Education of the University. Since 2018, he became Rector of the University of Macau and the director of the State Key Laboratory of Internet of Things for Smart City. His current research interests include smart grid, electricity economics, and operation and control of power systems.

Prof. Song was elected as the Vice-President of Chinese Society for Electrical Engineering (CSEE) and appointed as the Chairman of the International Affairs Committee of the CSEE in 2009. In 2004, he was elected as a Fellow of the Royal Academy of Engineering, U.K. In 2019, he was elected as a Foreign Member of the Academia Europaea.