electronics

MDPI

Article

Multi-Task Learning and Temporal-Fusion-Transformer-Based Forecasting of Building Power Consumption

Wenxian Ji [1], Zeyu Cao [2] and Xiaorun Li [1,*]

[1] College of Electrical Engineering, Zhejiang University, 866 Yuhangtang Rd., Hangzhou 310058, China; 11910094@zju.edu.cn
[2] School of Spatial Planning and Design, Hangzhou City University, 51 Huzhou Street, Hangzhou 310015, China; caozy@hzcu.edu.cn
* Correspondence: lxr@zju.edu.cn

Abstract: Improving the accuracy of the forecasting of building power consumption is helpful in reducing commercial expenses and carbon emissions. However, challenges such as the shortage of training data and the absence of efficient models are the main obstacles in this field. To address these issues, this work introduces a model named MTLTFT, combining multi-task learning (MTL) with the temporal fusion transformer (TFT). The MTL approach is utilized to maximize the effectiveness of the limited data by introducing multiple related forecasting tasks. This method enhances the learning process by enabling the model to learn shared representations across different tasks, although the physical number of data remains unchanged. The TFT component, which is optimized for feature learning, is integrated to further improve the model's performance. Based on a dataset from a large exposition building in Hangzhou, we conducted several forecasting experiments. The results demonstrate that MTLTFT outperforms most baseline methods (such as LSTM, GRU, N-HiTS) in terms of Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE), suggesting that MTLTFT is a promising approach for the forecasting of building power consumption and other similar tasks.

Keywords: power consumption forecasting; multi-task learning; deep learning; time series analysis; intelligent building

Citation: Ji, W.; Cao, Z.; Li, X. Multi-Task Learning and Temporal-Fusion-Transformer-Based Forecasting of Building Power Consumption. *Electronics* **2023**, *12*, 4656. https://doi.org/10.3390/electronics12224656

Academic Editors: Katia Lida Kermanidis, Phivos Mylonas and Manolis Maragoudakis

Received: 6 October 2023
Revised: 13 November 2023
Accepted: 14 November 2023
Published: 15 November 2023

1. Introduction

In order to slow down the trend of global warming and protect the Earth's ecological environment, more and more countries and regions are committed to mitigating the greenhouse effect by reducing carbon emissions. A significant portion of carbon emissions is produced through the use of fossil fuels, which provide electricity for commercial and domestic needs. As the primary consumers of electricity, commercial buildings hold great potential for reducing carbon emissions and alleviating global warming [1]. If the power consumption of commercial buildings can be accurately predicted without affecting normal commercial demand, building operators can purchase electricity according to actual demand [2] and reduce the cost of electricity while enabling power suppliers to allocate electricity more efficiently, reducing waste. Thus, the accurate prediction of power consumption in commercial buildings has garnered increasing attention from researchers.

The task of forecasting power consumption can be considered as a subfield of time series analysis, which has been extensively studied by researchers. Time series analysis aims to understand the underlying structure and patterns within sequential data points, making it a vital tool in various applications, from finance to meteorology. Statistical methods such as Auto-Regressive (AR) and Auto-Regressive Integrated Moving Average (ARIMA) have been successfully used by Box and Jenkins [3] in the economic sphere. The AR and ARIMA models, in particular, leverage the correlation between successive data points in a time series to make predictions. Their popularity in economic forecasting is

attributed to their ability to capture and model seasonality, trends, and other patterns in the data. These methods usually have a complete theoretical derivation process and modeling steps that are suitable for data with a priori knowledge or empirical assumptions. However, these methods may not be suitable for complex nonlinear data. The nonlinearity in data can arise from various factors such as abrupt changes, external influences, or inherent complexities in the underlying system. Traditional statistical methods might struggle to capture these nonlinearities, leading to suboptimal forecasting performance. Hence, the need for alternative methodologies that can handle such complexities becomes evident.

To better predict complex nonlinear time series, many machine learning (ML) methods have been used in the forecasting of building energy consumption, such as artificial neural networks (ANNs) [4], gene expression programming (GEP) [5], and support vector regression (SVR) [6]. With the help of ML, more complex distributions can be modeled, resulting in increased accuracy in the forecasting of power consumption. In recent years, the rise of deep learning [7,8] has further improved the accuracy of time series forecasting. Various neural network modules have been introduced for time series analysis, including convolutional neural networks (CNsN) [9], recurrent neural networks (RNNs) [10], and long-short term memory (LSTM) networks [11]. These different neural network models have various advantages for feature extraction and pattern recognition. As a result, some methods have utilized several modules for the forecasting of power consumption, such as CNN-LSTM [12] and RNN-LSTM [13]. These hybrid models take advantage of different modules and have achieved impressive prediction results.

In recent years, a novel module named Transformer [14] has garnered significant attention for its prowess in handling time series data. Unlike traditional methods, the Transformer is built upon a unique architecture that leverages an attention mechanism. This attention mechanism is pivotal in understanding the importance of different features within a dataset. It works by dynamically allocating varying weights to different features based on their relevance in a given context. As a result, the Transformer can discern and emphasize critical patterns while de-emphasizing less relevant ones. This capability not only enhances the accuracy of time series predictions but also provides insights into the underlying structure of the data. With its ability to assign different weights to distinct features, the Transformer has exhibited remarkable feature-learning ability, setting it apart from many conventional models.

Many Transformer-based algorithms have been proposed for time series forecasting, such as [15], which used Transformer for influenza prevalence analysis [16], which addressed the memory bottleneck of the Transformer for time series forecasting; and [17], which improved the Transformer for long sequence time series forecasting. For the forecasting of power consumption, some works have adopted the Transformer, such as [18], which proposed a Transformer-based model for power consumption prediction and anomaly detection, and [19], which combined Transformer and Light Gradient-Boosting Machine (Light-GBM) [20] for medium-term power consumption forecasting.

While deep learning approaches have shown remarkable outcomes, they are widely recognized for their extensive data demands [21–23]. Nevertheless, in practical settings, energy consumption behaviors tend to be distinct among various buildings. Such distinctions suggest that data might be somewhat limited when designing a prediction model specific to a single structure. Therefore, it can be challenging to directly utilize deep learning models for building power consumption forecasting [24]. To solve the data-deficiency problem in deep learning algorithms, multi-task learning (MTL) [25] has been considered as a promising area. MTL aims to improve the performance of multiple related learning tasks by leveraging useful information among them. By learning multiple tasks, models can capture more important information for the main task, especially when data are limited. Several works have been performed to adopt MTL in deep learning for time series forecasting. For instance, [26] adopted MTL for univariate time series forecasting, [27] improved the time series forecasting results by fusing near and distant future visions, and [28] used old data to transfer useful knowledge to current prediction for better time series prediction results.

Motivated by the challenges of forecasting building power consumption with constrained data, we identified a gap in existing methodologies. While there are techniques that perform adequately, they often do not fully harness the potential of limited datasets, especially when faced with complex nonlinear consumption patterns specific to individual buildings. To address this, we introduced a novel approach: multi-task learning-based temporal fusion transformers (MTLTFTs). Drawing inspiration from both the Transformer and MTL methods, MTLTFT assigns forecasting tasks that span from immediate to distant future predictions to the temporal fusion transformers. This strategy ensures the optimal utilization of the available data. Coupled with the efficiency of the transformer model, the MTLTFT method offers enhanced precision in forecasting building power consumption, presenting a significant advancement over traditional methods.

The paper is organized into six sections: introduction, related works, dataset, methodology, experiments and analysis, and conclusion. We introduce the background and the motivation of our method in the introduction section. The original methods and related works are briefly introduced in the related works section. We show an original dataset in the dataset section. The details of MTLTFT are shown in the methodology section. The details of experiments and corresponding analysis are contained in the experiments and analysis section. Lastly, the summary of our method is given in the conclusion section.

2. Related Works

2.1. Multi-Task Learning

Multi-task learning (MTL) aims to train multiple tasks in parallel, enhancing the performance of the main task through the incorporation of training signals from other related tasks [29]. By introducing auxiliary tasks, models can learn from both the main and auxiliary tasks, yielding improved forecasting results. The selection of auxiliary tasks is a critical aspect of MTL.

For time series forecasting, various approaches have been explored. The Multi-Level Construal Neural Network (MLCNN) [27] posits that near and distant future forecasting can serve as beneficial auxiliary tasks. Reference [30] employed a deep multi-task learning framework to forecast air quality by integrating data from different but related tasks. In another study, ref. [31] utilized MTL for electricity load forecasting, while the auxiliary task was to predict the outdoor temperature. The authors of [32] provide a comprehensive survey of MTL methodologies, underlining their effectiveness across different domains, including time series forecasting.

By leveraging information from auxiliary tasks, MTL is demonstrated to enhance prediction accuracy and model robustness, which is particularly valuable in time series forecasting.

2.2. Temporal Fusion Transformer

The temporal fusion transformer (TFT) is an attention-based deep neural network designed for multi-horizon time series forecasting [33]. While many attention-based models exist for time series forecasting, few of them are both highly accurate and interpretable. TFT addresses this issue by modifying the architecture of the original transformer and introducing interpretable multi-head attention. Additionally, TFT uses gating mechanisms to skip over unused components and implements a variable selection network to identify relevant input variables, resulting in improved forecasting accuracy. TFT also emphasizes the need to treat different input variables separately, including static and dynamic variables. The effectiveness of TFT for time series forecasting has been demonstrated in experiments [21,34], and as such, we have chosen to incorporate TFT into our MTLTFT model for forecasts of building power consumption.

3. Dataset

For the experiment, we obtained a dataset on building power consumption from the Hangzhou International Expo Center (HIEC) in China. HIEC covers a total floor area of 850,000 square meters, and by the end of December 2022, HIEC had witnessed more

than 7400 conferences and more than 260 exhibitions. Hangzhou, where HIEC is located, experiences a humid subtropical climate characterized by four distinct seasons: a warm and humid spring, a hot and humid summer with frequent rainfalls, a cool and clear autumn, and a cold and cloudy winter. This climate information is crucial as weather variations significantly influence building power consumption patterns. As a building with great power consumption, HIEC is a good research object for the forecasting of power consumption. Specifically, we collected the day-level electricity load information for HIEC for nearly three years (1 September 2019–30 December 2022). The center consists of three parts: conference, hotel, and exhibition areas. These areas are used for different purposes, so their electricity load patterns may vary. However, we found similar patterns of power consumption on air conditioners in different areas. It is important to note that in our analysis, we focused primarily on the consumption patterns of air conditioners, excluding other electricity consumption patterns. Furthermore, it was observed that the consumption of air conditioners accounted for approximately 40% of the total electricity usage across these areas. This significant proportion highlights the relevance of our focus on air conditioning systems in understanding the overall energy consumption at the center. So we recorded the air conditioning power consumption in these three areas as part of the dataset. We also recorded weather history information from China Meteorological Data Service Center because knowing the temperature range in the future can be helpful in forecasting the power consumption of air conditioners. Specifically, we recorded the maximum and minimum temperatures for each day in the dataset.

A sample display of the dataset can be found in Table 1. Different areas were categorically encoded, such that 0 means conference area, 1 means hotel area, and 2 means exhibition area. In the dataset, every record can be identified by the Date and Area columns. The Consumption (kWh) column is the consumption of air conditioners, which is the target value to forecast. The Max_temperature (°C) and Min_temperature (°C) columns are auxiliary input variables to help forecasting.

We also show part of the dataset in Figure 1. In the hotel area, the consumption of air conditioners is obviously higher in summer than in other months. This pattern is similar in the other two areas. This implies that temperature is an important variable that influences the power consumption of air conditioners. Also, some anomaly points (zero values) in the dataset are shown in Figure 1, so we cleaned the data before building the training and testing set. For those points with zero values, we replaced them with the mean of the neighboring normal values within a 15-day range. In our observations of the data, we have not come across a situation where there are zero values persisting for a consecutive span of fifteen days. Thus, this filling method is feasible and does not pose any issues.

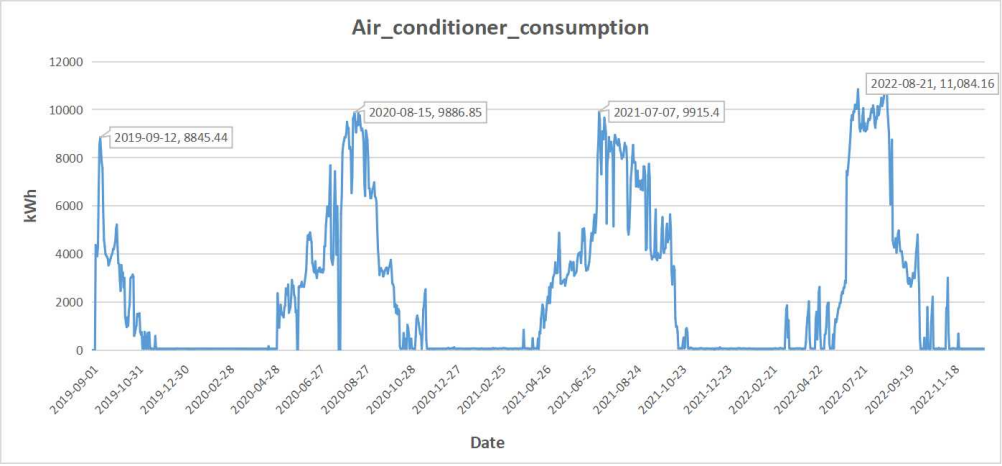

Figure 1. Daily consumption of the air conditioners in the hotel area.

Table 1. Daily samples of the dataset used in the paper.

Date	Consumption (kWh)	Max_Temperature (°C)	Min_Temperature (°C)	Area
5 January 2022	59.39	11	6	1
16 November 2021	667.93	19	10	2
4 May 2020	2366.46	36	20	1
15 November 2021	584.63	19	9	2
3 October 2021	5894.18	34	21	0

4. Methodology

We adopted a temporal fusion transformer and multi-task learning to deal with the real data for building power consumption. We divided the datasets into two primary parts: the training set and the testing set. Specifically, the proportion allocated for the training set was 80%, while the testing set constituted 20% of the total data. It is essential to note that we utilized 20% of the training set specifically as a validation set. This segmentation was pivotal as it ensured that we had a dedicated subset for model evaluation and fine-tuning. Subsequently, we leveraged these distinct sets to generate the requisite data for the various tasks inherent to our multi-task learning approach.

4.1. Simplified Temporal Fusion Transformer Model

Considering the feature of our dataset, we used a simplified temporal fusion transformer for the forecasting. The structure of our TFT model is shown in Figure 2. The original TFT model is designed to be good at extracting information from static data. However, there are no static data in our dataset, so we simplified the TFT model. Suppose we want to use historical information from the past k days to forecast the power consumption in the next τ days. Then, the inputs can be divided into past inputs and known future inputs. Past inputs are the consumption in the past k days. Known future inputs are the variables that can be known in advance, such as temperature values and time index in the next τ_{max} days. Generally, τ_{max} can be different from τ, but we set them as the same in our experiment for convenience.

Past inputs are fed into a series of Long Short-Term Memory (LSTM) encoders [35] for feature extraction. Similarly, known future inputs are fed into a series of LSTM decoders for further feature extraction. With these LSTMs, the time-dependent inputs are locally processed for temporal self-attention computing. The temporal self-attention mechanism [33] is a modified multi-head attention in transformer-based architectures designed to enhance explainability. This module evaluates the importance of each vector in the input features, that is, the attention, using three variables. Before computing the temporal self-attention, the hidden states of LSTMs are processed by the Gate and Add&Norm layers. The Gate layer represents the gating layer, which is based on Gated Linear Units (GLUs) [36]. Gating layers provide flexibility to suppress any parts of the architecture that are not required for a given dataset. Equation (1) shows the mathematical expression of GLUs, where X represents the input of the Gated Linear, W_1, W_2 are learnable weight matrix parameters, b_1, b_2 are corresponding bias parameters, $\sigma(.)$ is the sigmoid activation function, and $\odot$ is the element-wise Hadamard product.

$$\mathrm{GLU}(\boldsymbol{X}) = \sigma(\boldsymbol{W}_1\boldsymbol{X} + \boldsymbol{b}_1) \odot (\boldsymbol{W}_2\boldsymbol{X} + \boldsymbol{b}_2) \tag{1}$$

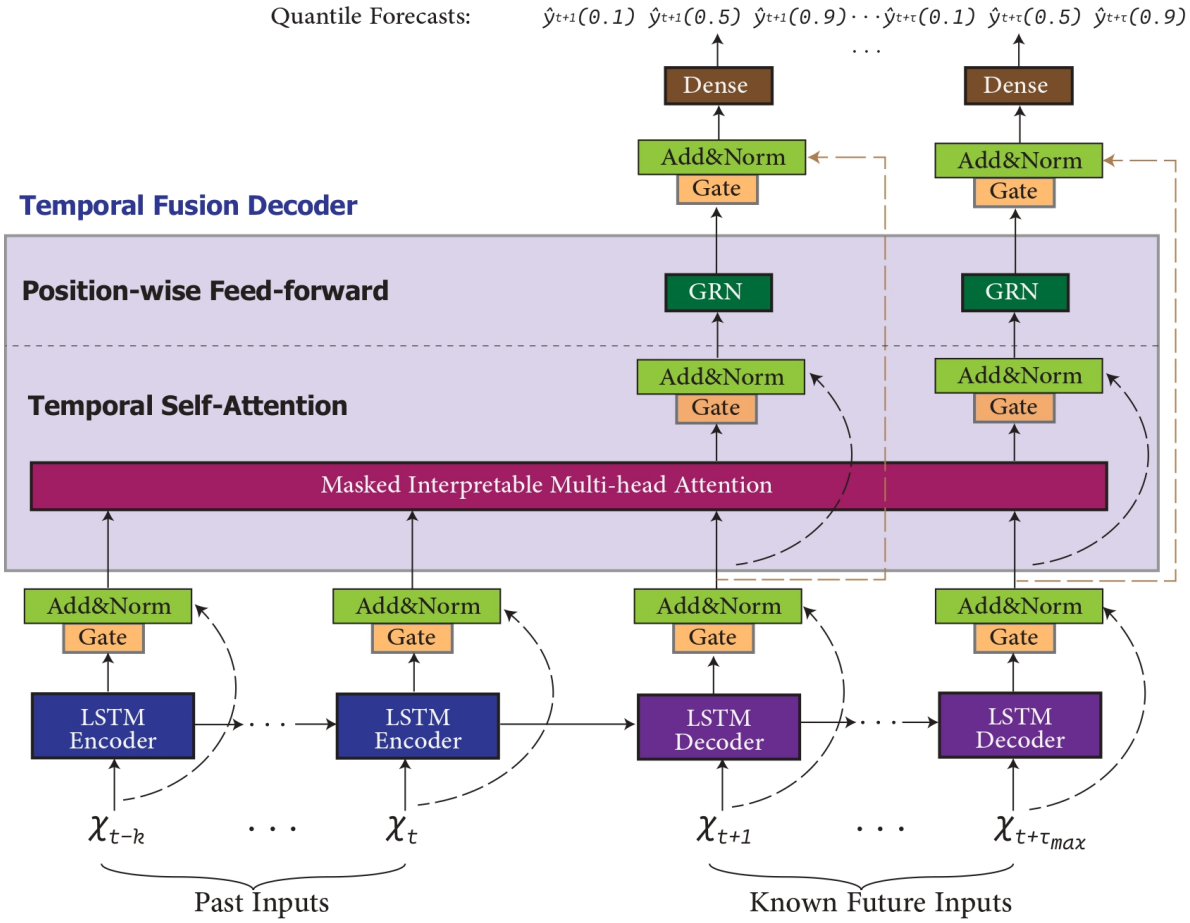

Figure 2. Structure of simplified temporal fusion transformer. GRN represents gated residual network blocks that enable efficient information flow with skip connections and gating layers. Time-dependent processing is based on LSTMs for local processing, and multi-head attention for integrating information from any time step.

Add&Norm means the combination of residual connection and layer normalization [37]. This layer has proven efficient for feature extraction in various transformer structures. Furthermore, the gated residual network (GRN) was proposed to give the model the flexibility to apply non-linear processing only where needed. Equations (2)–(4) illustrate the GRN. LayerNorm(.) is standard layer normalization, $\boldsymbol{a}, \boldsymbol{c}$ are the inputs of the GRN, $\boldsymbol{a}$ is seen as the primary input, and $\boldsymbol{c}$ is seen as an optional context vector. ELU is the Exponential Linear Unit function [38], and $\eta_1 \in \mathbb{R}^{d_{\text{model}}}, \eta_2 \in \mathbb{R}^{d_{\text{model}}}$ are intermediate layers. And $\boldsymbol{W}_3, \boldsymbol{W}_4, \boldsymbol{W}_5$ are weight matrix parameters, while $\boldsymbol{b}_3, \boldsymbol{b}_4$ are the corresponding bias parameters.

$$\text{GRN}(\boldsymbol{a}, \boldsymbol{c}) = \text{LayerNorm}(\boldsymbol{a} + \text{GLU}(\boldsymbol{\eta}_1)), \tag{2}$$

$$\eta_1 = \boldsymbol{W}_3\eta_2 + \boldsymbol{b}_3 \tag{3}$$

$$\eta_2 = \text{ELU}(\boldsymbol{W}_4\boldsymbol{a} + \boldsymbol{W}_5\boldsymbol{c} + \boldsymbol{b}_4) \tag{4}$$

GRN enables efficient information flow with skip connections and gating layers. Except for the GRN, masked interpretable multi-head attention layers [33] are also special components in TFT; with the rectified multi-head attention, the attention in the TFT becomes interpretable and easy to understand. So we used the same multi-head attention layers in our model; more details can be found in the original TFT. Finally, like the TFT, we adopted quantile loss [39] as our loss function. With the help of these useful blocks, the simplified version of TFT model is still good for forecasting tasks.

The masked interpretable multi-head attention (MIMHA) mechanism enhances the model's ability to focus on different parts of the input sequence, allowing for a better understanding and interpretation of the model's behavior. This is achieved by incorporating a masking strategy and making the attention scores interpretable.

The standard multi-head attention is formulated as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{5}$$

where Q, K, and V represent the query, key, and value matrices, respectively, and d_k is the dimension of the keys.

In the MIMHA, a masking matrix M is introduced to the attention mechanism to control the information flow, making the attention scores more interpretable. The masked attention is computed as

$$\text{MaskedAttention}(Q, K, V, M) = \text{softmax}\left(\frac{QK^T + M}{\sqrt{d_k}}\right)V. \tag{6}$$

The masking matrix M is designed such that it contains large negative values where attention should be masked, forcing the softmax function to output near-zero values at these positions. This enables the model to focus its attention on the unmasked positions, making the attention scores more interpretable.

Furthermore, the MIMHA employs a rectification strategy to ensure that the attention scores are non-negative, which simplifies the interpretation of the attention mechanism. The rectified attention is computed as

$$\text{RectifiedAttention}(Q, K, V, M) = \text{relu}(\text{MaskedAttention}(Q, K, V, M)), \tag{7}$$

where relu is the Rectified Linear Unit function, which zeroes out negative values, ensuring that all attention scores are non-negative.

Through the incorporation of a masking strategy and rectification, the MIMHA provides a more interpretable and easily understandable attention mechanism, which is crucial for analyzing and debugging the model, especially in tasks where understanding the model's focus is essential for ensuring correct and reliable predictions.

4.2. Multi-Task Generation Strategy

Though the dataset we used included three years' data, it is still not enough to train a good forecasting model. As a result, we proposed a multi-task generation strategy for data augmentation, as well as model optimization. As previously mentioned in MLCNN [27], utilizing both near- and distant-future visions enhances the model's ability to learn salient features. Building upon this, our multi-task generation strategy is depicted in Figure 3.

Given a time series $X = X_{t-k+1}, \ldots X_t$ where $X_i \in \mathbb{R}^n$ and n represent the variable dimension with $t - k + 1 \geq 0$, our primary goal is to predict the value of X_{t+h}, with the horizon h being determined by specific environmental needs. To further augment our data and optimize model performance, we extend our predictions to encompass values at $X_{t+h-i}, \ldots X_{t+h+i}$. This approach not only offers a broader view of the time series progression but also introduces auxiliary tasks that bolster the primary task. The number of these tasks is contingent on the value of i. As it nears h, the tasks multiply, peaking at $2i$ when $i < h$ and then eventually consolidating to the main task at $i = 0$. This structure effectively balances the intricacies of multiple auxiliary tasks and the central aim of forecasting the principal future point.

To be more specific, in MTLTFT, we assign multiple forecasting tasks to the temporal fusion transformers, including the main task of predicting the next h steps ahead, as well as auxiliary tasks of predicting the $h - i$ steps ahead and $h + i$ steps ahead ($0 < i < h$). The objective function of MTLTFT is to jointly minimize the loss function of all tasks, which helps the model to learn from all tasks simultaneously. The auxiliary tasks can provide

additional information and constraints to help the model better capture the patterns and relationships in the data, thus improving the performance of the main task. Furthermore, the MTL framework allows the model to share the learned representations among different tasks, which can also help the model to better generalize to new data. Therefore, MTLTFT can make the best use of limited data and improve the accuracy of the forecasting of building power consumption.

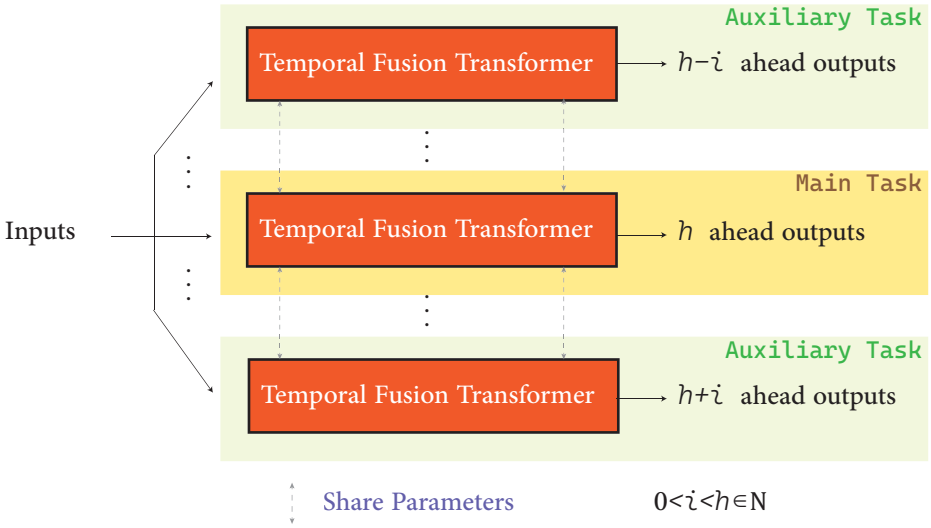

Figure 3. Illustration of the multi-task learning strategy we used.

Unlike the original MLCNN implementation, we did not design explicitly seperated parts for the main task and auxiliary tasks in the model. Instead, we shared the model weight among the auxiliary tasks and the main task, hoping that the multi-task generation can also act as a data-augmentation method. In the training phase, we directly put all the data segments together for training. This way, more data can be used for model training, improving the performance of the model.

4.3. Integration of Multi-Task Learning with Temporal Fusion Transformer

Multi-task learning (MTL) is an approach in which a model is trained on multiple tasks simultaneously, leveraging shared information among tasks to improve generalization. In the context of our research on forecasting building power consumption, MTL allows the model to predict various future lengths, such as 3 days, 5 days, and 7 days.

To integrate MTL with the simplified TFT, we modified the architecture to have multiple output heads, each dedicated to a specific forecasting task. Each head employs the quantile loss function, ensuring that the model provides accurate forecasts across different quantiles. The shared encoder layers, consisting of LSTMs and the temporal self-attention mechanism, extract common temporal features that are beneficial for all tasks. These shared features are then fed into task-specific decoders to produce forecasts for each task.

The main advantage of this integration is the regularization effect that MTL brings. By training on multiple tasks, the model is less likely to overfit the idiosyncrasies of a single task, leading to better generalization on unseen data. Moreover, leveraging shared temporal patterns across tasks can potentially improve the forecasting accuracy of the main task.

The combined MTLTFT model's structure can be visualized as a TFT model with multiple parallel output branches, each corresponding to a different forecasting task. These branches share the same encoder layers but have separate decoders tailored to their respective tasks.

Mathematically, the combined MTL-TFT model can be represented as

$$Y_i = \mathrm{TFT}_{\mathrm{decoder}_i}(\mathrm{TFT}_{\mathrm{encoder}}(X)) \quad \text{for} \quad i = 1, \ldots, T \tag{8}$$

where Y_i is the forecast for the ith task, T is the total number of tasks, and X is the input data. The shared encoder is represented by $\text{TFT}_{\text{encoder}}$, and the task-specific decoders are represented by $\text{TFT}_{\text{decoder}_i}$.

In our experiments, we observed that the integrated MTL-TFT model outperformed single-task models in terms of forecasting accuracy and robustness, especially when there were limited data for individual tasks.

5. Experiments and Analysis

5.1. Experiments Setup

To validate the utility of the proposed MTLTFT model, we used four baseline methods for comparison in the experiments. First, we adopted a simple model that uses the last known target value to make a prediction, named Baseline. Then, we adopted neural hierarchical interpolation (N-HiTS) [40], a designed LSTM network [11] and a designed gated recurrent unit (GRU) neural network [41] for comparison. LSTM and GRU are both types of recurrent neural networks (RNNs) that have been specifically designed to address the problem of vanishing gradients in traditional RNNs. They both use gating mechanisms to selectively forget or remember information from the past, and this helps them to maintain long-term dependencies in the time series. In contrast, N-HiTS does not use RNNs but instead uses a series of convolutional neural networks (CNNs) and fully connected layers to extract features from the input data. N-HiTS has been shown to perform better than LSTM on datasets with multiple time series and complex dependencies between them. GRU is also capable of capturing long-term dependencies but has been found to be less effective than LSTM in some cases. These methods are widely recognized as good deep learning algorithms for time series forecasting, so we used them to illustrate the advantages of MTLTFT.

In order to provide a comprehensive understanding of the differences between and similarities among the methods utilized, a comparison has been summarized in Table 2. This table outlines the type of models, the proportion of training data used, the nature of features (whether they are based on past inputs or also incorporate future known inputs), and the level of parameter complexity for each method.

All experiments were implemented on a personal computer with 32 GB RAM and a RTX 3090ti GPU. The coding environment was Pytorch [42]. We repeated all the experiments five times and recorded the average results.

Table 2. Comparison of methods used for time series forecasting.

Method	Model Type	Training %	Features Selection	Parameters
Baseline	-	0	-	Minimal
N-HiTS [40]	CNN	80	Past and Future known combined	Moderate
LSTM [11]	RNN	80	Past and Future known combined	High
GRU [41]	RNN	80	Past and Future known combined	High
MTLTFT	RNN, Transformer	80	Past and Future known seperated	High

The evaluation metrics are Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE). RMSE is a measure of the average deviation of the predicted values from the actual values. As shown in Equation (9), RMSE is computed with three variables: y_i is the actual value of the i-th observation, $\hat{y}_i$ is the predicted value of the i-th observation, and n is the total number of observations.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{9}$$

MAPE is a measure of the percentage difference between the predicted and actual values. Equation (10) shows how MAPE is computed, y_i is the actual value of the i-th observation, $\hat{y}_i$ is the predicted value of the i-th observation, and n is the total number of observations

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{10}$$

To extend these metrics for relative comparison, Relative RMSE and Relative MAPE are utilized. Relative RMSE is calculated by comparing the RMSE of the proposed model to that of a baseline model. It quantifies the improvement in or degradation of the proposed model over the baseline. The formula for Relative RMSE is given in Equation (11), where $\text{RMSE}_{\text{model}}$ is the RMSE of the proposed model, and $\text{RMSE}_{\text{baseline}}$ is the RMSE of the baseline model.

$$\text{Relative RMSE} = \frac{\text{RMSE}_{\text{model}} - \text{RMSE}_{\text{baseline}}}{\text{RMSE}_{\text{baseline}}} \tag{11}$$

Similarly, Relative MAPE is defined as the MAPE of the proposed model in relation to the MAPE of a baseline model. This metric offers a way to compare the percentage error of the model against a standard baseline, providing a relative scale of error. The equation for Relative MAPE is given in Equation (12), with $\text{MAPE}_{\text{model}}$ representing the MAPE of the proposed model, and $\text{MAPE}_{\text{baseline}}$ representing the MAPE of the baseline model.

$$\text{Relative MAPE} = \frac{\text{MAPE}_{\text{model}} - \text{MAPE}_{\text{baseline}}}{\text{MAPE}_{\text{baseline}}} \tag{12}$$

5.2. Results

The forecasting experimental results are shown in Table 3. Obviously, simply using the last known value as the prediction value is not good enough in our dataset. So Baseline obtained the worst results in terms of both two metrics. Four other deep learning models, including MTLTFT, achieved better results with RMSE lower than 1800. Furthermore, MTLTFT achieved the smallest RMSE of 1761.15, outperforming all the methods. Although MTLTFT was the best in terms of RMSE, N-HiTS achieved the smallest MAPE of 1.14. This shows that N-HiTS is good at minimizing the percentage errors between the actual values and the forecast values. However, MTLTFT achieved a close MAPE of 1.18, implying that MTLTFT makes a good trade-off between MAPE and RMSE.

Table 3. A comparison of forecasting results, including RMSE, relative RMSE, MAPE, and relative MAPE. The best results in each row are shown in bold.

Method	**Baseline/%**		**N-HiTS/%**		**LSTM/%**		**GRU/%**		**MTLTFT/%**	
RMSE	2292.52	0%	1777.24	−22%	1796.1	−22%	1783.8	−22%	**1761.2**	**−23%**
MAPE	1.84	0%	**1.14**	**−38%**	1.46	−21%	1.51	−18%	1.18	−36%

While N-HiTS excels at minimizing percentage errors, MTLTFT demonstrates superior capability in minimizing larger absolute errors, which, as previously mentioned, is paramount in our application of reducing building power consumption. It is crucial to note that RMSE and MAPE have different sensitivities; RMSE is sensitive to larger errors due to its squaring property, while MAPE focuses on percentage discrepancies. In contexts like ours, where larger forecast errors have more severe implications, RMSE provides a more pertinent assessment of model performance. That said, we acknowledge the merit of N-HiTS, especially in scenarios where percentage error is a focal point of evaluation. Our goal is to reduce the consumption of the building power, so we should pay more attention to larger forecast errors. As a result, MTLTFT shows its great potential for building power consumption forecasting in the experiments.

To further analyze the features of MTLTFT, we recorded the training process of MTLTFT in Figure 4. In this process, quantile losses [43] were computed on the train-

ing set and validation set, and thus, we obtained training loss curves and validation loss curves. According to the curves, the model costs about 200 epochs for convergence. With the progress of training, training loss and validation loss decreased gradually and they became close in the end. This means that the distributions of the training set and validated set are similar and the model may have good generalization.

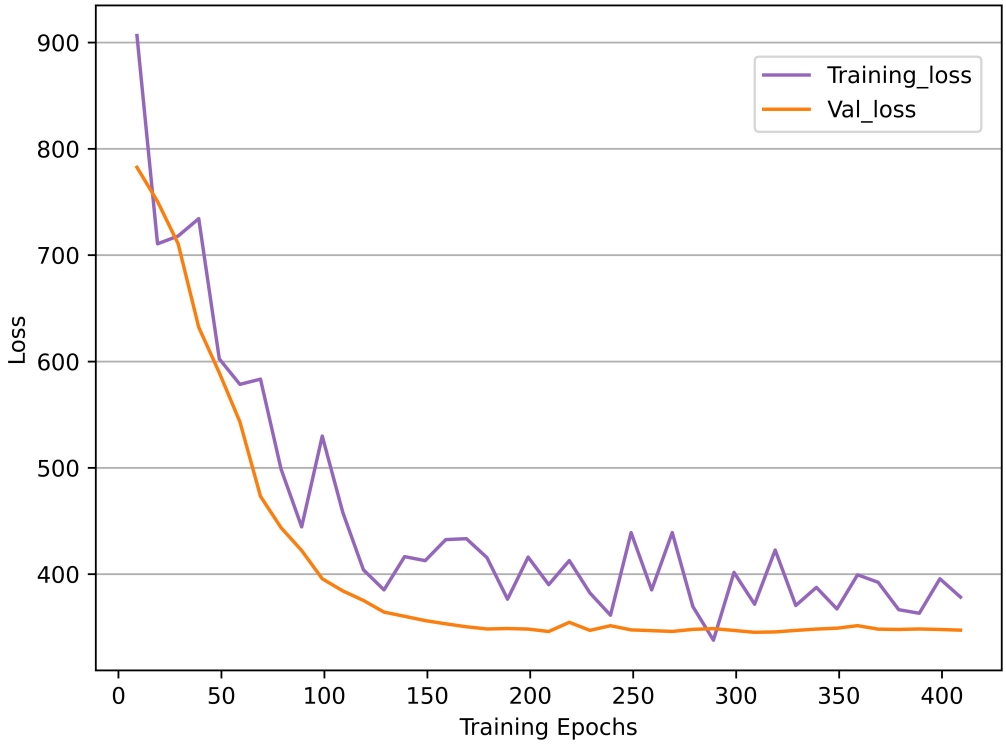

Figure 4. Training and validation loss over 400 epochs of MTLTFT.

However, the validation loss is a bit smaller than the training loss. This may be because the model performed better on the unseen validation set. We hold the opinion that a shortage of data is the reason that the validation loss is lower than the training loss. Data in the validation set are too similar to the training set, so it is too easy for the model to forecast on the validation set. As a result, we still need more data or data-augmentation methods for further exploration.

For further illustration, we show two visualizations of the forecasting results in Figure 5 (larger error) and Figure 6 (smaller error). In these figures, blue curves are actual values and orange curves are predicted values. Grey curves represent the attention values. The attention values are the features generated in Masked Interpretable Multi-head Attention part in Figure 2, which has the same length as the input variables. Different shades and colors indicate the probability of the predicted value produced by the model during the prediction. The deeper the color is, the larger the probability is. According to the figures, MTLTFT pays more attention to the increase in the input and responds to it in the prediction. The prediction of the model tends to be in the middle of the actual value range. This is why MTLTFT can obtain the smallest RMSE value and the best forecasting accuracy. However, MTLTFT is still not able to fit the actual value perfectly because of the shortage of data.

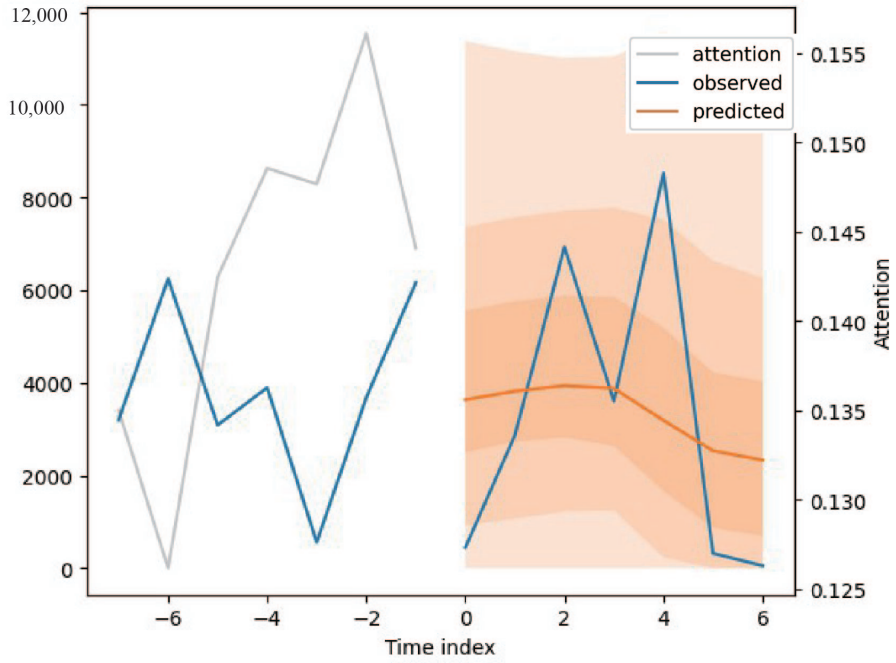

Figure 5. A forecasting example for the testing results of MTLTFT (larger error).

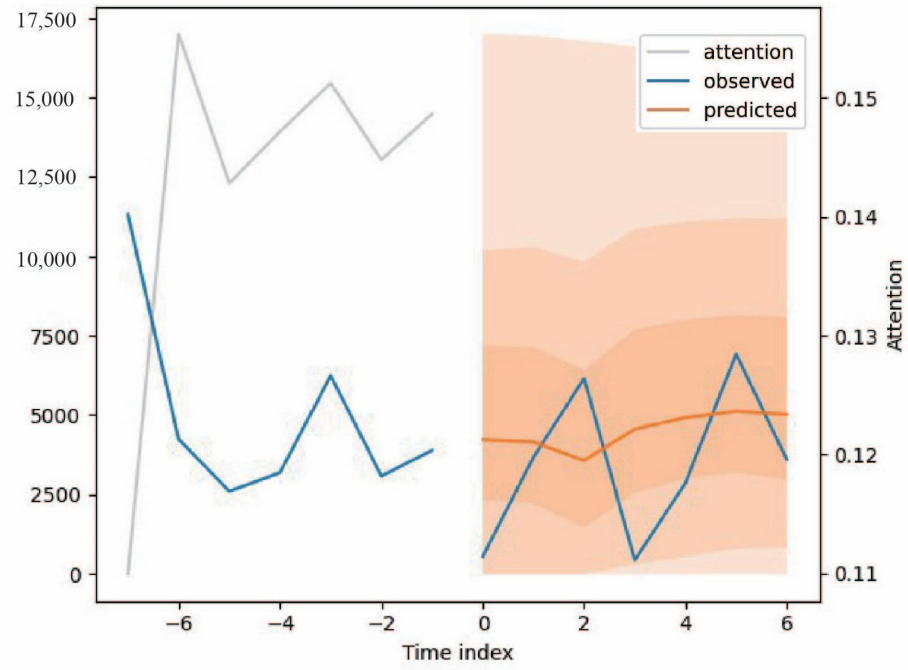

Figure 6. A forecasting example for the testing results of MTLTFT (smaller error).

6. Conclusions

In this research endeavor, we have pioneered the introduction of MTLTFT, a groundbreaking algorithm combining the robustness of a multi-task learning approach with the precision of temporal fusion transformers, optimized specifically for forecasting building power consumption. Our salient contributions encompass the following:

1. Innovating a multi-task learning technique that serves dual purposes: data augmentation and efficient model training;
2. Customizing temporal fusion transformers to cater specifically to the nuances of building power consumption forecasting;

3. Authenticating the efficacy of MTLTFT through rigorous validation on a real-world dataset, establishing its superiority in the domain.

Furthermore, our endeavors led to the compilation of a distinctive dataset on building power consumption, sourced from the esteemed Hangzhou International Expo Center. Empirical assessments revealed that MTLTFT achieved an RMSE of 1761.2 and an MAEPE of 1.18, underscoring its unparalleled potential in this forecasting arena. However, it is worth noting the challenges posed by data paucity for specific buildings, which inadvertently impacts the predictive precision. As we chart our future research trajectory, our emphasis will be on exploring a myriad of data augmentation strategies, aiming to maximize the utility of scarce datasets.

Author Contributions: W.J.: conceptualization, methodology, data preprocessing, and writing-original draft preparation; Z.C.: visualization, investigation, experimental training, and testing; X.L.: supervision and reviewing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Zhejiang Provincial Natural Science Foundation of China under Grant No. LGG22F030008 and by the Key Research and Development Projects of 'Vanguard' and 'Leading Goose' in Zhejiang Province under Grant No. 2023C01129.

Data Availability Statement: Data can be obtained upon request from the authors.

Acknowledgments: We extend our gratitude to the chief editor and the anonymous reviewers for their invaluable contributions to the quality of this manuscript.

Conflicts of Interest: The 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.

Abbreviations

The following abbreviations are used in this manuscript:

MTL	Multi-task Learning
TFT	Temporal Fusion Transformer
RNN	Recurrent Neural Networks
LSTM	Long Short-Term Memory
SVM	Support Vector Machine
GRU	Gated Recurrent Unit

References

1. Ürge-Vorsatz, D.; Cabeza, L.F.; Serrano, S.; Barreneche, C.; Petrichenko, K. Heating and cooling energy trends and drivers in buildings. *Renew. Sustain. Energy Rev.* **2015**, *41*, 85–98. [CrossRef]
2. Li, W.; Xu, P.; Lu, X.; Wang, H.; Pang, Z. Electricity demand response in China: Status, feasible market schemes and pilots. *Energy* **2016**, *114*, 981–994. [CrossRef]
3. Box, G.E.; Jenkins, G.M.; Reinsel, G.C.; Ljung, G.M. *Time Series Analysis: Forecasting and Control*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
4. Moon, J.; Park, S.; Rho, S.; Hwang, E. A comparative analysis of artificial neural network architectures for building energy consumption forecasting. *Int. J. Distrib. Sens. Netw.* **2019**, *15*, 1550147719877616. [CrossRef]
5. Kaboli, S.H.A.; Fallahpour, A.; Kazemi, N.; Selvaraj, J.; Rahim, N. An expression-driven approach for long-term electric power consumption forecasting. *Am. J. Data Min. Knowl. Discov.* **2016**, *1*, 16–28.
6. Chen, Y.; Mao, B.; Bai, Y.; Feng, Y.; Li, Z. Forecasting traction energy consumption of metro based on support vector regression. *Syst. Eng. Eory Pract.* **2016**, *36*, 2101–2107.
7. Mehta, C.; Chandel, N.; Dubey, K. Smart Agricultural Mechanization in India—Status and Way Forward. In *Smart Agriculture for Developing Nations: Status, Perspectives and Challenges*; Springer: Singapore, 2023; pp. 1–14.
8. Verma, J. Deep Technologies Using Big Data in: Energy and Waste Management. In *Deep Learning Technologies for the Sustainable Development Goals: Issues and Solutions in the Post-COVID Era*; Springer: Singapore, 2023; pp. 21–39.
9. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. In Proceedings of the 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August 2017; pp. 1–6.
10. Medsker, L.R.; Jain, L. Recurrent neural networks. *Des. Appl.* **2001**, *5*, 64–67.

11. Graves, A.; Graves, A. Long short-term memory. In *Supervised Sequence Labelling with Recurrent Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 37–45.
12. Shao, X.; Pu, C.; Zhang, Y.; Kim, C.S. Domain fusion CNN-LSTM for short-term power consumption forecasting. *IEEE Access* **2020**, *8*, 188352–188362. [CrossRef]
13. Yuniarti, E.; Nurmaini, N.; Suprapto, B.Y.; Rachmatullah, M.N. Short term electrical energy consumption forecasting using rnn-lstm. In Proceedings of the 2019 International Conference on Electrical Engineering and Computer Science (ICECOS), Batam, Indonesia, 2–3 October 2019; pp. 287–292.
14. Han, K.; Xiao, A.; Wu, E.; Guo, J.; Xu, C.; Wang, Y. Transformer in transformer. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 15908–15919.
15. Wu, N.; Green, B.; Ben, X.; O'Banion, S. Deep transformer models for time series forecasting: The influenza prevalence case. *arXiv* **2020**, arXiv:2001.08317.
16. Li, S.; Jin, X.; Xuan, Y.; Zhou, X.; Chen, W.; Wang, Y.X.; Yan, X. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
17. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 2–9 February 2021; Volume 35, pp. 11106–11115.
18. Zhang, J.; Zhang, H.; Ding, S.; Zhang, X. Power consumption predicting and anomaly detection based on transformer and K-means. *Front. Energy Res.* **2021**, *9*, 779587. [CrossRef]
19. Yang, G.; Du, S.; Duan, Q.; Su, J. A Novel Data-Driven Method for Medium-Term Power Consumption Forecasting Based on Transformer-LightGBM. *Mob. Inf. Syst.* **2022**, *2022*, 5465322. [CrossRef]
20. Qi, M.L. A highly efficient gradient boosting decision tree. In *Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2017.
21. Lim, B.; Zohren, S. Time-series forecasting with deep learning: A survey. *Philos. Trans. R. Soc. A* **2021**, *379*, 20200209. [CrossRef] [PubMed]
22. Sai Surya Teja, T.; Venkata Hari Prasad, G.; Meghana, I.; Manikanta, T. Publishing Temperature and Humidity Sensor Data to ThingSpeak. In *Embracing Machines and Humanity Through Cognitive Computing and IoT*; Springer: Singapore, 2023; pp. 1–9.
23. Rashid, E.; Ansari, M.D.; Gunjan, V.K.; Ahmed, M. Improvement in extended object tracking with the vision-based algorithm. In *Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough: Latest Trends in AI*; Springer: Cham, Switzerland, 2020; pp. 237–245.
24. Somu, N.; MR, G.R.; Ramamritham, K. A deep learning framework for building energy consumption forecast. *Renew. Sustain. Energy Rev.* **2021**, *137*, 110591. [CrossRef]
25. Zhang, Y.; Yang, Q. An overview of multi-task learning. *Natl. Sci. Rev.* **2018**, *5*, 30–43. [CrossRef]
26. Cirstea, R.G.; Micu, D.V.; Muresan, G.M.; Guo, C.; Yang, B. Correlated time series forecasting using deep neural networks: A summary of results. *arXiv* **2018**, arXiv:1808.09794.
27. Cheng, J.; Huang, K.; Zheng, Z. Towards better forecasting by fusing near and distant future visions. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 3593–3600.
28. Ye, R.; Dai, Q. MultiTL-KELM: A multi-task learning algorithm for multi-step-ahead time series prediction. *Appl. Soft Comput.* **2019**, *79*, 227–253. [CrossRef]
29. Crawshaw, M. Multi-task learning with deep neural networks: A survey. *arXiv* **2020**, arXiv:2009.09796.
30. Chen, L.; Ding, Y.; Lyu, D.; Liu, X.; Long, H. Deep multi-task learning based urban air quality index modelling. *Proc. Acm Interact. Mob. Wearable Ubiquitous Technol.* **2019**, *3*, 1–17. [CrossRef]
31. Liu, C.L.; Tseng, C.J.; Huang, T.H.; Yang, J.S.; Huang, K.B. A multi-task learning model for building electrical load prediction. *Energy Build.* **2023**, *278*, 112601. [CrossRef]
32. Vandenhende, S.; Georgoulis, S.; Van Gansbeke, W.; Proesmans, M.; Dai, D.; Van Gool, L. Multi-task learning for dense prediction tasks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3614–3633. [CrossRef]
33. Lim, B.; Arık, S.Ö.; Loeff, N.; Pfister, T. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *Int. J. Forecast.* **2021**, *37*, 1748–1764. [CrossRef]
34. Wu, B.; Wang, L.; Zeng, Y.R. Interpretable wind speed prediction with multivariate time series and temporal fusion transformers. *Energy* **2022**, *252*, 123990. [CrossRef]
35. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
36. Dauphin, Y.N.; Fan, A.; Auli, M.; Grangier, D. Language modeling with gated convolutional networks. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 933–941.
37. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv* **2016**, arXiv:1607.06450.
38. Clevert, D.A.; Unterthiner, T.; Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv* **2015**, arXiv:1511.07289.
39. Wen, R.; Torkkola, K.; Narayanaswamy, B.; Madeka, D. A multi-horizon quantile recurrent forecaster. *arXiv* **2017**, arXiv:1711.11053.
40. Challu, C.; Olivares, K.G.; Oreshkin, B.N.; Garza, F.; Mergenthaler, M.; Dubrawski, A. N-hits: Neural hierarchical interpolation for time series forecasting. *arXiv* **2022**, arXiv:2201.12886.

41. Dey, R.; Salem, F.M. Gate-variants of gated recurrent unit (GRU) neural networks. In Proceedings of the 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 6–9 August 2017; pp. 1597–1600.
42. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. In Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
43. Koenker, R.; Hallock, K.F. Quantile regression. *J. Econ. Perspect.* **2001**, *15*, 143–156. [CrossRef]