International Journal of Sensor Networks

ISSN online: 1748-1287 - ISSN print: 1748-1279
https://www.inderscience.com/ijsnet

Assessing deep learning performance in power demand forecasting for smart grid

Hengshuo Liang, Cheng Qian, Wei Yu, David Griffith, Nada Golmie

DOI: 10.1504/IJSNET.2024.10061818

Article History:

Received:	08 October 2023
Last revised:	11 October 2023
Accepted:	12 October 2023
Published online:	30 January 2024

Assessing deep learning performance in power demand forecasting for smart grid

Hengshuo Liang, Cheng Qian and Wei Yu*

Department of Computer and Information Sciences,
Towson University,
Towson, MD 21252, USA
Email: hliang2@students.towson.edu
Email: cqian1@students.towson.edu
Email: wyu@towson.edu
*Corresponding author

David Griffith and Nada Golmie

National Institute of Standards and Technology (NIST),
Gaithersburg, MD 20899, USA
Email: david.griffith@nist.gov
Email: nada.golmie@nist.gov

Abstract: This paper addresses the issue of forecasting power demands via deep learning (DL) techniques in smart grid (SG). Assessing proper DL models for power demand forecasting requires the consideration of factors (e.g., data pre-processing, computational resource usage, the complexity of learning models). We employ a two-tiered approach to carry out both short-term and long-term forecasting. Short-term forecasting emphasises model accuracy, while long-term forecasting assesses model robustness. Our evaluations utilise temporal fusion transformers (TFT) and the neural hierarchical interpolation for time series (N-HiTS)-based predictors, tested on a publicly available dataset. Our findings confirm that while TFT and N-HiTS perform similarly in short-term forecasting tasks, TFT displays superior robustness and accuracy in long-term forecasting tasks. The TFT model requires substantial computational resources, especially video RAM (VRAM), for a longer input data stream. Conversely, N-HiTS, though less confident in long-term forecasting, is shown to be more resource-efficient for handling longer input data streams.

Keywords: deep learning; smart grid; power demand forecasting; performance assessment; sensing and communication infrastructure.

Reference to this paper should be made as follows: Liang, H., Qian, C., Yu, W., Griffith, D. and Golmie, N. (2024) 'Assessing deep learning performance in power demand forecasting for smart grid', *Int. J. Sensor Networks*, Vol. 44, No. 1, pp.36–48.

Biographical notes: Hengshuo Liang is pursuing his PhD at Towson University. His research interests include the internet of things, cyberspace security, data analytics, and computer networks. His research interests include the internet of things, cyberspace security, data analytics, and computer networks.

Cheng Qian is pursuing his PhD at Towson University. His research interests include the internet of things, cyberspace security, data analytics, and computer networks.

Wei Yu is a Professor at Towson University. His research interests include cybersecurity and privacy, cyber-physical systems/internet of things, data science and machine learning driven applications.

David Griffith is a researcher in the Wireless Networks Division at NIST. His research focus is on mathematical modelling and simulation and his work includes communications systems for the smart grid and the internet of things (IoT), public safety communications, and resource allocation for 5G wireless networks.

Nada Golmie is an IEEE and a NIST Fellow. Her research in media access control and protocols for wireless networks led to papers presented at professional conferences, journals, and contributed to international standard organisations and industry led consortia. She leads several projects related to the modelling and evaluation of future generation wireless systems and protocols.

1 Introduction

The internet of things (IoT) technologies, as information communication infrastructure enablers, have enabled various smart systems in various application domains such as energy, cities, transportation, manufacturing, healthcare, and others, leading to effective operations and managements (Shafique et al., 2020; Xu et al., 2017, 2018; Wang et al., 2022; Chen et al., 2010; Liu et al., 2019; Liang et al., 2019b; Gao et al., 2017). In the energy domain, a smart grid (SG) is an electrical power grid empowered by the IoT that provides the process, management, and analysis of both the energy and information resources in the systems (Al-Turjman and Abujubbeh, 2019). The examples of components in an SG system include advanced metering infrastructure, renewable energy integration, and demand-supply balancing via dynamic pricing, among others, providing cost-effective two-way energy services to end users (Liu et al., 2012; Xu et al., 2017; Gao et al., 2012; Ghasempour, 2019; Colak et al., 2020; Yu et al., 2015; Moulema et al., 2015).

With data collected via sensing and communication infrastructures, predicting power usage in both the short-term and long-term is critical to enable the SG system's effective monitoring and control capability. Deep learning (DL) techniques are considered as viable data analytics tools to realise effective classification, prediction, and decision maker in different problem domains (Hatcher and Yu, 2018; Liang et al., 2020; Mohammadi et al., 2018; Liang et al., 2019a; Wang et al., 2021; Xu et al., 2020; Chen et al., 2021). Long short-term memory (LSTM) is a typical recurrent neural network model to handle time-series data streams, which provides accurate power usage prediction (Li et al., 2020; Alazab et al., 2020; Sun et al., 2021). LSTM utilises the gate mechanism in an unique structure. Each input data point for the current gate contributes to the input of the next gate, thus creating a chain of input data points. Such a design helps the model to capture insightful patterns in time-streaming data.

Although LSTM effectively handles time-streaming data, it requires more memory when the length of the input data stream increases. There are two reasons behind this. First, the gate mechanism creates a longer chain of gates with more input data points, and the later gate contains all previous input data points and previous output, consuming more memory. Second, when updating the model's loss function, LSTM learning rate and performance is usually improved by batch normalisation (Ioffe and Szegedy, 2015), which globally computes the mean and standard values over a subset of all samples. If some data samples have more quality meta-data features, more memory will be required to update the loss function.

Unlike LSTM, a new model called transformer (Vaswani et al., 2017) improves the memory issue by two mechanisms:

1 *Attention mechanism:* DL models utilise an attention mechanism to focus on specific parts of the input data sequence, resulting in the improved memory of more extended data sequence and essential details (Bahdanau et al., 2014).

2 *Layer normalisation:* When updating the model's loss function, transformer uses layer normalisation to compute mean and standard values for all sample features locally, which saves a lot of computing resources and time (Ba et al., 2016).

Transformer enables the parallel data input and dynamic weight adjusting, which leads to more accurate and efficient prediction than LSTM in the forecasting of SG (Habbak et al., 2023; Sun et al., 2022).

Nonetheless, attention mechanism and fully connected layers become more computationally intensive as the forecasting horizon length increases due to their quadratic scaling in memory and computational cost. This issue is particularly pronounced with long-horizon forecasting tasks, meaning the longer length of data streams. To deal with such an issue, a different structure called fully connected neural network blocks, a type of multilayer perceptron (MLP) (Gardner and Dorling, 1998), is considered. Such a method can achieve relatively good performance in forecasting for the SG and improve scalability without using a gate or attention mechanism and consume less computing resources (Wan et al., 2015; Zheng et al., 2017).

In this paper, we assess the performance of two state-of-the-art DL models for predicting power demand in the SG system. One representative model is the temporal fusion transformers (TFT) (Nazir et al., 2023) that employs transformer approach. The other representative model is the neural hierarchical interpolation for time series (N-HiTS) (Challu et al., 2022) that utilises MLP.

Our study has made the following two contributions.

1 We define the power demand prediction problem from the perspective of DL regarding data pre-processing, computing resource usage, and the complexity of DL models. We then determine an evaluation scenario for forecasting from short-term and long-term prediction aspects. Short-term prediction is about examining the accuracy of DL models; Long-term prediction is about testing the robustness of DL models.

2 We conduct extensive experiments, considering the training of two predictors based on TFT and N-HiTS for the downstream task concerning power demand forecasting with a publicly available SG dataset. In our experiments, TFT and N-HiTS achieve similar prediction performances concerning short-term prediction. However, TFT outperforms N-HiTS in long-term prediction tasks, but TFT has a bottleneck of computing resources (i.e., TFT requires larger video RAM (VRAM) when more extended input streaming data is given). Meanwhile, N-HiTS can provide a less reliable forecasting result in long-term forecasting tasks, but it can consume significantly fewer computing resources than TFT for longer input data streams. By examining various input data stream lengths and forecasting scenarios, we examine some

of the tradeoffs that analysts must consider when choosing which model to use.

The remainder of this paper is organised as follows. In Section 2, we briefly introduce the background of the SG, TFT, and N-HiTS. We review the related research efforts in Section 3. In Section 4, we present the workflow of applying DL for forecasting power demand in the SG, define the forecasting problem from the perspective of DL, and explain the dataset and forecasting evaluation scenarios. In Section 5, we conduct experiments to evaluate the efficacy of the investigated DL techniques. Finally, we conclude the paper in Section 6.

2 Preliminaries

This section discusses the challenges related to the SG, followed by a brief introduction to N-HiTS and TFT, representative DL techniques.

2.1 Smart grid

An SG system is an electricity transmission and distribution system that leverages advanced information processing and communication technologies to enable the monitoring and control of the SG (e.g., component operating status monitoring and control, predicting the power usage demand to end users to balance demand and supply for making efficient power use). The SG system leverages sensors (such as smart meters) to monitor the power demand and adjust the power supply. However, long-term, large-scale variations in demand, and the uncertainties in these variations, will require significant adjustments in power generation, so demand forecasting is vital to SG operations. The uncertainties of the SG can be caused by the power demand affected by different environmental factors, including weather, time of the day, and unexpected natural disasters, among others. How to perform accurate and cost-effective forecasting remains an unsolved issue (Quan et al., 2019). This is because the SG, as a large distributed system, consists of complex grid topology/structure and sensors and actuators deployed in different locations. The massive amounts of data generated in the SG need to be processed with strict performance requirements depending on applications in the SG. As the SG system is complicated, performing accurate demand response, voltage control, and power flow management is critical and challenging (Mahmood et al., 2018). It is also essential to design cost-effective data analytics techniques to process data efficiently and quickly extract insightful information that can be used in the grid operations (Karimipour et al., 2019).

2.2 Temporal fusion transformer

It is a typical transformer model designed explicitly for interpretable time-series data processing such as prediction. TFT combines the strengths of LSTM and transformer with the interpretability of traditional statistical models. It effectively understands temporal dynamics and complex nonlinear relations in the data, making it highly efficient for forecasting tasks across diverse time-series data streams in various applications. There are two critical components in TFT. One is LSTM, a recurrent neural network model variant designed to handle long-term dependencies in time-series data stream (Hochreiter and Schmidhuber, 1997). Its unique architecture uses gate mechanisms to effectively capture data patterns and prevent issues such as the vanishing gradient, making it helpful in learning information from time-series data streams. The other is transformer backbone (Vaswani et al., 2017). Unlike traditional recurrent neural networks, TFT relies solely on self-attention, allowing it to process data in parallel and capture dependencies in the input data regardless of distance between elements of the input vector. This architecture makes it exceptionally effective for tasks in natural language processing, such as translation and text generation.

2.3 Neural hierarchical interpolation for time series

N-HiTS (Challu et al., 2022) is a DL model based on MLP (Taud and Mas, 2018) that utilises a hierarchical approach to interpolate and predict time-series data streams. It learns to decompose the time-series data into components at different scales, each processed by an interpolation function. It effectively understands the complex patterns and trends in the data streams and performs accurate and efficient predictions on time-series data. This design facilitates the capture of complex patterns and trends in time-series data streams. Note that while transformer models rely on attention mechanisms to understand the dependencies in the input data, irrespective of their positional distance, N-HiTS uses a hierarchical structure to process different components of time-series data. Depending on the specific implementation, N-HiTS could be more computationally efficient than transformer models, especially when dealing with very long sequences, due to its hierarchical nature.

3 Related work

This section reviews research efforts closely relevant to our study.

3.1 Demand and supply in SG

How to effectively balance demand and supply is a critical issue that needs to be resolved in the SG. Various techniques have been leveraged to ensure resource allocation to support efficient demand and supply. For example, Kong (2020) leveraged device-to-device (D2D) communication to connect SG facilities and investigated an optimisation problem to allocate radio resources so that real-time pricing (RTP) can be supported. Kement et al. (2021) proposed a privacy-aware demand response (DR) scheme to minimise the peak demand and variation in the power supply. Likewise, Belhaiza et al. (2020) proposed a game theoretical model to maintain the viability of the SG infrastructure and discussed the order relationship between the user and provider utility.

3.2 Learning in SG

DL has been proposed to improve SG's efficiency in predicting, monitoring, and control capabilities. For example, Hossain et al. (2019) explored applying big data and machine learning in the SG. They discovered that big data could improve the accuracy of analysis of power marking by leveraging DL models with SG to obtain more precise predictions for electricity demand. Puhe and Rehtanz (2022) proposed an ML-based mechanism to quickly identify the corrupted sub-grid to disconnect from the grid system. Likewise, Gómez and García (2021) leveraged the transformer to predict the power usage on SG.

4 Our approach

This section covers the workflow of SG forecasting from a DL perspective, explains the dataset used for the experiments, and introduces our designed scenarios.

4.1 Workflow

Figure 1 illustrates the three steps of applying DL in forecasting in the SG:

1 *Data collection and refinement:* The SG system can regularly gather usage and state information via a variety of sensors connected through the smart grid communication networks and store it in its data centre deployed in edge or cloud server, which then refines the data using various mechanisms, such as removing unnecessary data, filling in any missing data, and normalising the data, among others.

2 *Feature extraction and training:* The DL model begins with the training process on those refined data for forecasting tasks. We use an attention mechanism-based learning model as an example in Figure 1. After multi-tuning turns, we obtain a well-trained learning model.

3 *Forecasting demand:* The trained learning model is ready to predict power demand based on the incoming historical power usage data stream and give forecasting as an output.

After illustrating DL for demand prediction in the SG, we now define the forecasting performance of DL in the following aspects:

1 *Data pre-processing:* DL performance is greatly affected by the number of input variables, also known as features. If there are more input variables, it can positively impact the effectiveness of DL. However, too many features can result in over-fitting, while too few can lead to under-fitting.

2 *Complexity of the DL model:* The term 'complexity' refers to a model's ability to learn the information/knowledge from data, which can be affected by the model's architecture and the amount of data it is trained on. A model with more layers or neurons (known as a larger model size) may have a greater capacity to learn complex features and, therefore, be considered more complex. However, other factors, such as the depth of the model (how many layers it has), the type of layers utilised (such as convolutional or recurrent layers), and the activation functions used, can affect the complexity of the model as well.

3 *Computing resource usage:* The performance of DL is affected by the computing resources available. Complex models require more power and memory, leading to longer training times and higher costs. When resources are limited, model complexity can be restricted, resulting in instability or hindered training. Using resources efficiently is essential for ensuring the robustness of the DL model.

Figure 1 DL-based demand prediction in SG (see online version for colours)

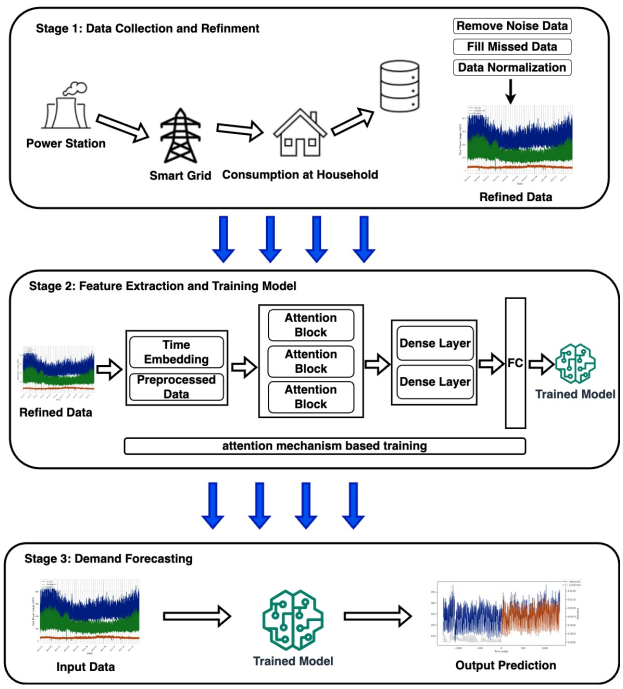


Table 1 Average daily energy usage for three targets for a month

consumer_id	*average_daily_power_usage (Kw)*
Residential	26.677
School	121.015
Factory	251.135

Table 2 Table header for the dataset

hourly_power_usage	*hours_from_start*	*days_from_start*	*date*	*consumer_id*	*hour*	*day*	*day_of_week*	*month*	*year*

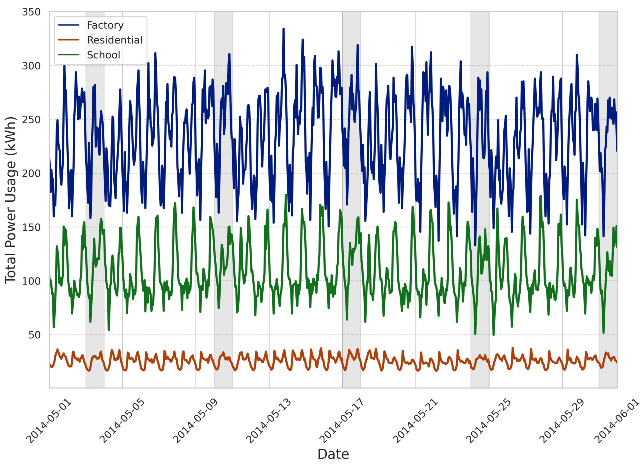


Figure 2 Example power usage (see online version for colours)

4.2 Data collection

As the supplementary dataset, we use a realistic power grid energy usage dataset (UCI Machine Learning Repository, 2015), which contains different levels of daily usage of 370 metres for four years. We choose three of those meters with significant margin usage levels. We obtain the daily mean power usage value from a month's captured data in Table 1. Then, we choose the lowest value to represent the power usage for the residential, the secondarily lowest value to represent the power usage for the school, and the highest value to represent the power usage for the factory. Figure 2 illustrates a month's power usage for residential, school, and factory, respectively. We can infer that the curves represent the weekly periodic data, and the grey bar highlights the maximum power usage each week. The output will be in three fields: *timestamps*, *consumer_id*, and *hourly_power_usage*. We present how to prepossess the data properly in Subsection 5.1.

4.3 Scenarios

Two factors affect the performance of the forecast with time series data as input:

1 *Look-back window size:* It refers to the number of previous data points considered when making forecasts in a time-series data stream.

2 *Prediction window size:* It refers to the number of future data points predicted by the model in the time-series data.

We design two scenarios for forecasting performance evaluation based on those two factors:

1 *Short-term task:* Our main objective is to assess the accuracy of DL models in making predictions. To achieve this, we conduct experiments using look-back window sizes of 1 day, 3 days, 7 days, and 28 days. We also define prediction window sizes of 1 day, 3 days, 7 days, and 28 days. To determine the impact of various combinations, we cross-combine the look-back windows and prediction window sizes and evaluate their effect on the accuracy of DL models.

2 *Long-term task:* Such a task tends to test the robustness of our DL models. For this, we use longer look-back window sizes of 1 month, 2 months, and 3 months. We also expand the prediction window sizes to 1 month, 2 months, and 3 months. We choose the same length for each pair of look-back and prediction window sizes to keep things consistent and ensure a fair comparison. We aim to see how well our models perform when making predictions over an extended period and whether they can maintain their performance stability in these conditions.

5 Performance evaluation

We conduct a comprehensive performance evaluation to validate the efficacy of two investigated DL models. In the following, we first describe our evaluation methodology and then present the results.

5.1 Methodology

5.1.1 Environments

We leverage the Pytorch (Paszke et al., 2019) as the platform for conducting experiments[1]. Our experiment uses a server with an Intel I9-9980XE CPU and an advanced NVIDIA RTX 4090 GPU. This configuration lets us easily handle the extensive computational requirements for power demand prediction tasks.

5.1.2 Data preprocessing

Based on the dataset discussed in Subsection 4.2, we decompose the feature *timestamp* into following eight features: *hours_from_start*, *days_from_start*, *date*, *hour*, *day*, *day_of_week*, *month*, and *year*. Combining this with the feature *consumer_id* and *hourly_power_usage*, we create a new table header for the dataset, such as the ones in Table 2.

5.1.3 Learning model settings

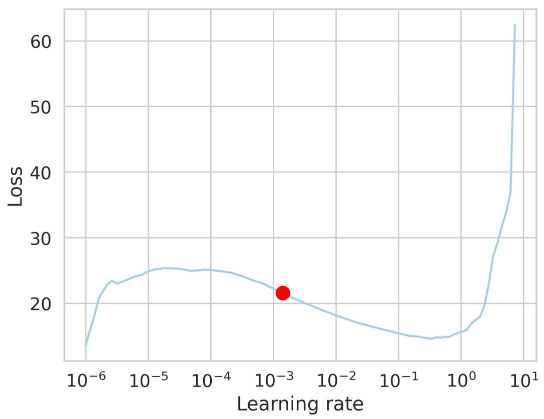
We use TFT and N-HiTS as the key DL models to evaluate forecasting performance with the following hyperparameters:

1 *TFT:* The hidden size is set to 16, attention head size to 4, dropout to 0.1, and hidden continuous size to 160. The model is trained with 3 million parameters.

2 *N-HiTS:* The hidden size is set to 64, the optimizer is Adamw (Loshchilov and Hutter, 2017), and the loss function uses MQF2DistributionLoss (Kan et al., 2022). The model is trained with 91 thousand parameters.

We utilise the default configuration for the hyperparameters. After assessing the results of our evaluation, we have found that the default setting is effective for obtaining good prediction results. Note that hyperparameter tuning could enhance the model's performance, which is not the main focus of our paper.

Figure 3 Suggested initial training learning rate (see online version for colours)



We consider two mechanisms to optimise the training process:

1 *Mechanism of finding the initial learning rate:* Based on our pre-configured learning setting, we leverage one built-in method in Pytorch called *lr_find*. This method provides a plot of loss and learning rate as shown in Figure 3. We can infer from the figure that the red dot represents our initial suggested learning rate. At this stage, the red dot is not at its lowest position. In this case, the method will find a point decreasing relatively quickly but not too close to the minimum loss value to prevent potential divergence.

2 *Early stopping mechanism:* With that suggested initial learning rate, we set max_epoch to 50. However, to accelerate the training part and prevent overfitting, we use a method called *EarlyStopping* in Pytorch within 50 epochs. It monitors the validation loss during the training process and stops the training when the metric stops improving or decelerating.

5.1.4 Performance metrics

We evaluate the performance of 16 scenarios properly from two perspectives: training and prediction. Three metrics are considered concerning training performance:

1 *Average p50 loss overall:* It represents the median loss during a model's training and offers a robust performance measure less influenced by extreme values. In this case, the average performance for the three targets is measured together.

2 *Average p50 loss per target:* Similarly to the first metric, it focuses on the model's performance for forecasting each target.

Concerning prediction performance, we consider the following metrics:

1 *Mean absolute error (MAE):* It is determined by averaging absolute differences between predicted and actual values, indicating the average magnitude of errors. It can be mathematically expressed as equation (1).

2 *Mean squared error (MSE):* It is determined as averaging squared differences between predicted and actual values, emphasising more significant errors. It can be mathematically expressed as equation (2).

3 *Root mean square error (RMSE):* It is determined by the standard deviation of the residuals and maintains the same unit as the target variable. It can be mathematically expressed as Equation (3).

4 *Attention curve for TFT:* A higher value in the attention curve at a particular time step or input feature indicates that the model assigns more importance to that specific element when generating its output.

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

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

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{3}$$

where y_i is the actual value for the i^{th} data point, $\hat{y}_i$ is the predicted value for the i^{th} data point, and n is the total number of data points in the dataset.

5.2 Results

5.2.1 Short-term prediction results

From Table 4 provided, we can analyse the training performance of the two models, TFT and N-HiTS, based on the look-back windows size (decode length) and prediction window size (prediction length) under consideration.

Table 3 Short-term prediction results for school

Target	*Encode length*	*Prediction length*	*TFT*			*N-HiTS*		
			MAE	*MSE*	*RMSE*	*MAE*	*MSE*	*RMSE*
School	1 day	1 day	5.9811	54.2079	7.3626	7.9916	116.429	10.7902
		3 days	2.924	15.3616	3.9194	11.0376	230.6841	15.1883
		7 days	2.5195	10.3885	3.2231	9.8245	182.4015	13.5056
		28 days	2.9291	15.7269	3.9657	15.3985	361.6085	19.016
	3 days	1 day	6.5094	61.3548	7.8329	9.8244	175.9917	13.2662
		3 days	5.4697	56.946	7.5463	10.7296	201.6049	14.1988
		7 days	2.2257	8.7965	2.9659	9.6917	170.3688	13.0525
		28 days	2.712	13.1448	3.6256	12.5986	251.5438	15.8601
	7 days	1 day	5.2043	43.9184	6.6271	10.8056	209.5272	14.4751
		3 days	2.0612	6.6473	2.5782	12.1327	269.1699	16.4064
		7 days	4.5894	39.1154	6.2542	9.5265	171.9215	13.1119
		28 days	2.8437	12.5256	3.5391	12.68	236.7377	15.3863
	28 days	1 day	8.8667	128.9882	11.3573	11.8812	257.3566	16.0423
		3 days	2.7792	15.0648	3.8813	11.1723	235.5087	15.3463
		7 days	3.9578	28.3005	5.3198	10.5625	185.6453	13.6252
		28 days	3.4032	19.9166	4.4628	14.5143	329.6602	18.1565

Table 4 Training results of short-term prediction

Encode length	*Prediction length*	*TFT*				*N-HiTS*			
		Average p50 loss overall	*Average p50 loss per target*			*Average p50 loss overall*	*Average p50 loss per target*		
			Residential	*School*	*Factory*		*Residential*	*School*	*Factory*
1 day	1 day	11.7155	1.1226	13.0794	20.9446	24.1557	1.5506	25.1503	45.7665
	3 days	14.8674	1.2503	11.3374	32.0146	25.2294	1.4515	17.8319	56.4049
	7 days	16.4171	1.4419	12.6795	35.13	22.5707	1.6094	19.3682	46.7346
	28 days	19.4486	1.2796	19.7753	37.291	29.6139	4.2175	32.1862	52.4381
3 days	1 day	23.0556	1.7768	23.6427	43.7474	17.8783	2.0675	13.0082	38.5592
	3 days	18.9631	1.0258	15.3672	40.4966	16.4600	1.5966	14.1042	33.6794
	7 days	21.1683	1.6787	13.7657	48.0606	21.1955	1.8872	17.3896	44.3098
	28 days	18.2232	1.1274	18.3533	35.1891	20.5284	3.069	24.6892	33.8272
7 days	1 day	25.3694	1.5029	25.8703	48.7351	29.4483	1.8165	25.0357	61.4926
	3 days	11.0821	0.8285	10.4434	21.9746	21.5173	1.3703	14.8279	48.354
	7 days	18.5781	1.7568	12.0121	41.9655	20.7979	1.8076	16.2597	44.3265
	28 days	19.9899	1.2155	20.2044	38.5498	22.00443	3.5044	26.1863	36.3226
28 days	1 day	21.1333	1.5389	22.341	39.5202	23.29732132	2.061	19.7893	48.0417
	3 days	11.2287	1.2817	10.6206	21.7838	18.6470	1.0663	14.5873	40.2874
	7 days	16.7219	1.5628	12.6273	35.9758	21.7468	2.6389	16.9264	45.6751
	28 days	18.0060	1.1883	18.3618	34.4679	28.4607	4.0885	32.1196	49.1742

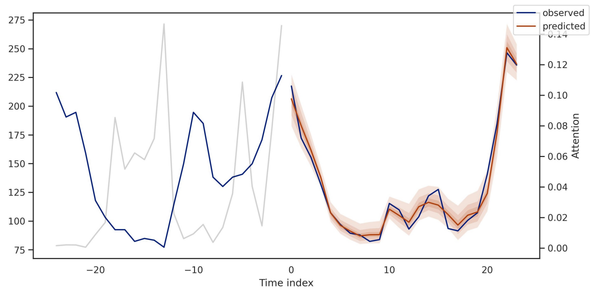
Table 5 Short-term prediction results for residential

Target	*Encode length*	*Prediction length*	*TFT*			*N-HiTS*		
			MAE	*MSE*	*RMSE*	*MAE*	*MSE*	*RMSE*
Residential	1 day	1 day	0.4542	0.3207	0.5663	1.364	2.5894	1.6092
		3 days	0.8284	3.1772	1.7825	2.1735	8.4219	2.9021
		7 days	0.6283	1.36	1.1662	1.7954	6.0589	2.4615
		28 days	0.9933	7.4346	2.7266	3.0181	14.2709	3.7777
	3 days	1 day	0.6172	0.5738	0.7575	1.4483	2.9907	1.7294
		3 days	0.8581	3.1912	1.7864	2.1471	9.3952	3.0652
		7 days	0.5614	1.0401	1.0198	1.9145	6.6381	2.5764
		28 days	0.5684	0.8249	0.9082	2.6773	12.2152	3.495
	7 days	1 day	0.5076	0.5319	0.7293	1.4249	3.3856	1.84
		3 days	0.755	3.0139	1.7361	1.8461	8.3506	2.8897
		7 days	0.7052	1.7644	1.3283	1.9221	6.9912	2.6441
		28 days	1.1148	6.2991	2.5098	2.6553	12.2923	3.506
	28 days	1 day	0.6823	0.7536	0.8681	1.4772	2.9461	1.7164
		3 days	0.7788	2.4812	1.5752	1.8185	7.297	2.7013
		7 days	0.7754	1.7267	1.3141	2.3557	10.1551	3.1867
		28 days	1.0282	7.9478	2.8192	2.7513	12.947	3.5982

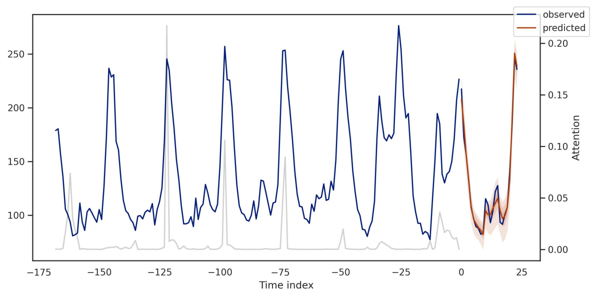
Table 6 Short-term prediction results for factory

Target	*Encode length*	*Prediction length*	*TFT*			*N-HiTS*		
			MAE	*MSE*	*RMSE*	*MAE*	*MSE*	*RMSE*
Factory	1 day	1 day	5.7099	54.2079	7.2077	103.1451	11,026.6768	105.008
		3 days	4.1273	27.7286	5.2658	112.3263	13,433.9775	115.905
		7 days	3.308	18.7053	4.325	124.1845	16,316.0361	127.7342
		28 days	3.1776	17.8393	4.2237	119.57	15,164.6279	123.1447
	3 days	1 day	13.4518	314.1987	17.7257	102.0523	10,821.5127	104.0265
		3 days	4.0661	25.8167	5.081	118.4471	14,863.4307	121.9157
		7 days	2.8019	13.507	3.6752	120.2038	15,341.4912	123.8608
		28 days	2.6193	11.649	3.4131	126.1388	16,920.2695	130.0779
	7 days	1 day	7.6615	95.357	9.7651	106.6045	11,701.9648	108.1756
		3 days	3.2024	16.5914	4.0733	116.7852	14,632.6895	120.9657
		7 days	4.5527	32.2035	5.6748	119.6854	15,334.167	123.8312
		28 days	4.9132	35.5907	5.9658	125.4715	16,673.4551	129.1257
	28 days	1 day	11.1951	204.2473	14.2915	102.5485	10,903.1318	104.4181
		3 days	3.4863	23.3489	4.8321	118.6385	15,020.1875	122.5569
		7 days	6.0435	53.3221	7.3022	118.2622	14,830.7725	121.7817
		28 days	3.8893	25.4946	5.0492	121.5253	15,914.4883	126.1526

Figure 4 TFT short-term forecast result for school, (a) forecast with look-back window size 1 days and prediction window size 1 day (b) forecast with look-back window size 7 days and prediction window size 1 day (see online version for colours)



(a)



(b)

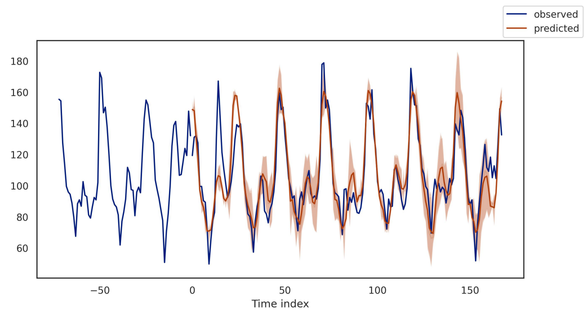
For TFT, when the length of the data used to train the TFT model is increased, the model's overall average p50 loss tends to increase, indicating a decline in performance. The same pattern is observed when the prediction length is increased, with the average p50 loss also tending to rise, indicating that the model struggles to make accurate predictions for events further in the future. For N-HiTS model, we can see that as the length of encoding and prediction increases, the average p50 loss also increases, indicating that larger input and output window sizes make it difficult for the N-HiTS model to maintain prediction accuracy. In comparison, the TFT model generally performs better than N-HiTS under similar conditions, as it tends to have a lower p50 loss. In short, we observe that both models perform better with shorter encoding lengths and prediction lengths, with the TFT model generally outperforming the N-HiTS model during the training stage.

During the prediction stage, we analyse the performance of both models for three different targets.

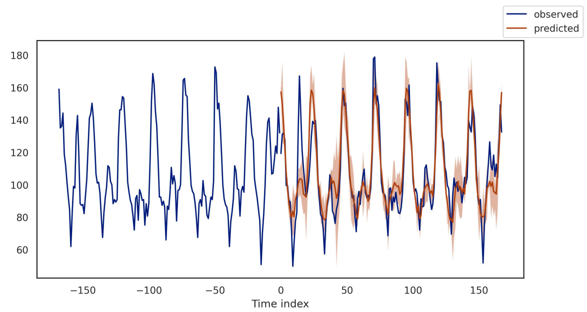
5.2.2 *Short-term prediction results for school*

Table 3 represents the school area forecast results. We can infer that when the look-back window size is held constant, the total error decreases as the prediction window size expands in both models. Such results occur because the model has better flexibility for error handling when generating forecasts over a more extended range. Conversely, when the prediction window size remains fixed, the total error increases as the look-back window size grows for both models. Such results occur because having more data points as background knowledge for the model results in a reduced allowance for error during short-range forecasting. In Figure 4, it is evident that when the prediction window size is kept constant, the forecast accuracy of TFT improves as the look-back range increases. Because the attention curve in Figure 4(b) has described a more approachable and similar trend of observed data points than the curve in Figure 4(a), the model can understand input features better with a more extended range look-back window size and improve prediction. Note that the value of attention in Figure 4(a) is randomly assigned, causing the TFT model to misunderstand the significance of specific time steps. For instance, some low peak demand values could be given a disproportionately high attention value, leading to a misjudgement that affects prediction. This could occur due to a lack of observed knowledge, resulting in less accurate prediction results. As the TFT model receives more inputs, the attention curve becomes more adaptable in discerning each input's significance, meaning that peak and lowest demand values are given greater attention than other demand values, resulting in more accurate prediction results. Also note that in all figures for short-range forecasting, a narrower shaded region will be displayed when the prediction is more accurate. When the prediction error increases, the shaded area becomes more significant and denser. Similar performance is shown for N-HiTS in Figure 5.

Figure 5 N-HiTS short-term forecast result for school, (a) forecast with look-back window size 3 days and prediction window size 7 days (b) forecast with look-back window size 7 days and prediction window size 7 days (see online version for colours)



(a)

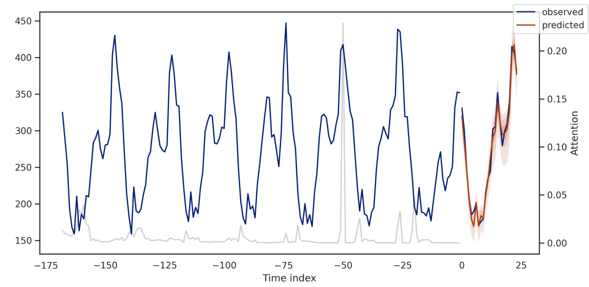


(b)

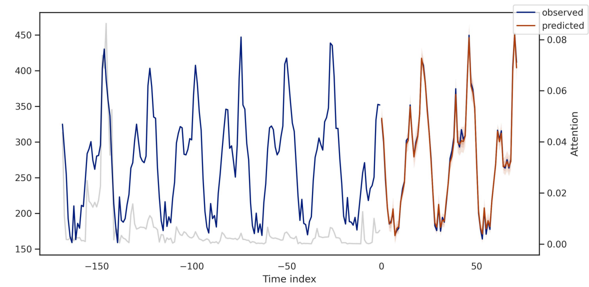
5.2.3 *Short-term prediction results for residential*

In Table 5 for residential forecast, when the look-back window size is constant, the error metrics tend to increase as the prediction window size increases for both models. Such a result suggests that predicting further into the future becomes more challenging and less accurate. When the prediction window size is constant, the error metrics do not follow a consistent pattern as the look-back window size increases for both models. It suggests that the relationship between look-back window size and prediction accuracy is more complex and may not exhibit a clear trend. In Figure 8, when using a fixed look-back window size in the TFT model, the forecasting line becomes less accurate as the prediction window size increases. It is because the Figure 8(a)'s attention curve more effectively highlights the significance of high-peak and low-bottom values in the observed data points, while the Figure 8(b)'s attention always prioritises low-bottom values over high-peaks. However, in the observed data from time index *0*, there are more high-peak values than low-bottom values, which results in the lower configuration delivering poorer performance than the upper one. Meanwhile, as seen in Figure 9, N-HiTS shows a similar tendency.

Figure 6 TFT short-term forecast result for factory (a) forecast with look-back window size 7 days and prediction window size 1 days (b) forecast with look-back window size 7 days and prediction window size 3 days (see online version for colours)



(a)



(b)

5.2.4 *Short-term prediction results for factory*

The factory forecast results displayed in Table 6 show that as the prediction window size increases for both models, the overall error decreases when a fixed look-back window size is used. This is because the model has more room for error when predicting over an extended period. Nonetheless, when the prediction length size is fixed, the overall error increases as the look-back window size increases for both models. This result is expected as more data points are reserved for the model to learn from, making it less likely to make errors when predicting short-range forecasts. As seen in Figure 6, TFT's predictions are more accurate and have a wider range when using fixed look-back windows. The grey line in both graphs indicates that data points with higher attention values significantly impact the forecast. However, as shown in Figure 7, N-HiTS performs worse as the prediction length increases, despite having the same encoding and prediction length set-up. Such results are expected because N-HiTS does not require positional information of the input sequence and therefore struggles to understand the relationship between data points, resulting in an increase in MAE and prediction length. Furthermore, N-HiTS performs worse with a fixed encode length as the prediction range increases.

Table 7 Training result of long-term prediction

Encode length	Prediction length	TFT				N-HiTS			
		Average p50 loss overall	Average p50 loss per target			Average p50 loss overall	Average p50 loss per target		
			Residential	School	Factory		Residential	School	Factory
1 months	1 months	18.0060	1.1883	18.3618	34.4679	28.4607	4.0885	32.1196	49.1742
2 months	2 months	20.0985	1.7388	25.9465	32.6104	34.6232	4.3665	36.0509	63.4524
3 months	3 months	*	*	*	*	30.2940	4.2862	32.6064	53.9895

Table 8 Long-term prediction results for school

Target	Encode length	Prediction length	TFT			N-HiTS		
			MAE	MSE	RMSE	MAE	MSE	RMSE
School	1 months	1 months	3.4032	19.9166	4.4628	14.5143	329.6602	18.1565
	2 months	2 months	2.1371	9.2171	3.036	21.2881	628.5302	25.0705
	3 months	3 months	*	*	*	21.8978	655.9341	25.6112

Table 9 Long-term prediction results for residential

Target	Encode length	Prediction length	TFT			N-HiTS		
			MAE	MSE	RMSE	MAE	MSE	RMSE
Residential	1 months	1 months	1.0282	7.9478	2.8192	2.7513	12.947	3.5982
	2 months	2 months	0.4244	0.3216	0.5671	4.5402	33.4167	5.7807
	3 months	3 months	*	*	*	5.0379	38.243	6.1841

Table 10 Long-term prediction results for factory

Target	Encode length	Prediction length	TFT			N-HiTS		
			MAE	MSE	RMSE	MAE	MSE	RMSE
Factory	1 months	1 months	3.8893	25.4946	5.0492	121.5253	15,914.4883	126.1526
	2 months	2 months	11.348	246.1751	15.69	129.8029	18,497.4883	136.0055
	3 months	3 months	*	*	*	133.9979	19,781.3613	140.6462

In summary, for the short-term task, both TFT and N-HiTS can have relatively good accuracy in the training and prediction stages. Furthermore, both models perform well for the sub-targets test, except N-HiTS performs worse than TFT in factory usage prediction.

5.2.5 Long-term prediction results

During the training stage, as seen in Table 7, we observe that both TFT and N-HiTS models are trained well with relatively small average p50 loss overall and average p50 loss per target for each configuration of encoding length and prediction length. However, for the TFT model, in the three months configuration test, it cannot finish the test (we denote with ‘*’ for all fields) due to being out of VRAM (it ran out of all 24 GB memory in RTX 4090). Furthermore, the N-HiTS can still smoothly finish the test with the same configuration because TFT uses the attention mechanism. It limits performance due to the input sequence length; a longer length will consume more memory. In this three months prediction, its input length is $30 \times 3 \times 24 = 2{,}160$, which is more than TFT can handle.

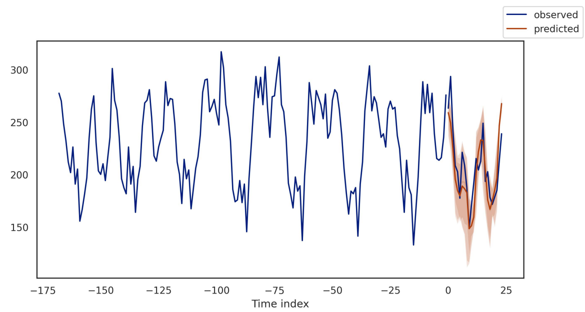
During the prediction stage, the results from Tables 8, 9 and 10 indicate that TFT is better than N-HiTS at forecasting the power demand for the school, residential, and factory. However, TFT cannot complete the forecasting and marks all fields as ‘*’ due to being out of VRAM, for the 3-month test. Despite both models having the lowest MAE, MSE, and RMSE for residential prediction out of the three tasks, the residential prediction result in Figure 10 shows that TFT is significantly better than N-HiTS during the 2-month test. In the sub-figure, the forecasting line of TFT is more precise and accurate, making predictions more reliable. Conversely, the forecasting line of N-HiTS almost becomes a straight line in the upper figure and cannot predict accurately for most data points.

In summary, TFT performs better when predicting accuracy for long-term tasks than N-HiTS. However, as the input length increases, N-HiTS outperforms TFT in handling the job.

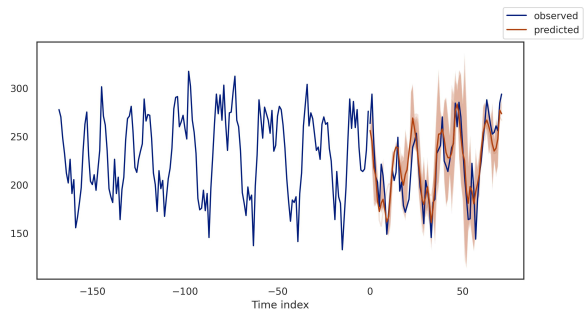
6 Final remarks

In this paper, we have addressed the power demand prediction issue in the SG system by leveraging advanced DL techniques. To comprehensively evaluate different models, we have considered data pre-processing, computational resources, and the complexity of DL models. Our evaluation scenario includes both short-term and long-term forecasting tasks. We have considered the accuracy of short-term predictions and assessed the model's robustness for long-term forecasts. This two-tiered approach ensures that the selected model performs well for immediate and long-term forecasting tasks. Our experiments have shown that the TFT and N-HiTS predictors, trained using a publicly available dataset, demonstrate the practical application of DL models in power demand forecasting. Both TFT and N-HiTS perform well in short-term forecasting tasks. However, TFT outperforms N-HiTS in long-term forecasting tasks, showcasing its accuracy and robustness in predicting future power demand over extended periods. However, TFT requires substantial computational resources, particularly VRAM, for processing longer input sequences. On the other hand, N-HiTS, while providing lower confidence in long-term forecasting than TFT, requires significantly fewer computational resources for longer input data streams, making it a more resource-efficient model for such tasks. In real-world practice, devices in the smart grid could have limited computing resources and require low-latency learning decisions. Thus, our designed learning model can be deployed on an edge server close to devices so that requirements for training computation and timely learning decisions can be satisfied.

Figure 7 N-HiTS short-term forecast result for factory, (a) forecast with look-back window size 7 days and prediction window size 1 day (b) forecast with look-back window size 7 days and prediction window size 3 days (see online version for colours)

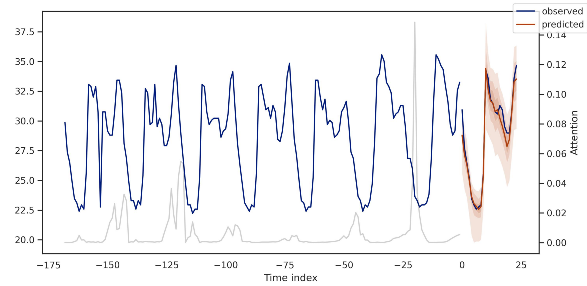


(a)

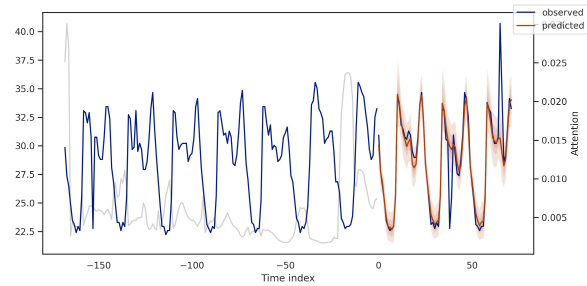


(b)

Figure 8 TFT short-term forecast result for residential, (a) forecast with look-back window size 7 days and prediction window size 1 day (b) forecast with look-back window size 7 days and prediction window size 3 days (see online version for colours)

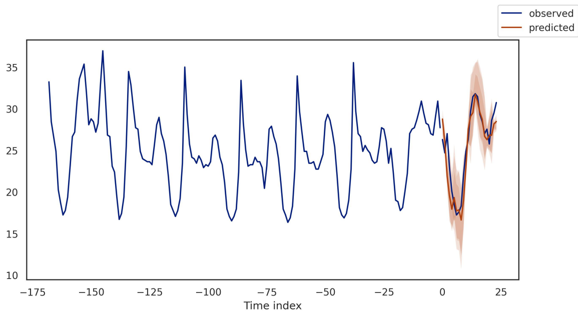


(a)

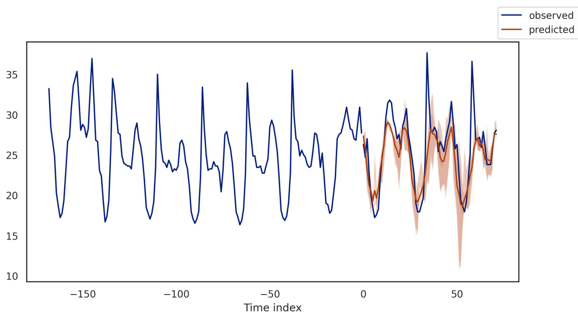


(b)

Figure 9 N-HiTS short-term forecast result for residential, (a) forecast with look-back window size 7 days and prediction window size 1 day (b) forecast with look-back window size 7 days and prediction window size 3 days (see online version for colours)

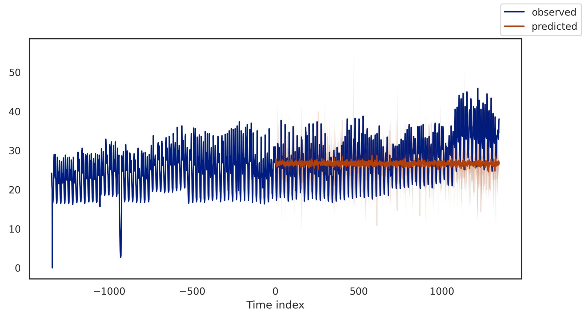


(a)

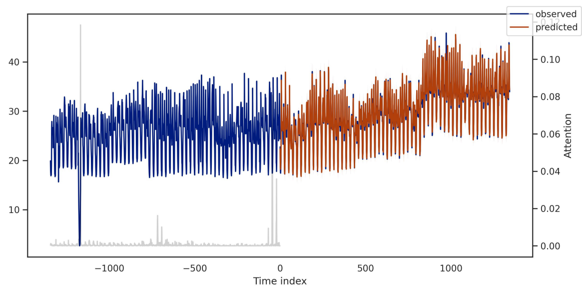


(b)

Figure 10 Long-term forecast results for residential, (a) N-HiTS forecast with look-back window size 2 months and prediction window size 2 months (b) TFT forecast with look-back window size 2 months and prediction window size 2 months (see online version for colours)



(a)



(b)

References

Al-Turjman, F. and Abujubbeh, M. (2019) 'IoT-enabled smart grid via SM: an overview', *Future Generation Computer Systems*, July, Vol. 96, No. C, pp.579–590.

Alazab, M., Khan, S., Krishnan, S.S.R., Pham, Q-V., Reddy, M.P.K. and Gadekallu, T.R. (2020) 'A multidirectional LSTM model for predicting the stability of a smart grid', *IEEE Access*, Vol. 8, pp.85454–85463.

Ba, J.L., Kiros, J.R. and Hinton, G.E. (2016) *Layer Normalization*, arXiv preprint arXiv:1607.06450.

Bahdanau, D., Cho, K. and Bengio, Y. (2014) *Neural Machine Translation by Jointly Learning to Align and Translate*, arXiv preprint arXiv:1409.0473.

Belhaiza, S., Baroudi, U. and Elhallaoui, I. (2020) 'A game theoretic model for the multiperiodic smart grid demand response problem', *IEEE Systems Journal*, Vol. 14, No. 1, pp.1147–1158.

Challu, C., Olivares, K.G., Oreshkin, B.N., Garza, F., Mergenthaler-Canseco, M. and Dubrawski, A. (2022) *N-HITS: Neural Hierarchical Interpolation For Time Series Forecasting*, arXiv preprint arXiv:2201.12886.

Chen, J., Cao, X., Cheng, P., Xiao, Y. and Sun, Y. (2010) 'Distributed collaborative control for industrial automation with wireless sensor and actuator networks', *IEEE Transactions on Industrial Electronics*, Vol. 57, No. 12, pp.4219–4230.

Chen, Z., Liao, W., Hua, K., Lu, C. and Yu, W. (2021) 'Towards asynchronous federated learning for heterogeneous edge-powered internet of things', *Digital Communications and Networks*, Vol. 7, No. 3, pp.317–326.

Colak, I., Bayindir, R. and Sagiroglu, S. (2020) 'The effects of the smart grid system on the national grids', *2020 8th International Conference on Smart Grid (icSmartGrid)*, pp.122–126.

Gao, J., Xiao, Y., Liu, J., Liang, W. and Chen, C.P. (2012) 'A survey of communication/networking in smart grids', *Future Generation Computer Systems*, Vol. 28, No. 2, pp.391–404.

Gao, W., Nguyen, J.H., Yu, W., Lu, C., Ku, D.T. and Hatcher, W.G. (2017) 'Toward emulation-based performance assessment of constrained application protocol in dynamic networks', *IEEE Internet of Things Journal*, Vol. 4, No. 5, pp.1597–1610.

Gardner, M.W. and Dorling, S. (1998) 'Artificial neural networks (the multilayer perceptron) – a review of applications in the atmospheric sciences', *Atmospheric Environment*, Vol. 32, Nos. 14–15, pp.2627–2636.

Ghasempour, A. (2019) 'Internet of things in smart grid: architecture, applications, services, key technologies, and challenges', *Inventions*, Vol. 4, No. 1, p.22.

Gómez, K.S.M. and García, J.R. (2021) 'Asset management model for the transformer fleet of the national laboratory of smart grids (LAB+i) based on fuzzy logic and forecasting', *2021 International Conference on Smart City and Green Energy (ICSCGE)*, pp.57–62.

Habbak, H., Mahmoud, M., Metwally, K., Fouda, M.M. and Ibrahem, M.I. (2023) 'Load forecasting techniques and their applications in smart grids', *Energies*, Vol. 16, No. 3, p.1480.

Hatcher, W.G. and Yu, W. (2018) 'A survey of deep learning: platforms, applications and emerging research trends', *IEEE Access*, Vol. 6, pp.24411–24432.

Hochreiter, S. and Schmidhuber, J. (1997) 'Long short-term memory', *Neural Computation*, Vol. 9, No. 8, pp.1735–1780.

Hossain, E., Khan, I., Un-Noor, F., Sikander, S.S. and Sunny, M.S.H. (2019) 'Application of big data and machine learning in smart grid, and associated security concerns: a review', *IEEE Access*, Vol. 7, pp.13960–13988.

Ioffe, S. and Szegedy, C. (2015) 'Batch normalization: accelerating deep network training by reducing internal covariate shift', *International Conference on Machine Learning*, PMLR, pp.448–456.

Kan, K., Aubet, F-X., Januschowski, T., Park, Y., Benidis, K., Ruthotto, L. and Gasthaus, J. (2022) 'Multivariate quantile function forecaster', *International Conference on Artificial Intelligence and Statistics*, PMLR, pp.10603–10621.

Karimipour, H., Dehghantanha, A., Parizi, R.M., Choo, K-K.R. and Leung, H. (2019) 'A deep and scalable unsupervised machine learning system for cyber-attack detection in large-scale smart grids', *IEEE Access*, Vol. 7, pp.80778–80788.

Kement, C.E., Gultekin, H. and Tavli, B. (2021) 'A holistic analysis of privacy-aware smart grid demand response', *IEEE Transactions on Industrial Electronics*, Vol. 68, No. 8, pp.7631–7641.

Kong, P-Y. (2020) 'Radio resource allocation scheme for reliable demand response management using D2D communications in smart grid', *IEEE Transactions on Smart Grid*, Vol. 11, No. 3, pp.2417–2426.

Li, J., Deng, D., Zhao, J., Cai, D., Hu, W., Zhang, M. and Huang, Q. (2020) 'A novel hybrid short-term load forecasting method of smart grid using MLR and LSTM neural network', *IEEE Transactions on Industrial Informatics*, Vol. 17, No. 4, pp.2443–2452.

Liang, F., Hatcher, W.G., Liao, W., Gao, W. and Yu, W. (2019a) 'Machine learning for security and the internet of things: the good, the bad, and the ugly', *IEEE Access*, Vol. 7, pp.158126–158147.

Liang, F., Qian, C., Hatcher, W.G. and Yu, W. (2019b) 'Search engine for the internet of things: lessons from web search, vision, and opportunities', *IEEE Access*, Vol. 7, pp.104673–104691.

Liang, F., Yu, W., Liu, X., Griffith, D. and Golmie, N. (2020) 'Toward edge-based deep learning in industrial internet of things', *IEEE Internet of Things Journal*, Vol. 7, No. 5, pp.4329–4341.

Liu, J., Xiao, Y., Li, S., Liang, W. and Chen, C.L.P. (2012) 'Cyber security and privacy issues in smart grids', *IEEE Communications Surveys & Tutorials*, Vol. 14, No. 4, pp.981–997.

Liu, X., Qian, C., Hatcher, W.G., Xu, H., Liao, W. and Yu, W. (2019) 'Secure internet of things (IoT)-based smart-world critical infrastructures: survey, case study and research opportunities', *IEEE Access*, Vol. 7, pp.79523–79544.

Loshchilov, I. and Hutter, F. (2017) *Decoupled Weight Decay Regularization*, arXiv preprint arXiv:1711.05101.

Mahmood, K., Chaudhry, S.A., Naqvi, H., Kumari, S., Li, X. and Sangaiah, A.K. (2018) 'An elliptic curve cryptography based lightweight authentication scheme for smart grid communication', *Future Generation Computer Systems*, April, Vol. 81, No. C, pp.557–565.

Mohammadi, M., Al-Fuqaha, A., Sorour, S. and Guizani, M. (2018) 'Deep learning for IoT big data and streaming analytics: a survey', *IEEE Communications Surveys & Tutorials*, Vol. 20, No. 4, pp.2923–2960.

Moulema, P., Yu, W., Griffith, D. and Golmie, N. (2015) 'On effectiveness of smart grid applications using co-simulation', *2015 24th International Conference on Computer Communication and Networks (ICCCN)*, pp.1–8.

Nazir, A., Shaikh, A.K., Shah, A.S. and Khalil, A. (2023) 'Forecasting energy consumption demand of customers in smart grid using temporal fusion transformer (TFT)', *Results in Engineering*, Vol. 17, p.100888.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L. et al. (2019) 'PyTorch: an imperative style, high-performance deep learning library', *Advances in Neural Information Processing Systems*, Vol. 32, pp.8026–8037.

Puhe, F. and Rehtanz, C. (2022) 'An adaptive subgrid identification strategy for future medium voltage smart grids based on machine learning', *2022 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pp.1–5.

Quan, H., Khosravi, A., Yang, D. and Srinivasan, D. (2019) 'A survey of computational intelligence techniques for wind power uncertainty quantification in smart grids', *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 31, No. 11, pp.4582–4599.

Shafique, K., Khawaja, B.A., Sabir, F., Qazi, S. and Mustaqim, M. (2020) 'Internet of things (IoT) for next-generation smart systems: a review of current challenges, future trends and prospects for emerging 5G-IoT scenarios', *IEEE Access*, Vol. 8, pp.23022–23040.

Sun, W., Li, P., Liu, Z., Xue, X., Li, Q., Zhang, H. and Wang, J. (2021) 'LSTM based link quality confidence interval boundary prediction for wireless communication in smart grid', *Computing*, February, Vol. 103, No. 2, pp.251–269.

Sun, X., Tang, Z., Du, M., Deng, C., Lin, W., Chen, J., Qi, Q. and Zheng, H. (2022) 'A hierarchical federated learning-based intrusion detection system for 5G smart grids', *Electronics*, Vol. 11, No. 16, p.2627.

Taud, H. and Mas, J. (2018) 'Multilayer perceptron (MLP)', *Geomatic Approaches for Modeling Land Change Scenarios*, pp.451–455.

UCI Machine Learning Repository (2015) *Electricity Load Diagrams 2011–2014* [online] https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014 (accessed March 2023).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I. (2017) 'Attention is all you need', *Advances in Neural Information Processing Systems*, Vol. 30, pp.6000–6010.

Wan, C., Zhao, J., Song, Y., Xu, Z., Lin, J. and Hu, Z. (2015) 'Photovoltaic and solar power forecasting for smart grid energy management', *CSEE Journal of Power and Energy Systems*, Vol. 1, No. 4, pp.38–46.

Wang, J., Varshney, N., Gentile, C., Blandino, S., Chuang, J. and Golmie, N. (2022) 'Integrated sensing and communication: enabling techniques, applications, tools and data sets, standardization, and future directions', *IEEE Internet of Things Journal*, Vol. 9, No. 23, pp.23416–23440.

Wang, J., Varshney, N., Zhang, J., Griffith, D. and Golmie, N. (2021) 'Deep learning based link-level abstraction for mmwave communications', *2021 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI)*, pp.391–398.

Xu, G., Yu, W., Griffith, D., Golmie, N. and Moulema, P. (2017) 'Toward integrating distributed energy resources and storage devices in smart grid', *IEEE Internet of Things Journal*, Vol. 4, No. 1, pp.192–204.

Xu, H., Liu, X., Yu, W., Griffith, D. and Golmie, N. (2020) 'Reinforcement learning-based control and networking co-design for industrial internet of things', *IEEE Journal on Selected Areas in Communications*, Vol. 38, No. 5, pp.885–898.

Xu, H., Yu, W., Griffith, D. and Golmie, N. (2018) 'A survey on industrial internet of things: a cyber-physical systems perspective', *IEEE Access*, Vol. 6, pp.78238–78259.

Yu, W., Griffith, D., Ge, L., Bhattarai, S. and Golmie, N. (2015) 'An integrated detection system against false data injection attacks in the smart grid', *Security and Communication Networks*, Vol. 8, No. 2, pp.91–109.

Zheng, J., Xu, C., Zhang, Z. and Li, X. (2017) 'Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network', *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, IEEE, pp.1–6.

Notes

1 Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.