Multi-Perspective Evaluations for Power Demand Prediction Models: Balancing Prediction Accuracy, Interpretability, and Computational Efficiency

Bowen Yang[1], Mustafa Gül[1], Yuxiang Chen[1]
[1]Department of Civil and Environmental Engineering, University of Alberta, Edmonton, Canada

Abstract

Reliable and robust power demand prediction is crucial for the effective management of building automation systems. Evaluating prediction models based only on accuracy is insufficient since sophisticated models usually suffer from low interpretability and high computational demand. This study proposes a multi-perspectives evaluation framework to compare the performance of two deep learning (DL) models and two tree-based models in terms of prediction accuracy, model interpretability, and computational efficiency. The findings reveal that although DL models demonstrate superior prediction accuracy, tree-based models outperform in interpretability, offering more transparent decision-making processes and requiring less training time than the DL models.

Highlights

- Building power demand prediction
- Prediction accuracy
- Model interpretation
- Computational efficiency

Introduction

The building sectors represent for approximately 40% of global energy consumption and 30% of greenhouse gas emissions (Fang et al. 2021). According to the Word Energy Outlook 2023 report (Internationale Energieagentur 2013), buildings and industrial sectors are responsible for more than 90% of global electricity usage. Building energy management (BEM) is significant in enhancing energy efficiency, aiming to monitor and control energy consumption effectively. Building demand forecasting, especially short-term load forecasting ranging from a few hours to several days, is crucial for efficiently managing and optimizing building energy systems. Accurate predictions of energy demand facilitate proactive decision-making, thereby ensuring occupant thermal comfort and improving energy efficiency.

In the context of building energy simulation and prediction, three approaches have been used: “white-box” (i.e., physical model), “grey-box” (hybrid model), and “black-box” (machine learning (ML) model). The “white-box” model relies on detailed physical process representations, such as heat transfer equations, to simulate building thermal behaviors. It requires comprehensive building information. Various software are available for building simulation, including EnergyPlus (Crawley et al. 2001), TRNSYS (Kalogirou 2001), and Modelica (Tiller 2001). However, this approach needs detailed building information and domain expertise to be accurate for simulation or calibration. The “grey-box” approach combines physical knowledge with data-driven techniques, which uses a simplified physical model to capture complex interactions within building systems. Nevertheless, the RC model is inappropriate for total power demand predictions as it does not inherently consider the energy consumption of appliances, lighting, heating ventilation, and air conditioning (HVAC) electrical loads. Conversely, the “black-box” approach leverages data and ML algorithms to reveal the patterns and relationships between inputs (i.e., features) and outputs (i.e., targets) without considering the physical processes. Many traditional ML models, including linear regression (Ciulla and D’Amico 2019), decision tree (DT) (Yu et al. 2010), support vector machine (Dong, Cao, and Lee 2005), k-nearest neighbor (Khan, Siddique, and Sakib 2019), and artificial neural network (ANN) (Afram et al. 2017), have been employed for load forecasting.

Although ML models have demonstrated acceptable prediction accuracy, there remains a great potential for improving the accuracy. Ensemble tree-based and deep learning (DL) models have recently drawn a lot of attentions. Wang et al. (Wang et al. 2018) developed a random forest (RF) model to predict hourly energy consumption in two educational buildings. Similarly, Feng et al. (Feng et al. 2021) proposed an extreme gradient boosting (XGBoost) to predict the space cooling demand for residential buildings. For DL models, In Ref. (Li et al. 2021), authors developed an attention mechanism-based recurrent neural network (RNN) to predict cooling load. Sekhar et al. (Sekhar and Dahiya 2023) built a hybrid DL model that integrates a convolution neural network and a long short-term memory (LSTM) model for load

forecasting. The complex structure of advanced deep learning models challenges their interpretability and understanding of the decision-making process, raising considerable discussion regarding the necessity of DL for prediction tasks. Ahmad et al. (Ahmad, Mourshed, and Rezgui 2017) compared the performance of RF and ANN models for HVAC energy consumption prediction. The author concluded that ANN slightly outperformed the RF. Yet, this comparison study only focused on prediction accuracy, neglecting other critical performance aspects such as model interpretability and computational efficiency. Such an oversight could hinder the practical application of these models. Given the lack of a comprehensive framework for assessing model performance. This study aims to address the gap by systematically comparing the DL and tree-based models in terms of prediction accuracy, model interpretation, and computational efficiency, providing the capabilities and limitations of these models.

Methodology

Research Outline

The research outline of this study is depicted in Figure 1 and contains three sections:

1) Step 1: Data collection and pre-processing

Data were collected from a residential unit in Edmonton, Canada, at a resolution of 1-hour. The dataset was divided into training and testing subsets for model training and evaluation purposes. To ensure the data quality, several data-preprocessing technologies were applied, including the interpolation for missing data fill-in, min-max normalization to scale features to a specific range, logarithmic transformation for numerical stability, and cyclical transformation to encode cyclical information. Notably, unlike DL models that inherently manage time dependencies, tree-based models require the creation of time lag features for time-series prediction.

Step 2: Model development

Two DL and two tree-based models were developed to predict the power demand. These algorithms are further elaborated in the next section.

2) Step 3: Model performance testing

Model performances were systematically evaluated based on three aspects. Prediction accuracy refers to the capability of a model to make correct predictions on the unseen dataset, which measures how well the model performs on the testing dataset. Model interpretation refers to how much humans can understand the model's prediction and the rationale behind its decisions. Finally, computational cost refers to the resources required for model training, including time and hardware capabilities.

Model description

1) RNN

RNN is a specific type of neural network that has been specifically developed to handle sequential data based on the model structure, allowing RNN to capture the information from the previous time steps. The RNN cell is presented by Eq. (1).

$$h_t = \tanh\left(W_x \cdot x_t + W_h \cdot h_{t-1} + b_h\right) \quad (1)$$

where h_t is the hidden state at time step t, which carries the information from the current input x_t and previous hidden state h_{t-1}. W_x and W_h are the weight matrix in mapping current input and hidden state. b_h is the bias vector for h_t. Then, the prediction can be calculated using the h_t, as shown in Eq. (2).

$$y_t = \sigma(W_y \cdot h_t + b_y) \quad (2)$$

where y_t denotes the output at time step t, which can be derived from the σ function. W_y is the wight matrix and b_y is the bias vector.

2) LSTM

Different from the RNN, LSTM has more complicated memory mechanisms that are designed to mitigate the risk of vanishing and exploding and the special mechanism also makes it effective in capturing long-term dependencies. The important components are the three gates that control the information flow through the LSTM cells.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (3)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (4)$$

$$\widetilde{C_t} = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (5)$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \quad (6)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (7)$$

$$h_t = o_t * \tanh\left(C_t\right) \quad (8)$$

As shown in Eq. (3), which denotes the forget gate, f_t at time t, calculated by sigmoid function σ. Forget gate is used to determine what information needs to be remained or forgotten from the previous cell state C_t. In Eq. (4), the input gate i_t controls what information can be added to the current cell state C_t, and potential cell state $\widetilde{C_t}$ represents the potential information that could be added to the cell state. Output gate o_t controls what information from the cell state and input should be implemented to make prediction for current time step. And the h_t denotes the hidden state, determining what information from cell state can be the output.

3) RF

RF is a parallel ensemble learning method that includes a larger amount of DT to improve prediction accuracy. RF

uses the bagging technique (i.e., randomly sampling with replacement) to create multiple subsets of the training dataset to train individual trees. Instead of employing a random sample technique with replacement from the whole dataset, the data is initially partitioned into sequential windows that do not overlap. To generate a bootstrap sample for each DT, a random selection of windows is made with replacement. This approach not only maintains the temporal sequence within each block but also involves diversity for all samples. The prediction is generated by averaging the predictions from all DTs.

4) XGBoost

XGBoost is a computationally efficient and scalable algorithm based on the gradient boosting decision tree technique. It is a sequential ensemble learning method that sequentially builds DT. In contrast to the RF, XGBoost uses the boosting technique to ensure each DT is small to prevent overfitting, and each DT aims to correct the residuals made by previous trees so the final DT could lower the bias and variance.

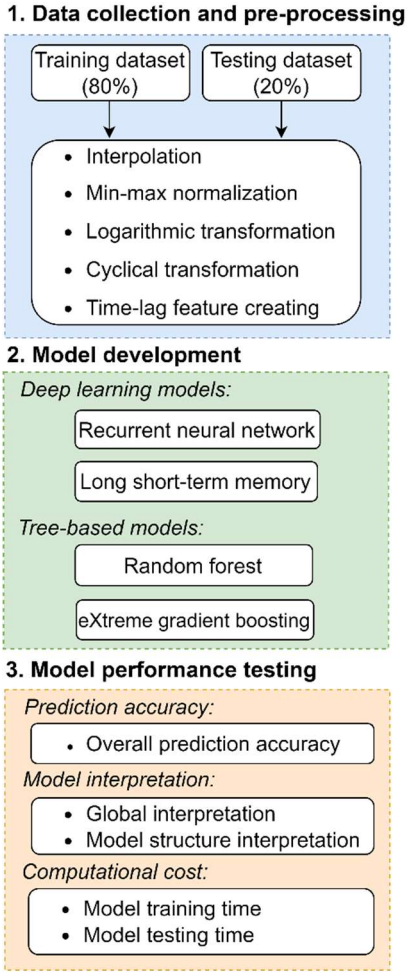

Figure 1: Outline of the research.

Prediction evaluation metric

The model's prediction accuracy is evaluated using two common metrics in this study: coefficient of the variation of root mean squared error (CV-RMSE) and mean absolute percentage error (MAPE). Both are scale-independent, meaning they are unaffected by the features' scales and can provide a holistic understanding of model prediction capability.

Two evaluation metrics are calculated by Eqs. (9) and (10).

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \quad (9)$$

$$CV - RMSE = \frac{\sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}}{\frac{\sum_{i=1}^{n} y_i}{n}} \quad (10)$$

where $\hat{y}_i$ is the predicted values of building power consumption, y_i denotes the actual power consumption, and n represents the number of the observations.

Model interpretation approach

Despite the high prediction accuracy provided by "black-box" models, the inherent model architecture and mechanism make the model opaque, making it challenging for decision-makers to understand how predictions are made. To address this, this study employs shapley additive explanations (SHAP) to explain the models. SHAP was introduced by Lundberg and Lee (Lundberg and Lee, n.d.) that the idea was inspired by game-theoretic concept, where the features are regarded as the "players" in a game and their contributions to a prediction are quantified, thereby providing the insights for model interpretation.

SHAP quantifies a model's prediction by summing the contributions of its features. Each feature's contribution represents how much it adds or subtracts from the predicted value. For example, to calculate the SHAP value for a feature A, at a particular observation, the step of calculation is below:

Step 1: Consideration of all possible feature subsets: For example, for a model with three features, the total possible subsets should be $2^3 = 8$.

Step 2: Model prediction for each subset: Perform predictions both with and without feature A to assess its impact.

Step 3: Contribution calculation: The contribution of feature A is determined by the difference in predictions made with and without feature A.

Step 4: Weight calculation: Assign weights for each subset.

Step 5: Average the contribution: The SHAP value for feature A is obtained by averaging its contributions across

all subsets, providing a measurable assessment of its influence on model predictions.

Computation efficiency

The models in this study were implemented using Python, and the models were tested on a computer equipped with an AMD Ryzen 5 PRO 1500 Quad-Core Processor (3.50 GHz) and 32GB of RAM. It is important to note that all models were subjected to the same operating environment to minimize the influence of external variables.

Result and discussion

Data collection

The effectiveness of the proposed models was demonstrated using data collected from a residential unit in Edmonton, Canada. In this dataset, the hourly electrical power consumption, gas consumption and voltage data were measured from January 1, 2021, to December 31, 2021. These measurements were used as inputs for model prediction. Furthermore, meteorological data and time-related features were also utilized as the power demand also depends on the weather conditions (e.g., outdoor temperature, humidity, and etc) and time of the day (e.g., normally, people use more energy at around 12 pm and 6 pm). Details of the input variables are shown in Table 1.

Table 1: Input variables.

Variable category	Input	Type	Unit
Weather condition	Outdoor temperature	Numeric	℃
	Dewpoint temperature	Numeric	℃
	Relative humidity	Numeric	%
	Wind speed	Numeric	km/h
	Atmospheric pressure	Numeric	kPa
Timestamp based features	Month	Categorical	January ~ December
	Day	Categorical	Monday ~ Sunday
	Hour	Categorical	0:00-23:00
Power consumption	Total electric power consumption/electrical power consumption of heat pump, air handling unit, and heat recovery ventilator	Numeric	Watts
Other	Voltage	Numeric	Volt
	Gas consumption	Numeric	Cubit foot

Prediction accuracy between DL and tree-based models

Table 2 shows the accuracy comparison of the two DL models and two tree-based models on validation and testing datasets. In comparison to the tree-based models, the DL models demonstrate reduced MAPE values of 12.71% and 12.00%, respectively, on the validation set. Similarly, the DL models exhibit reduced CV-RMSE values of 23.20% for RNN and 22.97% for LSTM, whereas the tree-based models have higher CV-RMSE values of 32.76% for RF and 26.4% for XGBoost.

On the testing set, the DL models continue to outperform the tree-based models in terms of MAPE, with RNN at 10.14% and LSTM at 9.53%, compared to RF at 13.5% and XGBoost at 12.91%. Similarly, for the CV-RMSE metric, the DL models exhibit lower values, with RNN at 23.74% and LSTM at 21.54%, while RF and XGBoost have higher CV-RMSE values of 26.87% and 26.62%, respectively.

Figure 2 provides a temporal comparison of model predictions, with each subplot illustrating the actual values (represented by the grey line) versus the predicted values (indicated by the red line). Obviously, all models could capture the general power consumption patterns (i.e., trending). Specifically, even though DL models outperform tree-based models slightly at the peak demand, all models do not perform well at the peak demand, which might be due to the complex occupant behaviors of the tenants. For instance, the notable spike in demand on October 23 resulted from a non-recurring event among the tenants, such as a party or gathering, suggesting anomalies in occupant behavior that are challenging for models to predict.

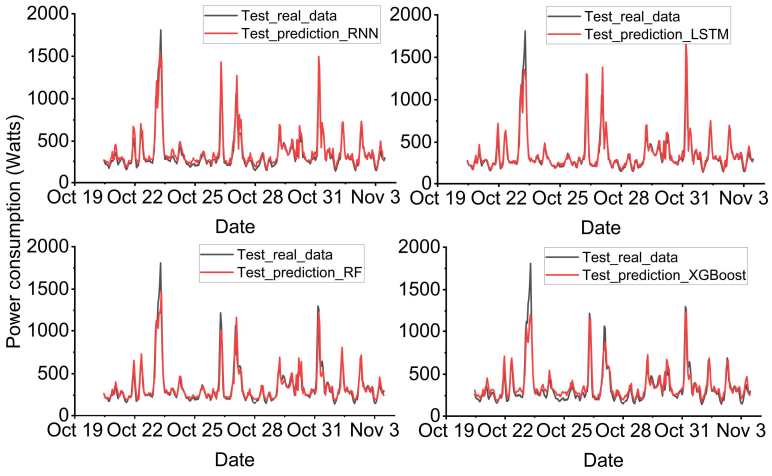

Figure 2: Prediction accuracy comparison between DL and tree-based models.

Table 2: The performance of DL and tree-based models.

Metrics	Validation			
	DL-based Models		Tree-based models	
	RNN	LSTM	RF	XGBoost
MAPE	12.71%	12.00%	17.62%	14.70%
CV-RMSE	23.20%	22.97%	32.76%	26.4%
Metrics	**Testing**			
	RNN	LSTM	RF	XGBoost
MAPE	10.14%	9.53%	13.5%	12.91%
CV-RMSE	23.74%	21.54%	26.87%	26.62%

Model global interpretation

The evaluation of model global interpretation is illustrated in Figure 3, which presents the feature contribution using SHAP values for both DL and tree-based models. Since there is no need to plot the contributions for all features, only the top five features were visualized in the SHAP summary plot.

Within the SHAP summary plot, each dot in SHAP summary plot indicates a SHAP value of a single sample from the dataset. The color ranges from blue to red, indicating the features' values, more significant the value, the brighter the color. A consistent observation across all models is the importance of the previous total power consumption feature, which is reasonable as the power demand in the next time step is high depending on the previous power consumption. One interesting finding is that while the SHAP summary plot shows that the DL models primarily focus on the previous power consumption data, the tree-based models provide which lagged previous power consumption is more essential. For example, tree-based models identify the "*Total power consumption lag 1*" as more critical than "*Total power consumption lag 2*" and "*Total power consumption lag 3*", indicating the most recent power consumption data are more informative compared to the furthest total power consumption. This difference is attributed to the inherent capability of the DL models to capture temporal dependencies, while tree-based models require detailed information of lagged features to accommodate time series dynamics.

Additionally, the cyclical features like "*Hour cos*", "*Hour sin*", and "*Month sin*" are essential features for DL models, showing that the models efficiently capture and leverage the data's periodic patterns, which could be useful for time series forecasting applications, such as the load forecasting, where recognizing and incorporating cyclical trends is imperative for accurate predictions.

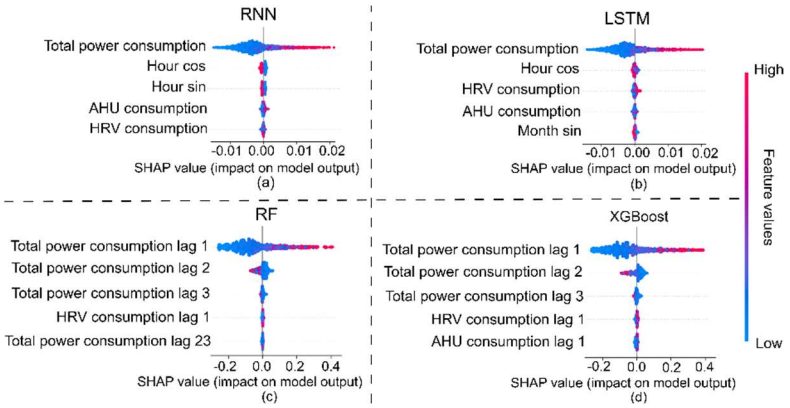

Figure 3: SHAP summary plot for DL and tree-based models.

Model structure interpretation

Figure 4 shows the two layers of hidden state visualization of RNN and LSTM models. On the heatmaps, the X-axis represents hidden state units, while the Y-axis represents the previous time steps. The color intensity in each cell indicates the activation value of a certain hidden state unit at a given time step. Columns of consistent brightness suggest focused attention by the model on certain features over time, while uniformly dark columns indicate inattention. The heatmap in the first layer of the RNN model shows a various activation pattern across hidden state units and time steps, reflecting that the various hidden state units are activated at different time steps and demonstrating the RNN's capability to capture temporal information from the input data. Similar to the LSTM model, but the heatmap for the first layer of the LSTM model shows a more structured pattern than the RNN, with different high and low activation regions, and some hidden state units are always active or inactive throughout the time step. This behavior is characteristic of the LSTM's ability to retain or forget information through its gating mechanism selectively.

The second layer heatmaps exhibit more uniformity because the first layer attempts to learn patterns from the input data directly, whereas the second layer compresses the information into a singular vector and only captures the most pertinent information for prediction. Overall, compared to the hidden state of the RNN, the LSTM model demonstrates a more systematic pattern of hidden state activations, especially in the second layer, indicating the gate mechanism of the LSTM is successfully acquiring the ability to effectively store and process important temporal information for power demand forecasting.

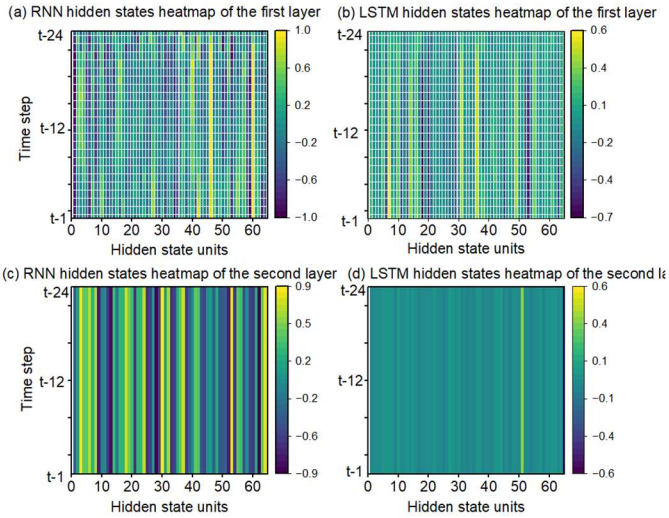

Figure 4: Hidden state visualization for deep learning models.

Despite observing notable feature significance in the heatmaps, the specific features that correspond to bright activations remain unidentified. In contrast, tree-based models provide a clear decision-making process that is more understandable by following the path from the root to a leaf node. Since an ensemble tree model contains a large group of DTs, it is infeasible to show all the DTs. For demonstration purposes, take RF as an example. Figure 5 illustrates an RF structure at the top of the tree showing the root node that includes the entire dataset with 5582 samples, and it is split based on the feature "*Total power consumption lag1*". This split directs samples either left or right, depending on whether they fall below or above the threshold of 0.441. Each internal node within the tree symbolizes a decision criterion based on a feature and its threshold value, also displaying the mean squared error and the number of samples at that node. In the tree structure, every internal node corresponds to a decision criterion that is determined by a particular feature and its associated threshold value. Additionally, the node shows the mean squared error and the number of samples associated with that node.

Clearly, the "*Total power consumption lag1*" feature appears to be the most essential feature for this particular DT, aligning with the SHAP analysis finding. It has also been identified that lagged power consumption features significantly impact the RF model. The tree-based model structure visualization provides insight into the inner working process, thus improving interpretability for decision-makers to understand the behavior of the models and how the model produces predictions.

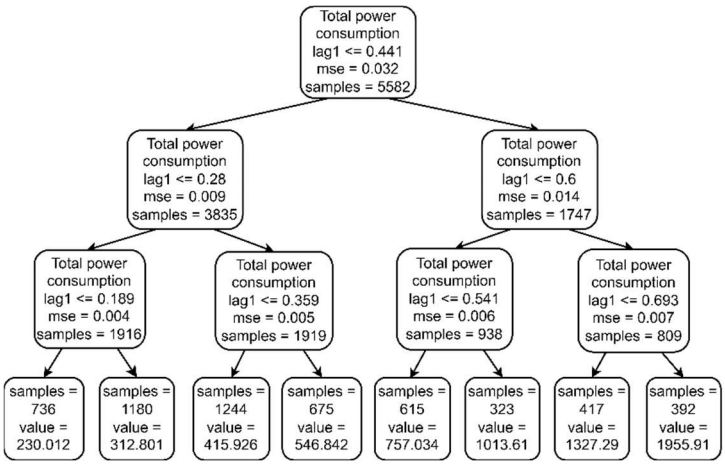

Figure 5: Tree structure of RF.

Computational efficiency

Table 3 displays the comparison of the computational time required for training and testing of the different models. The computational time is presented relative to the most efficient tree-based model, XGBoost, which is the baseline for training time (set to 1), and RF, which serves as the baseline for testing time (set to 1). During the training phase, the RNN model requires a computational time that is notably longer, 110.86 times that of the XGBoost baseline. The LSTM model demands even more time, 243.2 times that of XGBoost, reflecting the intensive computational resources necessitated by the complexity inherent to these recurrent architectures. Those who employ recurrent architectures, such as LSTMs and RNNs, need significant computational resources throughout the training process because of the sophisticated parameter updates and sequential data processing. Conversely, the RF model is more time-efficient in training, requiring 49.0 times less time than the XGBoost baseline. This efficiency can be attributed to the ability of tree-based models to construct individual DT independently and in parallel, optimizing the use of available computational resources. In the testing phase, the computational times for the RNN and LSTM models are 7.1 and 7.0 times longer, respectively, than that of the RF baseline. This is expected, given that DL models inherently need more computational resources for inference operations owing to their intricate architectures and sequential data processing.

Table 3: Computational time of the models.

Model	Algorithm	Training	Testing
DL	RNN	149.99 (110.8)	0.537 (7.0)
	LSTM	329.003 (243.2)	0.550 (7.1)
Tree-based	RF	66.353 (49.0)	0.077 (1)
	XGBoost	1.353 (1)	0.09 (1.2)

Conclusion

Achieving a balance between prediction accuracy, model interpretation, and computational efficiency is significant in predicting building power demand. This study evaluates short-term power demand prediction for a residential unit, employing two DL and two tree-based models, assessing their performance in terms of how accurate, understandable, and fast they are. The key findings are shown below:

1) DL models generally surpass tree-based models in terms of performance on training and testing datasets. Although all models capture the overall power consumption trend, they do not perform well at the peak demand, which might be attributed to the unpredictable and complex occupant behaviors of the tenants.
2) Regarding model interpretability, DL and tree-based models identify previous power consumption as a pivotal predictor. Yet, tree-based models could offer clearer insights into the influence of specific lagged power consumptions. Furthermore, tree-based models include an inherent transparency that provides advantages to decision-makers, enabling them to make well-informed decisions to enhance building energy efficiency.
3) The more sophisticated structures of DL models require more time for training and testing —110 and 243 times longer and 7.0 and 7.1 longer than the baseline for model training and testing, respectively, compared to tree-based models' baselines.

While DL and tree-based models demonstrate the capture of the power demand trend, notable gaps between predictions and actual data remain at peak demands. Instead of the power demand point forecasting, future work will explore the capabilities of the probabilistic models, aiming to quantify uncertainty and potentially enhance peak demand forecasting.

Acknowledgement

The authors would acknowledge our sincere appreciation to the financial supports from the NSERC, Landmark Group of Companies Inc.

References

Afram, Abdul, Farrokh Janabi-Sharifi, Alan S. Fung, and Kaamran Raahemifar. 2017. "Artificial Neural Network (ANN) Based Model Predictive Control (MPC) and Optimization of HVAC Systems: A State of the Art Review and Case Study of a Residential HVAC System." *Energy and Buildings* 141 (April): 96–113. https://doi.org/10.1016/j.enbuild.2017.02.012.

Ahmad, Muhammad Waseem, Monjur Mourshed, and Yacine Rezgui. 2017. "Trees vs Neurons: Comparison between Random Forest and ANN for High-Resolution Prediction of Building Energy Consumption." *Energy and Buildings* 147 (July): 77–89. https://doi.org/10.1016/j.enbuild.2017.04.038.

Ciulla, G., and A. D'Amico. 2019. "Building Energy Performance Forecasting: A Multiple Linear Regression Approach." *Applied Energy* 253 (November): 113500. https://doi.org/10.1016/j.apenergy.2019.113500.

Crawley, Drury B., Linda K. Lawrie, Frederick C. Winkelmann, W. F. Buhl, Y. Joe Huang, Curtis O. Pedersen, Richard K. Strand, et al. 2001. "EnergyPlus: Creating a New-Generation Building Energy Simulation Program." *Energy and Buildings*, Special Issue: BUILDING SIMULATION'99, 33 (4): 319–31. https://doi.org/10.1016/S0378-7788(00)00114-6.

Dong, Bing, Cheng Cao, and Siew Eang Lee. 2005. "Applying Support Vector Machines to Predict Building Energy Consumption in Tropical Region." *Energy and Buildings* 37 (5): 545–53. https://doi.org/10.1016/j.enbuild.2004.09.009.

Fang, Xi, Guangcai Gong, Guannan Li, Liang Chun, Wenqiang Li, and Pei Peng. 2021. "A Hybrid Deep Transfer Learning Strategy for Short Term Cross-Building Energy Prediction." *Energy* 215 (January): 119208. https://doi.org/10.1016/j.energy.2020.119208.

Feng, Yanxiao, Qiuhua Duan, Xi Chen, Sai Santosh Yakkali, and Julian Wang. 2021. "Space Cooling Energy Usage Prediction Based on Utility Data for Residential Buildings Using Machine Learning Methods." *Applied Energy* 291 (June): 116814. https://doi.org/10.1016/j.apenergy.2021.116814.

Hannan, Mahammad A., Mohammad Faisal, Pin Jern Ker, Looe Hui Mun, Khadija Parvin, Teuku Meurah Indra Mahlia, and Frede Blaabjerg. 2018. "A Review of Internet of Energy Based Building Energy Management Systems: Issues and Recommendations." *IEEE Access* 6: 38997–14. https://doi.org/10.1109/ACCESS.2018.2852811.

Internationale Energieagentur, ed. 2013. *World Energy Outlook 2013*. World Energy Outlook 2013. Paris.

Kalogirou, Soteris A. 2001. "Use of TRNSYS for Modelling and Simulation of a Hybrid Pv–Thermal Solar System for Cyprus." *Renewable Energy* 23 (2): 247–60. https://doi.org/10.1016/S0960-1481(00)00176-2.

Khan, Mohammad Mahmudur Rahman, Md. Abu Bakr Siddique, and Shadman Sakib. 2019. "Non-Intrusive Electrical Appliances Monitoring and Classification Using K-Nearest Neighbors." In *2019 2nd International Conference on Innovation in Engineering and Technology (ICIET)*, 1–5. Dhaka, Bangladesh: IEEE. https://doi.org/10.1109/ICIET48527.2019.9290671.

Li, Ao, Fu Xiao, Chong Zhang, and Cheng Fan. 2021. "Attention-Based Interpretable Neural Network for Building Cooling Load Prediction." *Applied Energy* 299 (October): 117238. https://doi.org/10.1016/j.apenergy.2021.117238.

Lundberg, Scott M, and Su-In Lee. n.d. "A Unified Approach to Interpreting Model Predictions."

Sekhar, Charan, and Ratna Dahiya. 2023. "Robust Framework Based on Hybrid Deep Learning Approach for Short Term Load Forecasting of Building Electricity Demand." *Energy* 268 (April): 126660. https://doi.org/10.1016/j.energy.2023.126660.

Tiller, Michael. 2001. *Introduction to Physical Modeling with Modelica*. Springer Science & Business Media.

Wang, Zeyu, Yueren Wang, Ruochen Zeng, Ravi S. Srinivasan, and Sherry Ahrentzen. 2018. "Random Forest Based Hourly Building Energy Prediction." *Energy and Buildings* 171 (July): 11–25. https://doi.org/10.1016/j.enbuild.2018.04.008.

Yu, Zhun, Fariborz Haghighat, Benjamin C. M. Fung, and Hiroshi Yoshino. 2010. "A Decision Tree Method for Building Energy Demand Modeling." *Energy and Buildings* 42 (10): 1637–46. https://doi.org/10.1016/j.enbuild.2010.04.006.