From RNNs to Foundation Models: An Empirical Study on Commercial Building Energy Consumption

Shourya Bose
UC Santa Cruz
shbose@ucsc.edu

Yijiang Li
Argonne National Laboratory
yijiang.li@anl.gov

Amy Van Sant
National Renewable Energy Laboratory
amy.vansant@nrel.gov

Yu Zhang
UC Santa Cruz
zhangy@ucsc.edu

Kibaek Kim
Argonne National Laboratory
kimk@anl.gov

Abstract

Accurate short-term energy consumption forecasting for commercial buildings is crucial for smart grid operations. While smart meters and deep learning models enable forecasting using past data from multiple buildings, data heterogeneity from diverse buildings can reduce model performance. The impact of increasing dataset heterogeneity in time series forecasting, while keeping size and model constant, is understudied. We tackle this issue using the ComStock dataset, which provides synthetic energy consumption data for U.S. commercial buildings. Two curated subsets, identical in size and region but differing in building type diversity, are used to assess the performance of various time series forecasting models, including finetuned open-source foundation models (FMs). The results show that dataset heterogeneity and model architecture have a greater impact on post-training forecasting performance than the parameter count. Moreover, despite the higher computational cost, finetuned FMs demonstrate competitive performance compared to base models trained from scratch.

1 Introduction

In smart grids, accurate forecasting of future load consumption across different timescales is critical for both operational tasks, such as generator scheduling, and long-term planning, such as capacity expansion. A key area of focus is short-term load forecasting (STLF), which predicts energy consumption over timeframes from an hour to a day, at the level of individual buildings (Espinoza et al., 2007). The canonical pipeline for STLF involves collecting smart meter data of multiple buildings into a single dataset, training a single model on the same, and validating it on data split along the time axis (Kong et al., 2019). Pooling large datasets is particularly crucial for training large models, including foundation models, as it enables the models to learn diverse patterns and complex relationships from a wide variety of buildings. This global approach of training on pooled datasets requires forecasting models to effectively manage the heterogeneity that arises from differences in building data (Hewamalage et al., 2022). The heterogeneity may manifest in scale, variance, seasonality, or trend of different time series.

We use the ComStock dataset (Parker et al., 2023) to study the effect of heterogeneity on various STLF models. It is a synthetic dataset containing energy consumption of the U.S. commercial building stock alongside additional features such as building type. We leverage these features to control heterogeneity in curated ComStock subsets by varying building type diversity. We generate two such datasets, with one exhibiting greater heterogeneity. In this context, we evaluate multiple time series forecasting models from the literature, broadly categorized into two types:
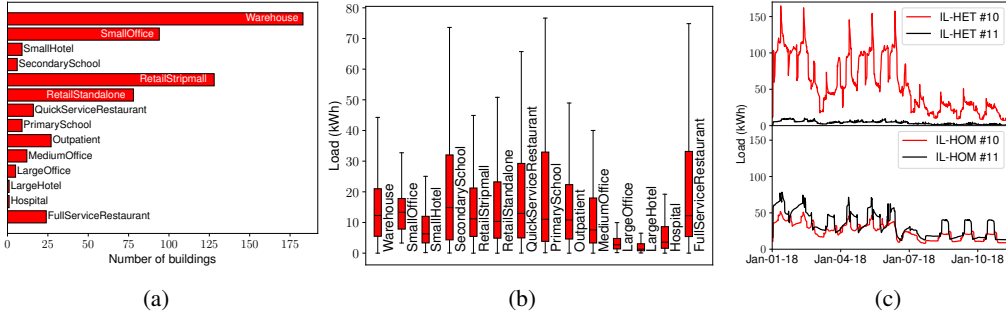
Figure 1: (a) Distribution of building type in `IL-HET`. (b) Boxplot of the spread of load values for each building type in `IL-HET`. (c) Load shapes from two buildings each from `IL-HET` and `IL-HOM`.

base models and foundation models (FMs). Base models are trained from scratch on a given dataset, while FMs are pre-trained on a large corpora, and only need to be finetuned for the given downstream application. We sample across a wide cross-section of available base models: recurrent neural networks (RNNs) (LSTM (Hochreiter and Schmidhuber, 1997), LSTNet (Lai et al., 2018)), transformer-based architectures (Transformer (Vaswani et al., 2017), Informer (Zhou et al., 2021)), decomposition-based architectures (Autoformer (Wu et al., 2021)), patch-based architectures (PatchTST (Nie et al., 2023)), and 2D-backbone architectures (TimesNet (Wu et al., 2023)). For foundation models, we select TimesFM (Das et al., 2024), which handles univariate inputs and is tested in both zero-shot (ZS) and finetuned (FT) modes. This setup enables us to assess the benefits of pre-training alongside the impact of heterogeneity.

This paper makes three key contributions: (i) the introduction of two curated building energy consumption datasets from ComStock, which are of the same size but vary in heterogeneity; (ii) a performance comparison of various base models on these datasets; and (iii) an evaluation of foundation models in both zero-shot and finetuned modes, compared to base models.

2 ComStock Dataset

ComStock, developed by the National Renewable Energy Laboratory and funded by the U.S. Department of Energy, is a detailed model of the U.S. commercial building stock. It uses statistical sampling methods and advanced energy simulations via OpenStudio (Guglielmetti et al., 2011) to estimate energy consumption across the building stock using 350,000 unique building energy models. These models are created from over 80 input distributions, including building characteristics such as building type, construction year, HVAC system, and occupancy schedules. The dataset provides detailed outputs like energy consumption at 15-minute resolution across different end uses and critical metrics, resulting in approximately 10 TB of data per release.

For our study, we distill the ComStock data into two curated datasets from buildings in Illinois, termed `IL-HET` (heterogeneous) and `IL-HOM` (homogeneous). Each dataset contains data from 592 building models for the actual meteorological year 2018. `IL-HET` includes all 14 available building types in ComStock (see Figure 1a), maintaining the overall dataset distribution, thus introducing heterogeneity. In contrast, `IL-HOM` comprises only buildings of type `Warehouse`, providing a more homogeneous set due to correlations among similar building loads. The heterogeneity in `IL-HET` is illustrated through its temporal spread in Figure 1b, while `IL-HOM` demonstrates a more uniform pattern (Appendix A). Both datasets are available at at `https://huggingface.co/datasets/APPFL/Illinois_load_datasets`.

Apart from the load in kWh, we incorporate several features in the dataset. We incorporate temporal information using two categorical features that represent the index of the 15-minute interval of the day and the day of the week. As mentioned earlier, we include the actual meteorological year 2018 weather features by the way of temperature and wind speed. Furthermore, we include three building-specific static features *viz* floor space, wall area, and window area. This feature selection is broader than the one in conventional energy datasets (see Appendix A), representative of the typical data used by utility companies (Jawed, 2018), and is validated via correlation analysis on a separate selection of buildings not included in either `IL-HET` or `IL-HOM` (see Appendix B).

2

3 Experiment Design

A pointwise time series forecasting problem can be generally represented as

$$\hat{\mathbf{y}}_{t+T} = f_{\boldsymbol{\theta}}(\mathbf{y}_{[t-L+1:t]}, \mathbf{u}_{[t-L+1:t+T]}, \mathbf{x}_{[t-L+1:t]}, \mathbf{s}),$$

where, for our experiments, $\mathbf{y}_t \in \mathbb{R}^1$ is the building energy consumption, $\mathbf{x} \in \mathbb{R}^2$ includes the two weather features, $\mathbf{u}_t \in \mathbb{R}^2$ denotes two time induces, and $\mathbf{s} \in \mathbb{R}^3$ includes the three static building characteristics, and T and L are the 'lookahead' and 'lookback' durations, respectively. The forecast model $f(\cdot)$ that is parameterized by the vector $\boldsymbol{\theta} \in \mathbb{R}^h$ can be trained in a supervised fashion, followed by evaluating its performance on unseen test sets. We use two metrics: *normalized mean squared error (NMSE)* and *normalized mean absolute error (NMAE)*, which given time indices of interest $t \in \mathcal{T}$, are defined as $\text{NMSE} := \frac{1}{n|\mathcal{T}|} \sum_{t \in \mathcal{T}} \|\hat{\mathbf{y}}_t - \mathbf{y}_t\|_2^2 / \sigma_{\mathbf{y}}^2$ and $\text{NMAE} := \frac{1}{n|\mathcal{T}|} \sum_{t \in \mathcal{T}} \|\hat{\mathbf{y}}_t - \mathbf{y}_t\|_1 / \sigma_{\mathbf{y}}$. Here $\sigma_{\mathbf{y}}$ is the normalizing factor. NMSE is used as the loss function for training.

The dataset is divided into train, validation, and test sets in a ratio of $0.8 : 0.1 : 0.1$ along the time axis for each building. Each set is z-normalized feature-wise using statistics from the train set, and the standard deviation of energy consumption is used as $\sigma_{\mathbf{y}}$. Data from all buildings are then concatenated to form unified datasets. Our choice of models (cf. Section 1) aims to provide a comprehensive evaluation of various time series models from the literature. Detailed descriptions of the model categories are provided in Appendix C. Since the pretraining of TimesFM was done by Das et al. (2024) for a maximum lookback of 512, this becomes the limiting factor in choosing the lookback of all models. Correspondingly, all models use $L = 512$. Furthermore, we only use the energy consumption feature with TimesFM since it is a univariate model.

Owing to the diversity of base models as well as TimesFM, we carry out hyperparameter tuning for inferring the optimal learning rate for each model, heterogeneity level and lookahead separately, the details of which are provided in Appendix D. The base models are trained for 20 epochs, while TimesFM is finetuned using full-parameter fine tuning for 10 epochs. The impact of using parameter-efficient finetuning for TimesFM is in the scope of future work. All models are trained with a batch size of 1024, and the training of each model uses a node with 4 Nvidia A100 GPUs. The final trained model weights are available at `https://huggingface.co/APPFL/Building_load_forecasting`, while zero-shot experiments with TimesFM are performed using the off-the-shelf pre-trained `timesfm-1.0-200m` (`https://huggingface.co/google/timesfm-1.0-200m`) without any modification.

4 Experiment Results

The performance of the trained models on the test set is summarized in Table 1, and predictions on the test set are visualized in Appendix E. Among the base models, the best performance on all experiments is achieved by TimesNet except one case where PatchTST outperforms it. This implies that the 2D-backbone of TimesNet and the periodicity-based patch generation of PatchTST are both effective in learning the patterns needed to generate high-quality forecasts. On the other hand, the relatively lower performance of Transformer, Autoformer, and Informer can be attributed to loss of temporal information in self-attention mechanisms and embeddings (Zeng et al., 2023). This leads to such models being outperformed by simpler architectures such as LSTM and LSTNet. It is important to note that patch-based architectures such as PatchTST and TimesFM (whose underlying architecture is patch-based) performs better on `IL-HET` than `IL-HOM`, which is a phenomenon that could indicate an increase in their performance with higher heterogeneity in the training data. This will be further investigated in future works. We also note that our implementation of early-stopping (i.e. terminating training when continued improvements are no longer observed on the test set) yielded disparate results across `IL-HET` and `IL-HOM` on base models, with `IL-HET` training terminating earlier across most models. This is discussed in Appendix D.

Comparing the base models to TimesFM, we see that the fined-tuned TimesFM is able to outperform the best base models, and by a large margin on the `IL-HET` dataset. It is also worth pointing out that TimesFM is a univariate model while the best base models, e.g., TimesNet and PatchTST, are multivariate models. This demonstrates the inherent benefit of the vast corpora of information absorbed by the model during pre-training when it comes to downstream forecasting tasks. Nonetheless, we should take the differences in model sizes between these models into account
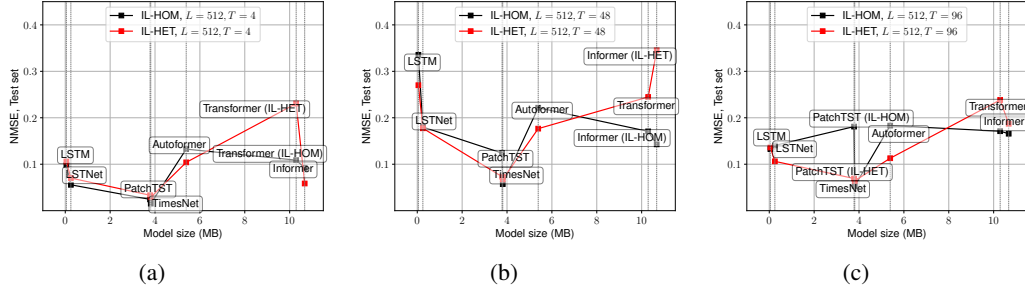
Figure 2: Test set performance of base models for $L = 512$ and (a) $T = 4$, (b) $T = 48$, (c) $T = 96$.

as well. On the other hand, the zero-shot performance of TimesFM is inadequate compared to other base models, highlighting the importance of full-parameter finetuning to unlock FM's potentials.

Finally, we analyze the importance of base model size on performance in Figure 2. This not only visually reinforces the impact of increasing heterogeneity on model performance, but also highlights the point that model architecture is more important than brute-forcing model size. Evidently, the trend that relatively small models such as LSTM, LSTNet, TimesNet, and PatchTST can outperform bigger models highlights the importance of efficiently learning temporal relations and feature-dependence on forecasts, rather than increasing parameter sizes.

Table 1: Experiment results for different models tested on the two datasets. The best results for each (L, T) pair are in **bold**, while the second-best is *italicized and underlined*.

IL-HET										
(L,T)	**LSTM**		**LSTNet**		**Transformer**		**Autoformer**		**Informer**	
	NMSE	NMAE	NMSE	NMAE	NMSE	NMAE	NMSE	NMAE	NMSE	NMAE
$(512, 4)$	0.1059	0.1623	0.0702	0.1280	0.2306	0.2316	0.1040	0.1689	0.5850	0.6808
$(512, 48)$	0.3464	0.2701	0.1512	0.1774	0.2551	0.2447	0.1208	0.1766	0.3248	0.3451
$(512, 96)$	0.1361	0.1568	0.1062	0.1449	0.2384	0.2299	0.1129	0.1727	0.1871	0.1808

(L,T)	**TimesNet**		**PatchTST**		**TimesFM (ZS)**		**TimesFM (FT)**	
	NMSE	NMAE	NMSE	NMAE	NMSE	NMAE	NMSE	NMAE
$(512, 4)$	*0.0289*	*0.0588*	0.0332	0.0598	0.0371	0.0598	**0.0078**	**0.0295**
$(512, 48)$	*0.0674*	*0.0924*	0.0732	0.0953	0.1072	0.1159	**0.0341**	**0.0687**
$(512, 96)$	0.0642	0.0892	*0.0641*	*0.0870*	0.1050	0.1125	**0.0339**	**0.0657**

IL-HOM										
(L,T)	**LSTM**		**LSTNet**		**Transformer**		**Autoformer**		**Informer**	
	NMSE	NMAE	NMSE	NMAE	NMSE	NMAE	NMSE	NMAE	NMSE	NMAE
$(512, 4)$	0.0945	0.1457	0.0557	0.1259	0.1058	0.1572	0.1325	0.1994	0.0920	0.1538
$(512, 48)$	0.3355	0.2858	0.1891	0.2148	0.1707	0.2081	0.2216	0.2762	0.1428	0.1966
$(512, 96)$	0.1326	0.1868	0.1404	0.1997	0.1711	0.2110	0.1831	0.2358	0.1657	0.2264

(L,T)	**TimesNet**		**PatchTST**		**TimesFM (ZS)**		**TimesFM (FT)**	
	NMSE	NMAE	NMSE	NMAE	NMSE	NMAE	NMSE	NMAE
$(512, 4)$	*0.0182*	*0.0684*	0.0432	0.0868	0.0559	0.1025	**0.0054**	**0.0309**
$(512, 48)$	*0.0572*	*0.1230*	0.1239	0.1864	0.3111	0.2873	**0.0417**	**0.1003**
$(512, 96)$	*0.0532*	*0.1163*	0.1166	0.1810	0.2911	0.2686	**0.0495**	**0.1065**

5   Concluding Remarks

In this paper, we tested various base models, as well as a finetuned and zero-shot FM on two datasets, one more heterogeneous than the other. The results demonstrate the impact of heterogeneity on model performance, the capability of models like TimesNet and PatchTST to be robust against heterogeneity, and the superior capability of finetuned FMs on both datasets. These observations serve towards grounding the problem of heterogeneity in time series models and act as stepping stones in domains such as federated learning, where data heterogeneity is inherent to the domain.

Acknowledgments and Disclosure of Funding

References

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st Intl. Conf. on Machine Learning*, pages 10148–10167, 2024. URL `https://proceedings.mlr.press/v235/das24c.html`.

Patrick Emami, Abhijeet Sahu, and Peter Graf. Buildingsbench: A large-scale dataset of 900k buildings and benchmark for short-term load forecasting. In *Proceedings of the 37th Conf. on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL `https://openreview.net/forum?id=c5rqd6PZn6`.

Marcelo Espinoza, Johan A.K. Suykens, Ronnie Belmans, and Bart De Moor. Electric load forecasting. *IEEE Control Systems Magazine*, 27(5):43–57, 2007.

Rob Guglielmetti, Daniel Macumber, and Nicholas Long. Openstudio: An open source integrated analysis platform. In *Proceedings of Building Simulation 2011: 12th Conf. of IBPSA*, volume 12 of *Building Simulation*, pages 442–449, Sydney, Australia, November 2011. IBPSA. URL `https://publications.ibpsa.org/conference/paper/?id=bs2011_1245`.

Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Global models for time series forecasting: A simulation study. *Pattern Recognition*, 124:108441, 2022. ISSN 0031-3203. URL `https://www.sciencedirect.com/science/article/pii/S0031320321006178`.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Ferhaan Jawed. Electric program investment charge (EPIC) 2.07 – real time loading data for distribution operations and planning. `https://www.pge.com/assets/pge/docs/about/corporate-responsibility-and-sustainability/PGE-EPIC-Project-2.07.pdf`, 2018.

Weicong Kong, Zhao Yang Dong, Youwei Jia, David J. Hill, Yan Xu, and Yuan Zhang. Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Trans. on Smart Grid*, 10(1):841–851, 2019.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *Proceedings of the 41st Intl. ACM SIGIR Conf. on Research & Development in Information Retrieval*, SIGIR '18, page 95–104, New York, NY, USA, 2018. Association for Computing Machinery.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=Jbdc0vTOcol`.

Andrew Parker, Henry Horsey, Matthew Dahlhausen, Marlena Praprost, Christopher CaraDonna, Amy LeBar, and Lauren Klun. Comstock reference documentation (v. 1). Technical Report NREL/TP-5500-83819, National Renewable Energy Laboratory (NREL), Golden, CO (United States), 2023. URL `https://www.nrel.gov/docs/fy23osti/83819.pdf`.

James R Schofield, Richard Carmichael, S Tindemans, Mark Bilton, Matt Woolf, Goran Strbac, et al. Low carbon london project: Data from the dynamic time-of-use electricity pricing trial, 2013. *UK Data Service, SN*, 7857(2015):7857–7851, 2015.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

Artur Trindade. ElectricityLoadDiagrams20112014. UCI Machine Learning Repository, 2015. DOI: https://doi.org/10.24432/C58C86.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems*, 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Proceedings of Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=J4gRj6d5Qm`.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *Proceedings of the 7th Intl. Conf. on Learning Representations*, 2023. URL `https://openreview.net/forum?id=ju_Uqw384Oq`.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9):11121–11128, Jun. 2023. doi: 10.1609/aaai.v37i9.26317. URL `https://ojs.aaai.org/index.php/AAAI/article/view/26317`.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the 35th AAAI Conference*, 2021. URL `https://ojs.aaai.org/index.php/AAAI/article/view/17325`.

A   Analysis of `IL-HET` and `IL-HOM`

Table 2: Statistics for loads in `IL-HOM` and `IL-HET`.

Dataset	Mean (loads)	Standard deviation (loads)
IL-HOM	17.4307	24.3129
IL-HET	21.8440	51.3635

Both `IL-HET` and `IL-HOM` contain load values of 592 buildings, along with 7 other features. Since the data is recorded at a granularity of 15 minutes for one year, the total number of data points are 35060. Thus, each of the datasets contain a total of 166,044,160 observations. Considering the fact that both datasets contain the same number of points, the higher heterogeneity of `IL-HET` can be seen by compaing the means and standard deviations of the loads.
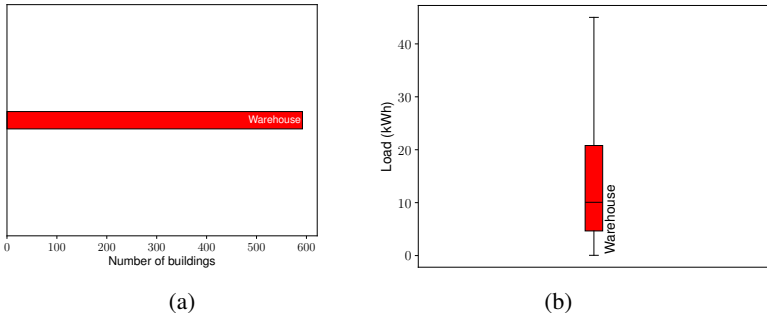

Figure 3: (a) Distribution of building type in `IL-HOM`. (b) Boxplot of the spread of values in the time series of each building type in `IL-HOM`.

Further evidence of the homogeneity of `IL-HOM` can be inferred from the equivalent figures to Figures 1a and 1b as presented in Figures 3a and 3b. While both figures contain only one element thanks to `IL-HOM` only containing buildings of the type `Warehouse`, the lower spread as well as variance of loads is evident from Figure 3b as compared to `IL-HET`.

Table 3: `IL-HOM` and `IL-HET` compared to other building energy datasets.

Dataset	# buildings	Sampling frequency	Duration	Total load observations	Non-time features
IL-HOM	592	15 minutes	1 year	20,755,520	5
IL-HET	592	15 minutes	1 year	20,755,520	5
Electricity Load Diagrams	370	15 minutes	4 years	51,894,720	0
London Smart Meters	5,567	30 minutes	~2 years	168,000,000	0
Buildings-900K	900,000	1 hour	2 years	15,777,000,000	3

We compare `IL-HOM` and `IL-HET` to other building-level load datasets, including Electricity Load Diagrams (Trindade, 2015), Low Carbon London (Schofield et al., 2015), and Buildings-900K (Emami et al., 2023). While other datasets exceed ours in terms of the number of data points, we provide a larger number of features as shown in Table 3, which provides important context to load forecasting models. As a matter of fact, the only other dataset on the list with non-time features is Buildings-900K which includes the latitude/longitude of the building's PUMA (Public Use Microdata Area) and a binary indicator for whether the building is commercial or residential.

B   Correlation Analysis for Feature Selection

In this section, we discuss our choice of features shown in Section 2. For this analysis, we consider 42 randomly selected Illinois buildings from ComStock which are not in either `IL-HOM` or `IL-HET` (which acts as a kind of 'validation set' aiding in the construction of the datasets). We split our analysis into two parts: choice of weather (dynamic) features and choice of static features. Weather data available with ComStock includes seven features *viz.* four temperature/wind features and three heat radiation features. We ignore the latter since they are not available in typical meteorological reports. Among the former four features, we calculate their Pearson correlation coefficient with the load values, and average the correlation across all 42 buildings. The results are presented in Table 4.

Table 4: Correlation analysis of weather features.

Feature name	Average correlation coefficient
Dry Bulb Temperature [$^\circ C$]	0.2507
Wind Speed [m/s]	0.1396
Wind Direction [Deg]	0.0787
Relative Humidity [%]	-0.2794

From Table 4, we select Dry Bulb Temperature [$^\circ C$] and Wind Speed [m/s], since Wind Direction [Deg] is almost uncorrelated with load values and Relative Humidity [%] is negatively correlated. On the other hand, ComStock contains many static features to choose from. In order to calculate static features' correlation to loads, we average the loads across all time indices and calculate the correlation of the averaged loads with the building's static features.

Table 5: Correlation analysis of static features.

Feature name	Correlation coefficient
External window area (m^2)	0.9714
Floor space (ft^2)	0.9245
External wall area (m^2)	0.7385
Number of air loops	0.7123
Total natural gas consumption	0.7112
Hot water supply volume (m^3)	0.2672
Occupant equivalent full load hour (EFLH)	0.2672
HVAC cooling type	0.1386

We choose the top three correlated features in order to prevent overcrowding of input data to the models with static features. Thus, the aforementioned analysis informs our choice of features, which in turn determines the models' learning capabilities.

C   Model Architectures

Our choice of base models is meant to capture the major trends towards time series forecasting using deep architectures in recent years. To that end, we summarise these trends and the positioning of our chosen architectures within those trends.

- **Recurrent Neural Network Architectures:** These models consist of a learnable kernel that is unrolled recursively in time to generate forecasts. In this category, we use long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) with a dense decoder, and LSTNet (Lai et al., 2018), which combines RNNs with convolutional layers.
- **Transformer-based Architectures:** These models use self-attention and cross-attention mechanisms in an encoder-decoder framework to parallelize the generation of the output sequence, given an input sequence. In this category, we use the vanilla Transformer (Vaswani et al., 2017) and Informer (Zhou et al., 2021), with the latter incorporating the *ProbSparse* attention mechanism.

Contrary to conventional implementations, we use positional embeddings for temporal inputs (i.e. **u**), alongside the standard positional embeddings.

- **Decomposition-based Architectures:** These models replace the attention mechanism of the Transformer with other mechanisms that leverage inherent characteristics of sequential data. In this category, we use Autoformer (Wu et al., 2021), which uses an auto-correlation mechanism, alongside the decomposition of sequences into seasonal and trend patterns.

Table 6: Model sizes, parameter counts and structural hyperparameters for ($T = 512, L = 48$).

Model name	Model hyperparameters	Parameter count	Model size
LSTM	Hidden size: 32 LSTM Layers: 1 Dropout: 0.1	13,856	0.05 MB
LSTNet	Conv. layer output channels: 32 Conv. kernel size: 12 GRU hidden size: 128 Recurrent layer skips: $\{4, 4\}$ Attention window size: 7 Dropout: 0.1	66,473	0.25 MB
Transformer	Token size: 128 Num. heads: 4 Encoder layers: 3 Decoder layers: 3 FCNN hidden dimension: 512 Activation: GeLU Dropout: 0.1	2,698,625	10.29 MB
Autoformer	Token size: 128 Num. heads: 4 Encoder layers: 3 Decoder layers: 3 FCNN hidden dimension: 512 Moving average window: 25 TA Delay top-k: $5\log(L)$ Dropout: 0.1	1,413,633	5.39 MB
Informer	Token size: 128 Num. heads: 4 Encoder layers: 3 Decoder layers: 3 FCNN hidden dimension: 512 Activation: GeLU Dropout: 0.1	2,798,211	10.67 MB
TimesNet	Token size: 128 Encoder layers: 3 k for top-k amplitudes: 5 Num. kernels: 3 Inception block intermediate channels: 64 Dropout: 0.1	998,289	3.81 MB
PatchTST	Token size: 128 Num. heads: 3 Encoder layers: 3 Patch length: 16 Stride: 8 FCNN hidden dimension: 512	990,384	3.78MB

- **2D-backbone architectures:** These models convert input 1D temporal sequences into 2D tensors with properties that help the model better identify short- as well as long-term patterns in said sequences. In this category, we use TimesNet (Wu et al., 2023), which uses the Fourier transform to identify periodicity, which is then leveraged for tensor conversion. Following the same, the tensors are passed through 2D convolution-based Inception blocks (Szegedy et al., 2015) to generate the forecast.

- **Patch-based Architectures:** These models split the input sequence into smaller sub-sequences called *patches*, followed by learning the interactions between different patches to better predict the forecast. In this category, we use PatchTST (Nie et al., 2023), which uses patching in addition to channel-independence to generate forecasts.

A detailed description of the model sizes, parameter counts, and structural hyperparameters for the aforementioned models is provided in Table 6.

On the other hand, TimesFM is a decoder-only foundation model built for time series forecasting. The model consists of input layers, stacked transformers, and output layers and considers an NMSE training loss function. The pre-trained TimesFM model `timesfm-1.0-200m` is trained on a vast amount and variety of data with both synthetic data and real-world data. There are two major model hyperparameters, namely, `model_dim`, which determines the size of processed data, and `num_layers`, which controls the number of stacked transformers in the model. A summary of the model size, parameter counts, and structural hyperparameters in TimesFM is provided in Table 7.

Table 7: Model sizes, parameter counts and structural hyperparameters for TimesFM (for a comprehensive list, see (Das et al., 2024))

Model name	Model hyperparameters	Parameter count	Model size
TimesFM	Token size: 1280 Num. Layers: 20	200m	854MB

D Hyperparameter Tuning

Given these models span a wide range of architectures, using the same value for the learning rate for all of them would lead to poor performance. To that end, we tuned the learning rate over the validation set for all lookaheads ($T = 4, 48, 96$) and all datasets (`IL-HET` and `IL-HOM`) with choices $\text{lr} = \{10^{-3}, 5 \times 10^{-4}, 10^{-4}, 5 \times 10^{-5}, 10^{-5}, 5 \times 10^{-6}, 10^{-6}, 5 \times 10^{-7}\}$. The results are reported in Tables 8 and 9.

Table 8: Learning rate tuning over the validation set on the dataset `IL-HET`. The metric is NMSE, and the best learning rates are highlighted.

Model	10^{-3}	5×10^{-4}	10^{-4}	5×10^{-5}	10^{-5}	5×10^{-6}	10^{-6}	5×10^{-7}
				$L = 512, T = 4$				
LSTM	0.8258	0.2535	0.2020	**0.1853**	0.2804	0.2823	0.3878	0.4077
LSTNet	0.4564	0.2236	**0.1417**	0.1709	0.2202	0.3225	0.3207	0.3397
Transformer	3.0464	3.0721	0.8025	1.2834	1.5921	1.0574	**0.4246**	0.6596
Autoformer	0.6904	0.5245	0.5835	0.4326	0.3083	0.3834	0.2688	**0.2649**
Informer	3.9926	3.7363	3.0798	3.1364	1.5351	**0.7345**	1.0424	1.4432
TimesNet	0.0699	0.0667	0.0728	0.0627	**0.0603**	0.0703	0.1000	0.1088
PatchTST	0.0678	0.0622	0.0628	**0.0597**	0.0619	0.0666	0.0829	0.0924
				$L = 512, T = 48$				
LSTM	3.8491	2.1324	0.3300	**0.2981**	0.3283	0.3052	0.3924	0.4104
LSTNet	0.2866	0.3833	0.2166	**0.2125**	0.3213	0.3544	0.3315	0.3432
Transformer	3.3272	3.3638	0.7665	0.8136	1.6502	1.0295	**0.4785**	0.7056
Autoformer	0.6104	0.5294	0.5150	0.4226	0.4226	0.3457	**0.2744**	0.2761
Informer	2.8727	3.0340	3.3786	3.6050	1.2244	**0.3451**	1.4153	1.9887
TimesNet	0.1043	0.1096	0.1117	0.1032	**0.0976**	0.0985	0.1121	0.1175
PatchTST	0.0967	**0.0953**	0.1009	0.0955	0.0984	0.1034	0.1082	0.1122
				$L = 512, T = 96$				
LSTM	0.7134	0.3811	0.2001	**0.1843**	0.2719	0.2809	0.3868	0.4069
LSTNet	0.8501	0.2931	0.1617	**0.1595**	0.2360	0.3235	0.3202	0.3397
Transformer	2.7635	3.0211	0.5371	1.3160	1.7465	1.1508	**0.4568**	0.6904
Autoformer	0.7147	0.4788	0.5125	0.4777	0.3339	0.3632	0.2721	**0.2704**
Informer	3.2822	3.1977	2.9025	3.6369	1.2768	1.4782	**1.1671**	2.0538
TimesNet	0.0966	0.0931	0.0973	0.0914	**0.0891**	0.0985	0.1129	0.1203
PatchTST	0.0874	**0.0874**	0.0897	0.0876	0.0908	0.0951	0.1014	0.1053

Table 9: Learning rate tuning over the validation set on the dataset `IL-HOM`. The metric is NMSE, and the best learning rates are highlighted.

Model	10^{-3}	5×10^{-4}	10^{-4}	5×10^{-5}	10^{-5}	5×10^{-6}	10^{-6}	5×10^{-7}
			$L = 512, T = 4$					
LSTM	0.2058	0.1832	**0.1554**	0.1618	0.2546	0.3488	0.5317	0.5421
LSTNet	0.2145	0.1604	**0.1490**	0.1819	0.3139	0.3417	0.4325	0.4871
Transformer	0.6288	0.5218	0.2916	**0.2189**	0.2717	0.2736	0.3208	0.3363
Autoformer	0.3000	0.2691	**0.2050**	0.2067	0.2350	0.2452	0.3032	0.3404
Informer	0.6004	0.5635	0.3656	0.2733	**0.2078**	0.2143	0.3040	0.3499
TimesNet	0.0835	**0.0804**	0.0933	0.0885	0.0976	0.1105	0.1863	0.2157
PatchTST	0.1049	**0.0908**	0.0987	0.1007	0.1099	0.1196	0.1466	0.1679
			$L = 512, T = 48$					
LSTM	0.3416	**0.3173**	0.3278	0.3371	0.3558	0.3897	0.5408	0.5500
LSTNet	0.2479	**0.2456**	0.2631	0.2912	0.4025	0.4197	0.4526	0.4988
Transformer	0.6229	0.3294	0.2991	0.3052	0.2859	**0.2834**	0.3273	0.3404
Autoformer	0.2992	0.2806	0.2806	**0.2762**	0.3030	0.3182	0.3428	0.3643
Informer	0.6450	0.5477	0.4009	0.3403	**0.2319**	0.2385	0.3393	0.3826
TimesNet	0.1439	0.1327	0.1258	**0.1234**	0.1322	0.1432	0.2065	0.2307
PatchTST	0.1973	**0.1889**	0.1994	0.1927	0.1986	0.2017	0.2176	0.2283
			$L = 512, T = 96$					
LSTM	0.2997	0.2389	**0.2321**	0.2402	0.3000	0.3673	0.5341	0.5444
LSTNet	0.2422	0.2350	**0.2257**	0.2566	0.3508	0.3741	0.4422	0.4911
Transformer	0.6371	0.3380	0.2984	0.3042	0.2915	**0.2897**	0.3288	0.3407
Autoformer	0.2990	0.2743	**0.2401**	0.2448	0.2766	0.2868	0.3327	0.3564
Informer	0.6068	0.5414	0.4018	0.3598	**0.2571**	0.2774	0.3366	0.3705
TimesNet	0.1503	0.1443	0.1325	**0.1299**	0.1523	0.1745	0.2247	0.2515
PatchTST	0.1855	0.1830	0.1856	**0.1809**	0.1883	0.1911	0.2037	0.2114

The training is carried out using the above model-specific learning rates. As described in Section 3, we use an early stopping criteria of no improvement in the validation set metrics (also called *patience*) for 5 epochs. On the other hand, all training terminates in 20 epochs whether or not the patience criteria is met. The epoch at which each model's training terminates is given in Table 10.

Table 10: Termination epoch for each trained model. The asterisk ($*$) indicates that early stopping was not triggered.

Model	`IL-HET`			`IL-HOM`		
	$T = 4$	$T = 48$	$T = 96$	$T = 4$	$T = 48$	$T = 96$
LSTM	18	20*	20*	4	4	17
LSTNet	2	12	6	10	6	8
Transformer	4	4	4	20*	20*	20*
Autoformer	20*	20*	20*	20*	1	5
Informer	2	1	11	6	3	11
TimesNet	3	3	1	20*	7	15
PatchTST	1	5	3	8	11	1

The observations of Table 10 can be explained by the fact that training of `IL-HET` is more unstable than `IL-HOM`, which leads to early stopping being triggered very early into the training run. This is further verified by letting Transformer for $L = 512, T = 96$ train to completion without early stopping on `IL-HET`, which reveals instability in training after epoch 4 (see Figure 4).
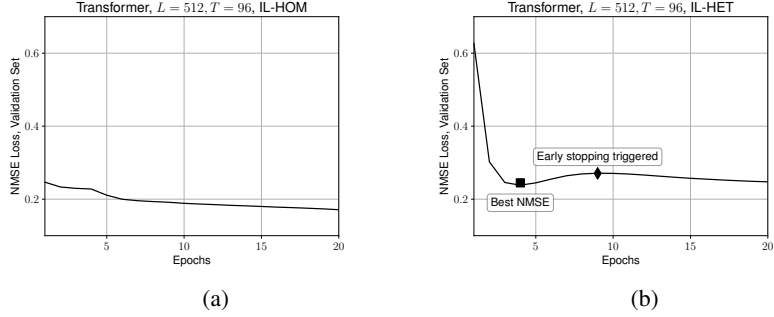
Figure 4: Training instability in Transformer training on `IL-HET` compared to `IL-HOM`.

Similar to the base models, the learning rate for TimeFM was tuned over the validation set of all lookaheads ($T = 4, 48, 96$) with choices $\mathtt{lr} = \{10^{-3}, 5 \times 10^{-4}, 10^{-4}, 5 \times 10^{-5}, 10^{-5}, 5 \times 10^{-6}, 10^{-6}, 5 \times 10^{-7}\}$ over 1 epoch. The results are reported in Tables 11 and 12. With $T = (48, 96)$, we perform additional experiments with $\mathtt{lr} = \{10^{-7}, 5 \times 10^{-8}\}$.

Table 11: Learning rate tuning over the validation set of the problem for TimesFM. The metric is NMSE, and the best learning rates are highlighted.

Model	10^{-3}	5×10^{-4}	10^{-4}	5×10^{-5}	10^{-5}	5×10^{-6}	10^{-6}	5×10^{-7}
(L, T)				`IL-HOM`				
$(512, 4)$	639.06	6.7198	0.0177	0.0166	**0.0074**	0.0076	0.0077	0.0082
$(512, 48)$	656.45	7.7152	0.0875	0.0512	0.0384	**0.0365**	0.0383	0.0404
$(512, 96)$	169.77	0.9800	0.1185	0.0483	0.0403	**0.0401**	0.0413	0.0432
(L, T)				`IL-HET`				
$(512, 4)$	2285.12	89.396	0.0331	0.0252	0.0144	0.0121	**0.0118**	0.0124
$(512, 48)$	247.92	10.034	0.0814	0.0724	0.0486	0.0473	0.0398	**0.0369**
$(512, 96)$	753.21	1.9964	0.0539	0.0568	0.0465	0.0462	0.0383	**0.0369**

Table 12: Additional learning rate tuning over the validation set of the problem for TimesFM for `IL-HET` with $(512, 48)$ and $(512, 96)$. The metric is NMSE, and the best learning rates are highlighted.

Model	10^{-7}	5×10^{-8}
(L, T)		`IL-HET`
$(512, 48)$	0.04121	0.04711
$(512, 96)$	0.04027	0.04585

Early-stopping based on *patience* is also implemented in finetuning the TimesFM model where validation is performed to compute the progress of the finetuning in terms of the NMSE loss every 1,000 steps in each epoch. *Patience* is increased by one if the NMSE loss does not decrease and the so-called early termination of the current epoch is executed if *patience* reaches five. In our finetuning experience, the early termination condition was never triggered.

E   Predictions

In Figure 5, we show the predictions for the first 500 points of the 5th customer in both the datasets. These visualizations offer insights into how well the trained models capture the underlying patterns in individual building data. However, it is important to note that the predictions for a single building may not fully represent the model's performance across the entire test set.
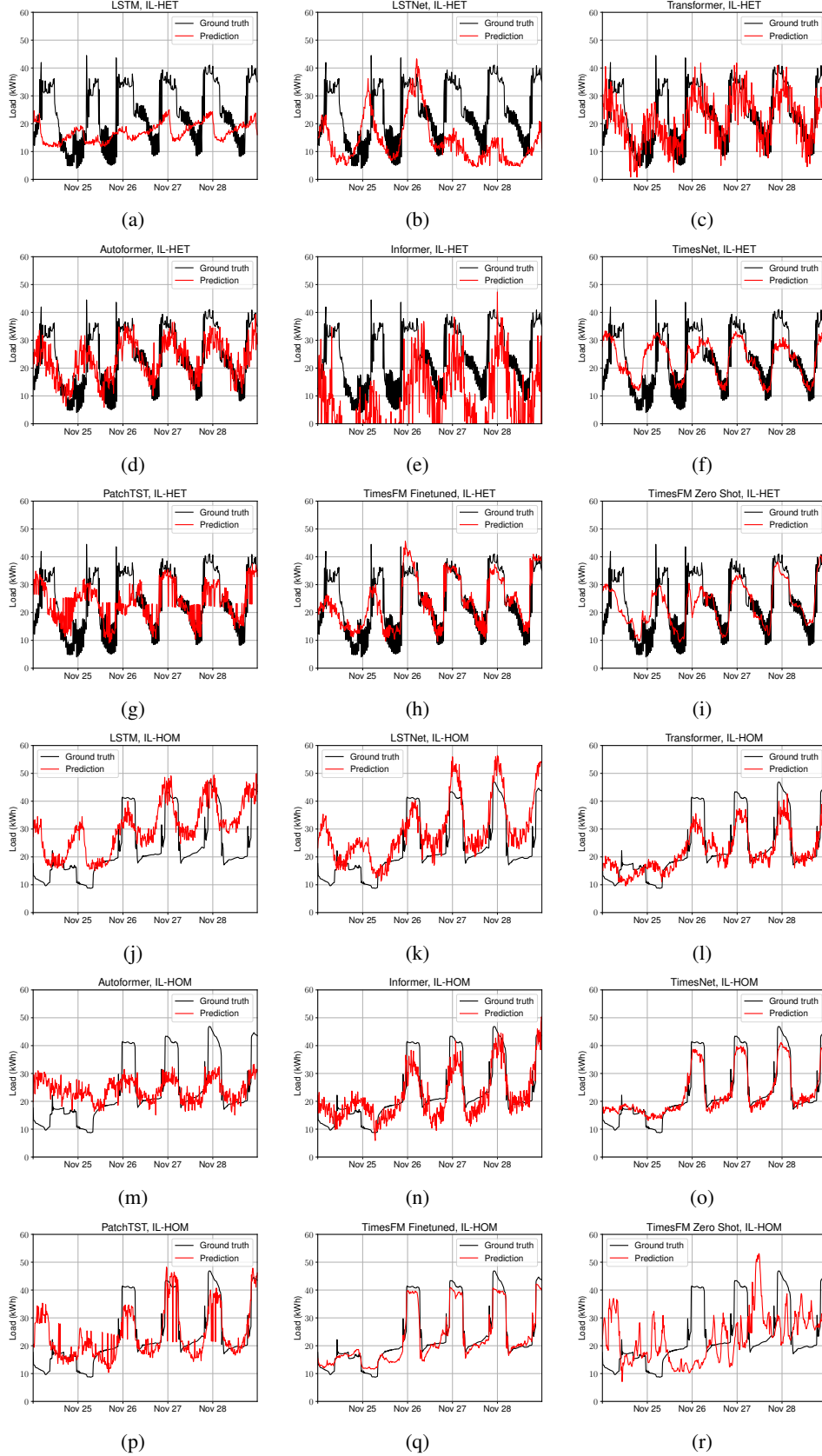
Figure 5: Predictions of the test set loads for two selected buildings from `IL-HOM` and `IL-HET` datasets for $T = 512, L = 48$, respectively, for the best models reported in Table 6.