Economics World, Jan.-Mar. 2025, Vol. 12, No. 1, 1-12
doi: 10.17265/2328-7144/2025.01.001

Predicting Energy Demands of Retail Stores—Can Deep Learning Help?

Levente Szabados, Csilla Obadovics
University of Sopron, Sopron, Hungary

In the context of the energy and climate crises, it is crucial for organizations to utilize advanced methods to reduce energy consumption and energy costs. This study explores the application of deep learning models for predicting energy demands in retail stores, which can enhance market efficiency and contribute to grid stability. We analyze a detailed electricity consumption dataset from a hypermarket in Hungary, focusing on 48-hour forecasts at 15-minute intervals. Our methodology includes the implementation of classical models such as ARIMA and linear regression, as well as state-of-the-art deep learning models like TiDE and foundational models such as Lag-Llama in a "zero shot prediction" as well as a "finetuning" scenario.

Keywords: energy demand prediction, Deep Learning, foundational time series models, transfer learning

Motivation

In the context of the climate crisis and the green energy transition, it is of paramount importance to utilize all possible means to reduce energy consumption. Electricity, as a perishable good, cannot be economically stored in large quantities, making market efficiency crucial for maintaining low prices. Better forecasts of short-term energy needs, coupled with the expansion of day-ahead and intraday energy trading platforms, can make energy more affordable for companies and enhance market efficiency.

This improved efficiency has significant economic benefits, especially in the retail sector, where profit margins are typically thin. For example, food retail companies can greatly benefit from reduced energy costs. Additionally, better energy demand forecasting has a positive effect on grid stability, since the increasing introduction of renewable energy sources, while beneficial for sustainability, can have a destabilizing effect on the grid (Biber et al., 2022). Accurate demand predictions can help mitigate these destabilizing effects by ensuring a more balanced and reliable energy supply.

Artificial intelligence (AI) techniques, particularly deep learning, are often criticized for their high energy consumption and thus seen as contributors to climate problems (Patterson et al., 2021), however, applying deep learning to predict retail energy demands could serve as a counterpoint, potentially leading to significant energy savings and demonstrating a valuable application of AI in addressing climate challenges. This dual benefit — economic and environmental—underscores the motivation for exploring deep learning techniques in this domain.

Levente Szabados, Dozent, Ph.D. candidate, Frankfurt School of Finance and Management, University of Sopron, Sopron, Hungary.
Csilla Obadovics, Ph.D., Professor, University of Sopron, Sopron, Hungary.
Correspondence concerning this article should be addressed to Levente Szabados,Frankfurt School of Finance and Management, University of Sopron, Sopron, Hungary.

Broader Context

Classical time series modeling techniques have long been valued for their robustness and reliability. These methods, including ARIMA Box and Jenkins, 1970, Exponential Smoothing (Winters, 1960) and State space models, have proven effective in many applications. However, they often struggle when confronted with more complex datasets characterized by high dimensionality, non-linearity, and intricate temporal dependencies.

The Makridakis Competitions contributors, 2024, known as the M-competitions, serve as a broad-ranging challenge to measure the state-of-the-art in time series fore-casting models. The M4 competition marked a significant shift towards the adoption of machine learning and, more specifically, deep learning techniques. Subsequent M5 and M6 competitions further demonstrated the increasing benefits of specialized deep learning architectures such as N-BEATS (Oreshkin et al., 2020) and TiDE (Das et al., 2023).

Parallel to these developments in time series forecasting, the field of Natural Language Processing (NLP) and computer vision witnessed the emergence of foundational models Bommasani et al. (2021). These models, pretrained on large datasets, exhibited remarkable transferability across various tasks, often delivering impressive "zero-shot" performance without the need for task-specific training.

In the realm of time series forecasting, this paradigm holds great promise. A new wave of techniques and models is emerging, many of which are designed to leverage the advantages of foundational models. Notable examples include TimeGPT (Garza, & Mergenthaler-Canseco, 2023), Unified (Woo et al., 2024), Decoder-Only (Das et al., 2024), Lag-Llama (Rasul et al., 2024), Chronos (Ansari et al., 2024), MOIRAI (Goswami et al., 2024), and MOMENT. These models, like Chronos, MOIRAI, MOMENT, and Lag-Llama, are often available as open-source, fostering a collaborative environment for further advancements. (For a broader survey of foundational models in time series domain see Ye et al. (2024).

In this article, we will endeavor to test and compare all these innovative techniques. By evaluating their performance on our dataset, we aim to determine their efficacy in predicting energy demands for retail stores, ultimately contributing to more efficient energy management and consumption.

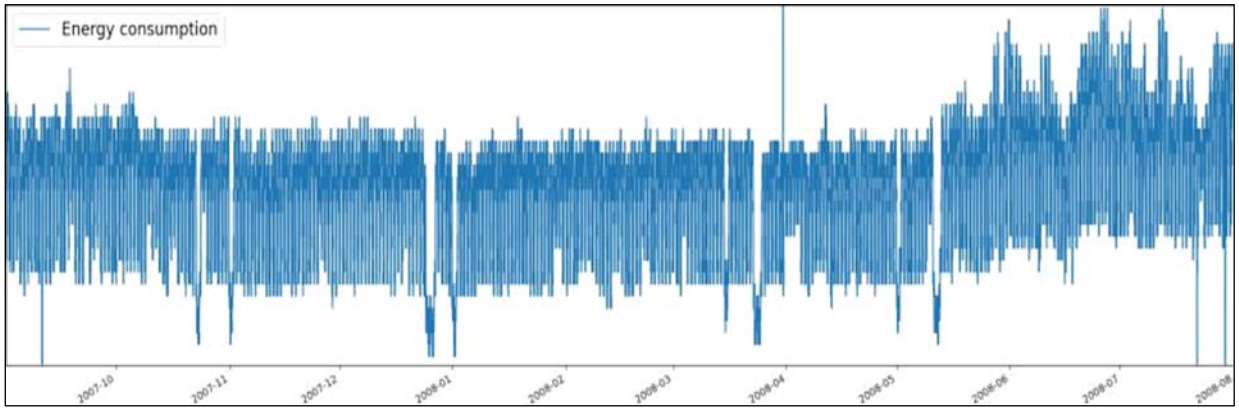

Figure 1. Time series dataset overview.

Data and Feature Engineering for Modeling

The proprietary dataset we use in this study is a detailed record of electricity consumption from a hypermarket in Hungary, recorded at 15-minute intervals over the span of nearly a year, from September 1, 2007, to July 31, 2008. Additionally, hourly temperature data for the same period was obtained from the Hungarian

National Meteorological Service's Climate Department. This dataset includes various critical variables such as weather conditions, which are essential for accurately predicting short-term electricity demand.

The data preparation process involved organizing and restructuring the dataset, creating new variables, and merging the consumption data with temperature records. Specific attention was given to handling outliers, such as power outages and holiday effects, to ensure the robustness of the predictive models. By analyzing this comprehensive dataset, the study aimed to develop a 48-hour forecast model, capable of estimating the electricity consumption with a 15-minute resolution, thereby aiding in reducing energy costs.

Before we applied multiple competing models on the dataset, we carried out an exploratory data analysis, that strongly informed our decisions in feature engineering for our models.

General Features of the Dataset

Analysing the dataset, we immediately realised, that there are a few outliers, that are orders of magnitude larger than any "normal" datapoint (see Figure 1), so winsorization had to be applied in order to not skew the descriptive statistics, as well as to ensure numeric stability for the modeling efforts downstream. We decided to clip the dataset below 80 and above 410.

Also, after investigation we found, that public holidays have a major effect, and are one of the main causes of outliers, hence we decided to include a binary feature about public holidays in Hungary for the later modeling phase (in case of multivariate capable models).

Despite the fact that the Dickey-Fuller stationarity test (Dickey, & Fuller, 1979) has a P value of basically 0, hence formally the time series can be considered stationary, there is a noticeable "regime change" in the data at around May 2008, which emphasizes the importance of "holdout" (validation set) at the end of the series for checking the model performance.

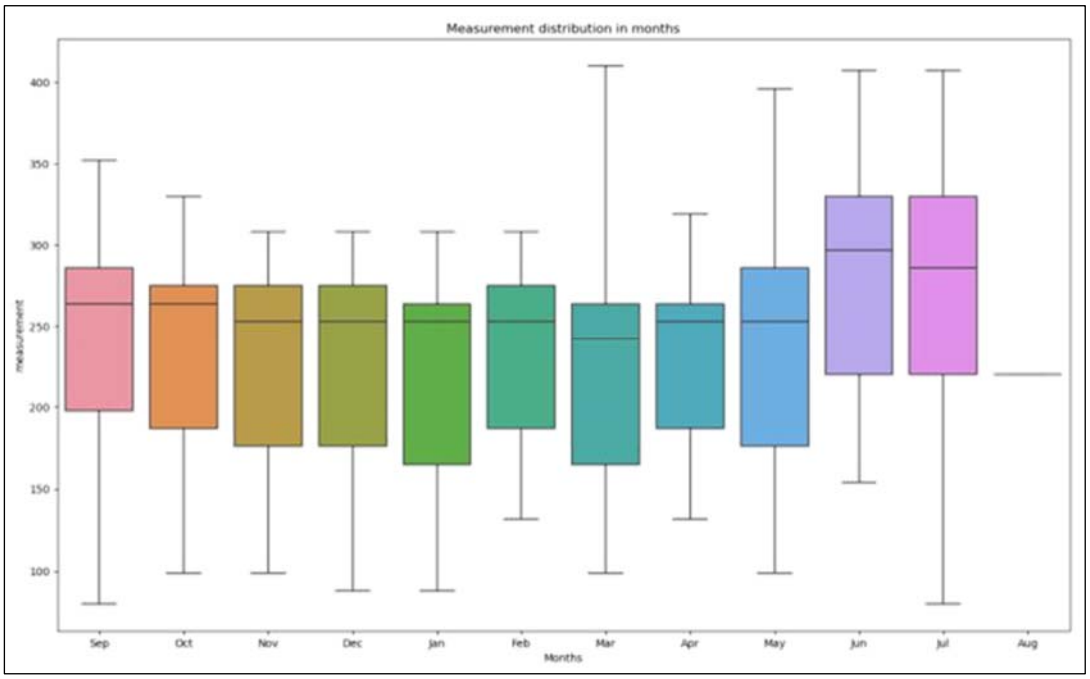

Figure 2. Monthly distribution.

Looking at the monthly distribution of data (see Figure 2) (even if August is basically missing) we can see, that generally the summer months are having a higher median, thus we can assume, the main energy consumption

comes from cooling.

The analysis of the weekly distribution (see Figure 3) shows stronger energy consumption on “shopping days” of Friday and Saturday, emphasizing, that a mere “weekend vs. weekday” feature is not enough, so a day of week feature will be essential in setting up our models.

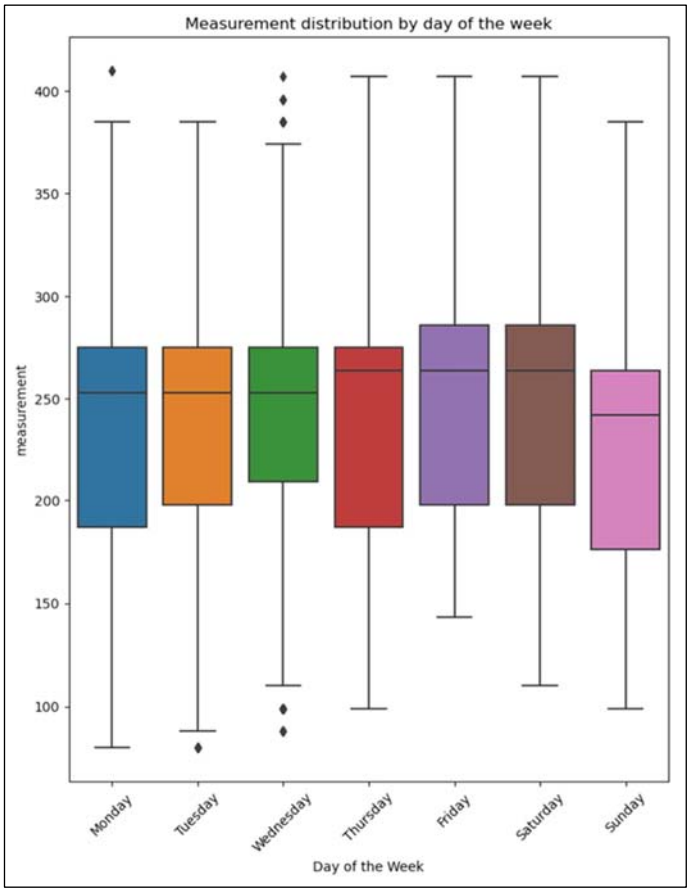

Figure 3. Weekly distribution.

There is also a noticeable change in the intraday (see Figure 4) patterns of different days in a week, so a granular encoding of the hour of the day together with the day of the week must be provided to our models.

Finally, we can also see (see Figure 5), that store traffic—and with it, energy consumption—is also influenced by some specific days in the month (usually around pay-days), so a day of month feature for our models seems useful.

Observing the dominant importance of time variables, especially those with a cyclical structure, like day of week, hour of day etc. and the fact that we want to utilize also some techniques for modeling, that can utilize additional data in “multivariate” settings, we chose to apply “sine and cosine” time encoding.

We created cyclical sine and cosine features for the following time elements: ‘month’, ‘day’, ‘weekday’, ‘day of week’, ‘hour’, ‘quarter’, ‘week of year’. (For the encoding we utilized the “Time Axes Encoders” (Developers, 2022b) functionality of the DARTS (Herzen et al., 2022) modeling framework.

Finally, as for the suitable time window for the time series models, we took a look at the partial autocorrelation structure of the time series. (see Figure 6)

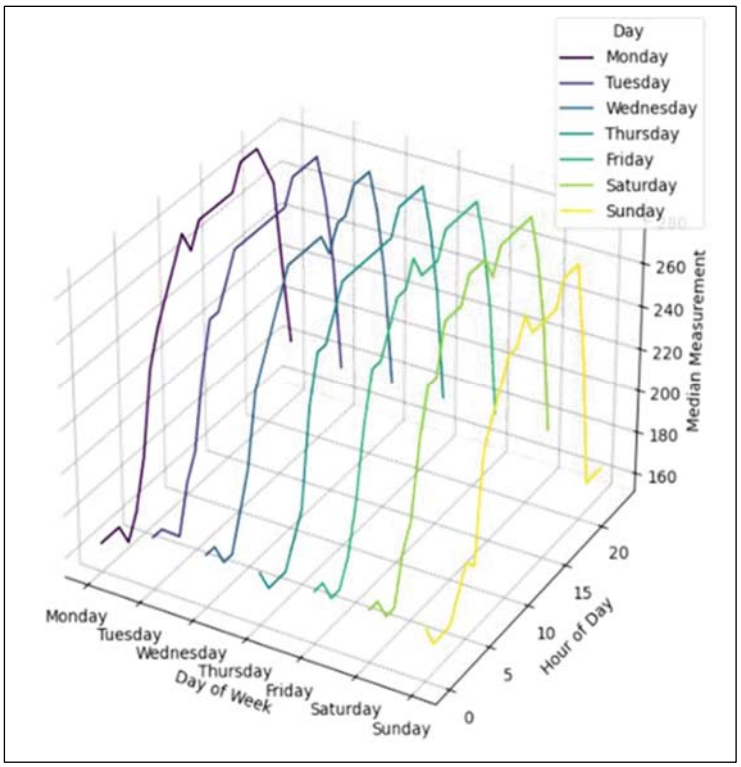

Figure 4. Daily distribution during the week.

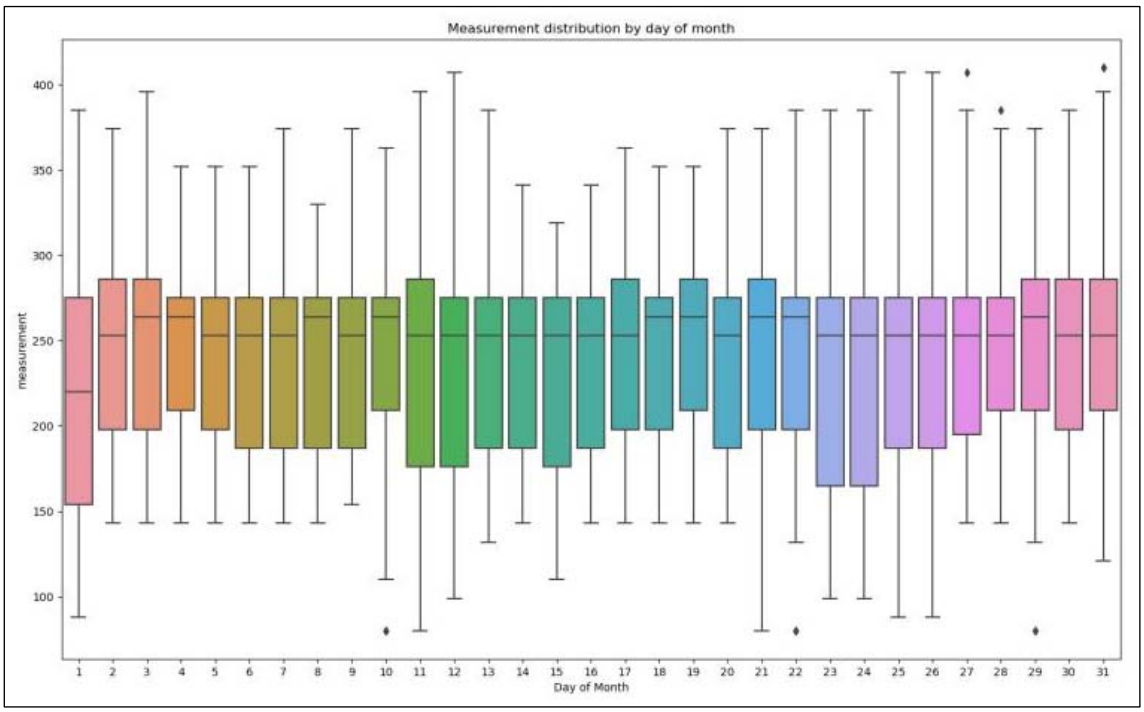

Figure 5. Days of the month.

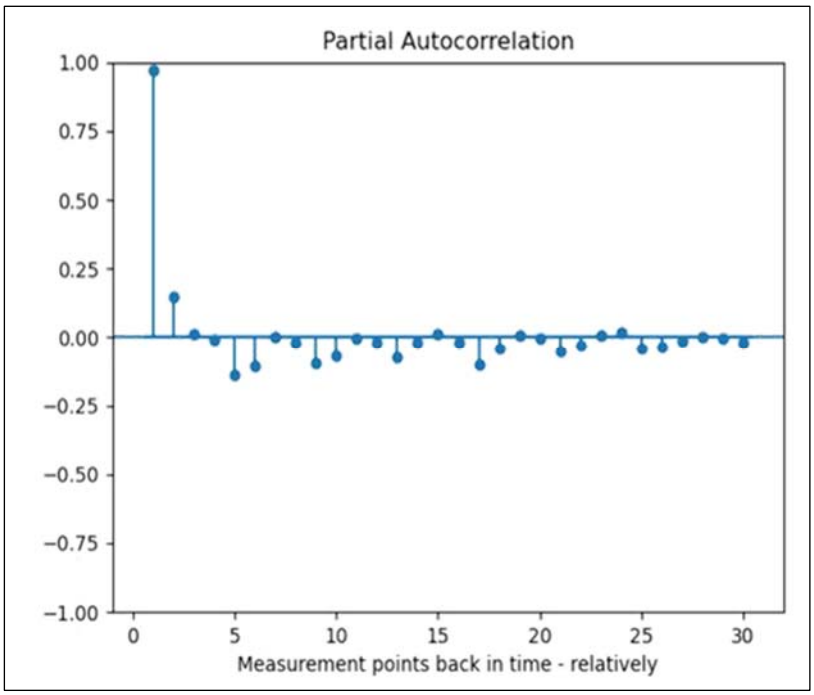

Figure 6. Partial autocorrelation.

We can see a longer list of still significant partial autocorrelations, that would hint to a longer modeling context window, so we choose a one-week lookback window (24 hours * 4 quarter hours * 7 days).

Temperature Dependence in Energy Consumption

The analysis of hourly temperature data from the nearby weather station and the measurement of energy consumption show a strong relationship. More specifically, even if we remove the weekly and daily seasonal patterns from the consumption measurement, there is significant observable Granger Causality (Granger, 1969) at lag 2. This leads us to assume, that with an approximately 2 hours delay the change in temperature has an effect on the energy consumption.

Based on this observation, for the multivariate capable models we included the local temperature as a covariate that will also be available in the future. In this case we assume, that an adequate weather forecast can be available during the later application of our models.

Modeling Approach

Motivated by the regime change observed in the dataset we choose to set a challenging split point and apply the holdout method for a train-validation-test split. The validation thus starts at 2008-05-18 and lasts till 2008-07-11, and we will reserve a final test set for later purposes. All results will be shown on the validation set.

Because of the fact that Deep Learning can not be feasibly applied below a certain number of data points (and we intended to explicitly compare it's performance to more "classical methods") we choose to keep the 15 minute interval for inputs and expected outputs alike, and based on the business requirements implemented a 48 hours (thus 48*4 intervals) forward prediction. (Any possible improvements with hourly resampling, we leave for future work.)

Our most naive baseline model was the autoregressive integrated moving average (ARIMA) model (Box, & Jenkins, 1970), specifically the "Auto-ARIMA" functionality (Developers, 2022a) "wrapped by" DARTS

(Herzen et al., 2022). Since this method produced unacceptable results, our next best baseline was a simple linear regression model (as implemented by Scikit-learn (Pedregosa et al., 2011), (again via DARTS), which had access to the time encoding as well as the temperature features.

Our approach to investigating the efficiency of deep learning models was two folds: for one, we started with a "dedicated" (so randomly initialized and trained only on the specific dataset) model representing the state-of-the-art, TiDE (Das et al., 2023).

As for the foundational models, we decided to test the zero-shot learning scenario, which is favoured by the proponents of many foundational models in general, and pre-trained time series models in particular, but we also wanted to see, if the pre-training and fine-tuning paradigm (represented in Natural Language Processing by Howard, and Ruder (2018) is applicable here. Because of it's ready availability to such endeavors, we decided to utilize the Lag-Llama model from Rasul et al. (2024), available on Github (Garcia et al., 2024).

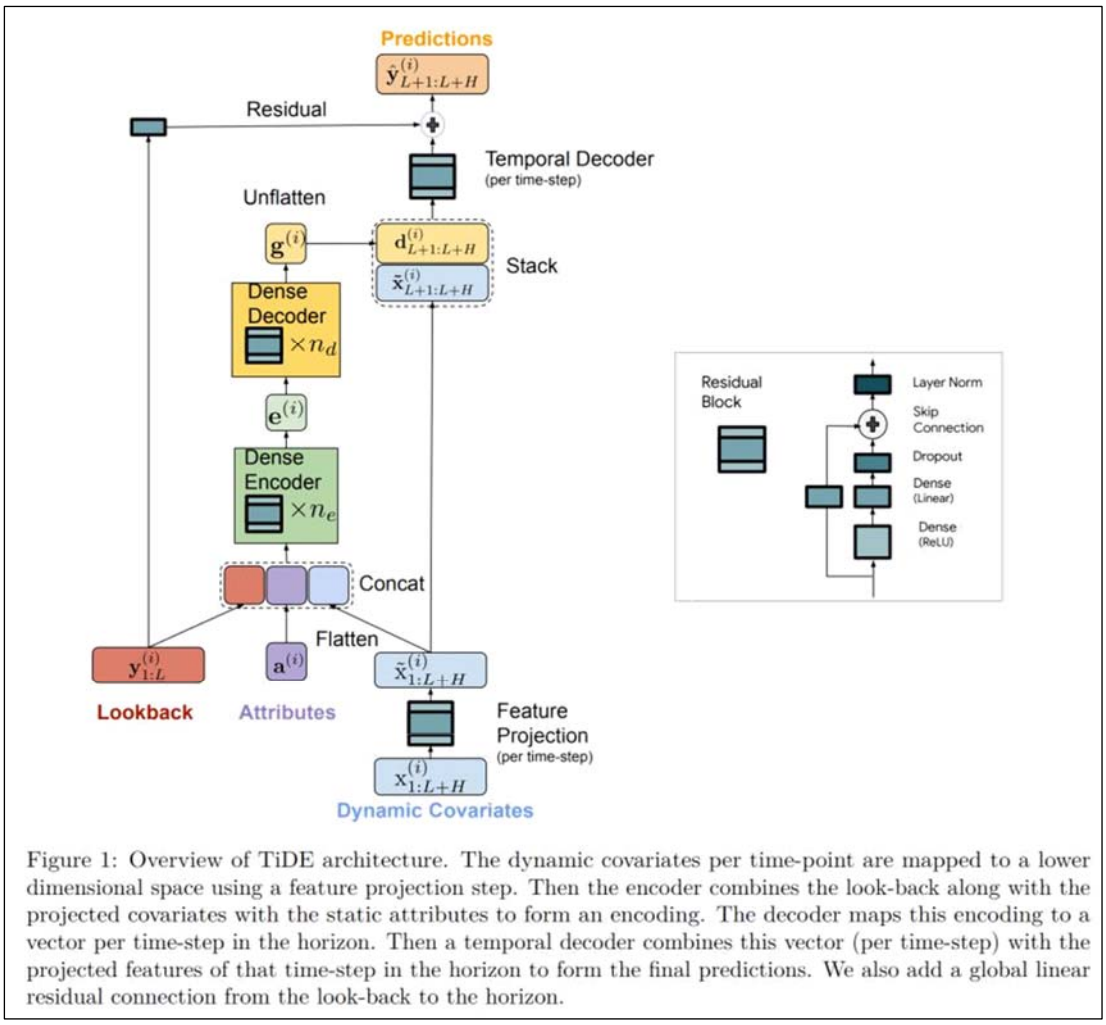

Figure 1: Overview of TiDE architecture. The dynamic covariates per time-point are mapped to a lower dimensional space using a feature projection step. Then the encoder combines the look-back along with the projected covariates with the static attributes to form an encoding. The decoder maps this encoding to a vector per time-step in the horizon. Then a temporal decoder combines this vector (per time-step) with the projected features of that time-step in the horizon to form the final predictions. We also add a global linear residual connection from the look-back to the horizon.

Figure 7. TiDE architecture from the original.

Source: Das et al. (2023).

Details of the "dedicated" training. TiDE (Time-series Dense Encoder) (Das et al., 2023) is a state-of-the-art model specifically designed for long-term forecasting, making it particularly suitable for our 48-hour forecast

window, which translates to 192 intervals of 15-minute data. Belonging to the decomposition family like N-BEATS (Oreshkin et al., 2020), TiDE excels in handling the complexity of long-term horizon predictions. It achieves this by incorporating both static and dynamic covariates and supports multioutput, joint forecasting for multiple time series.

The architecture of TiDE involves several key steps. Initially, dynamic covariates are mapped to a lower-dimensional space using residual blocks. These projections, combined with the look-back window and static attributes, are then embedded using an encoder with multiple residual blocks. This encoding captures the essential patterns and relationships in the data. The decoding phase transforms these hidden representations into future predictions through a dense decoder followed by a temporal decoder. The dense decoder stacks several residual blocks to produce an intermediate vector, which is reshaped and further refined by the temporal decoder, ensuring accurate forecasts by incorporating future covariates through a "highway" mechanism. (For more details see Figure 7) This sophisticated architecture allows TiDE to effectively handle the intricate demands of long-term forecasting in our dataset.

During training (via DARTS) our model configuration was as follows: The model architecture comprised 2 encoder and 2 decoder layers, with the latter's output dimension set to 128, indicating the size of the tensor produced by the decoder layers. A hidden size of 512 is selected for the hidden layers, enhancing the model's capacity to capture complex patterns in the data. Additionally, the model includes a temporal decoder hidden size of 128.

Normalization and regularization techniques turned out to be crucial to prevent overfitting, ensure generalization, but also to enhance numeric stability, since some outliers were making it difficult to get smooth convergence. Layer normalization is also enabled, providing stability to the learning process. A dropout rate of 0.1 is applied as a form of regularization, reducing the risk of overfitting by randomly omitting a subset of features at each iteration.

For the optimization process, we employ the Rectified Adam (RAdam) (Liu et al., 2021) optimizer, chosen for its efficiency and stability in training deep learning models (as well as preventing the possible loss of generalization power for the Adam (Kingma, & Ba, 2017) optimizer and other adaptive gradient methods). The loss function utilized was the Mean Absolute Error Loss, which is appropriate for our forecasting objectives and further enhanced convergence stability. The training process took 150 epochs with a batch size of 100 for further regularization.

The "foundational model": Lag-Llama. Lag-Llama (Rasul et al., 2024) is a foundation model designed for univariate probabilistic time series forecasting, utilizing a decoder-only transformer architecture. The model processes lagged features from prior time series values and integrates these with covariates. This approach leverages the temporal dependencies inherent in time series data. The architecture comprises multiple causally masked transformer decoder layers, each incorporating RMS Norm (Zhang, & Sennrich, 2019) for pre-normalization and Rotary Positional Encoding (RoPE) (Su et al., 2023) to handle temporal information effectively.

The pretraining phase of Lag-Llama by the original authors involved training on a large corpus of diverse time series datasets. This corpus included 7,965 different univariate time series, amounting to approximately 352 million data windows. The model size was scaled to capture extensive temporal patterns, enabling it to generalize across various domains. Pretraining from scratch allowed Lag-Llama to develop strong zero-shot generalization capabilities, demonstrating robust performance on unseen datasets without any prior finetuning.

One of the standout features of Lag-Llama is its ability to perform zero-shot pre-dictions. This capability is crucial for real-world applications where new, previously unseen time series data must be forecasted without retraining the model. In evaluations, Lag-Llama showed comparable or superior performance to state-of-the-art models specifically trained on those datasets.

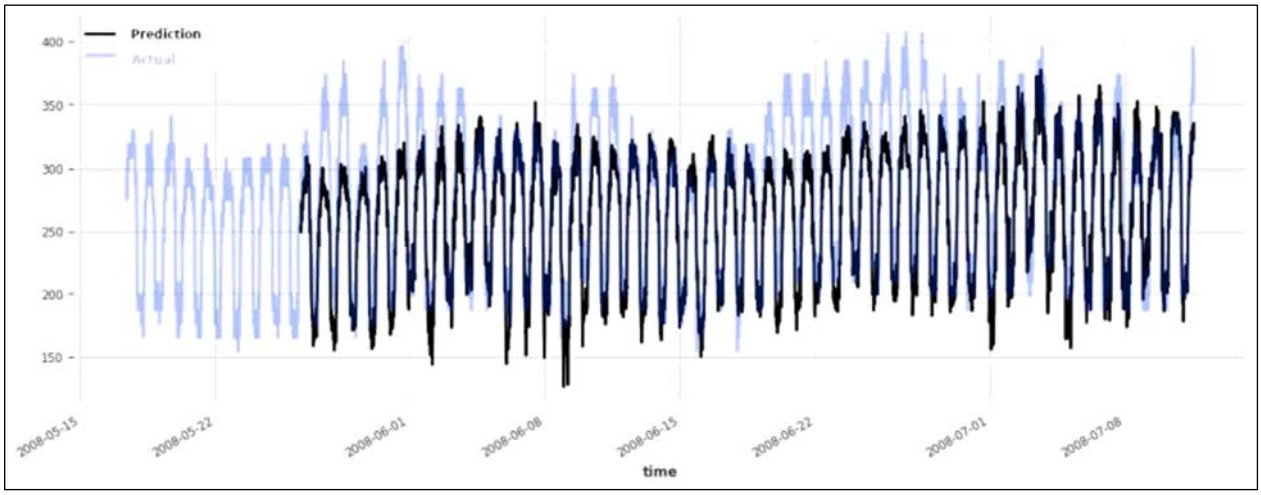

Figure 8. Linear regression results (on valid).

For further enhancement, Lag-Llama supports fine-tuning on specific datasets. Finetuning adjusts the pretrained model's parameters to the characteristics of the new data, significantly improving its forecasting accuracy. This dual capability of strong zero-shot performance and effective fine-tuning makes Lag-Llama a versatile tool in time series forecasting, suitable for both short-term and long-term predictions, including our 48-hour forecast window.

Please note, that Lag-Llama is a univariate model, so it is in a sense "handicapped", since it can not get as input the holiday or temperature data, though it has a built-in capability to create and handle time encoding features.

Results

Taking a look at the model results 1, and comparing the results of the models on the validation data, we can see the clear dominance of the TiDE model over the simple baselines, thus we can conclude that even with the extensive feature engineering we carried out, there are some notable non-linear patterns in the date that the TiDE model can capitalize on (See Figures 8 and 9).

Table 1

Comparison of Model Performance

Model	MAE	MAPE	R2
Auto-ARIMA	69.36	26.33	-1.22
Linear Regression	25.90	9.05	0.698
TiDE	22.81	8.17	0.768
Lag-Llama (Zero shot)	43.9	17.32	0.10
Lag-Llama (Finetuned)	13.25	4.81	0.89

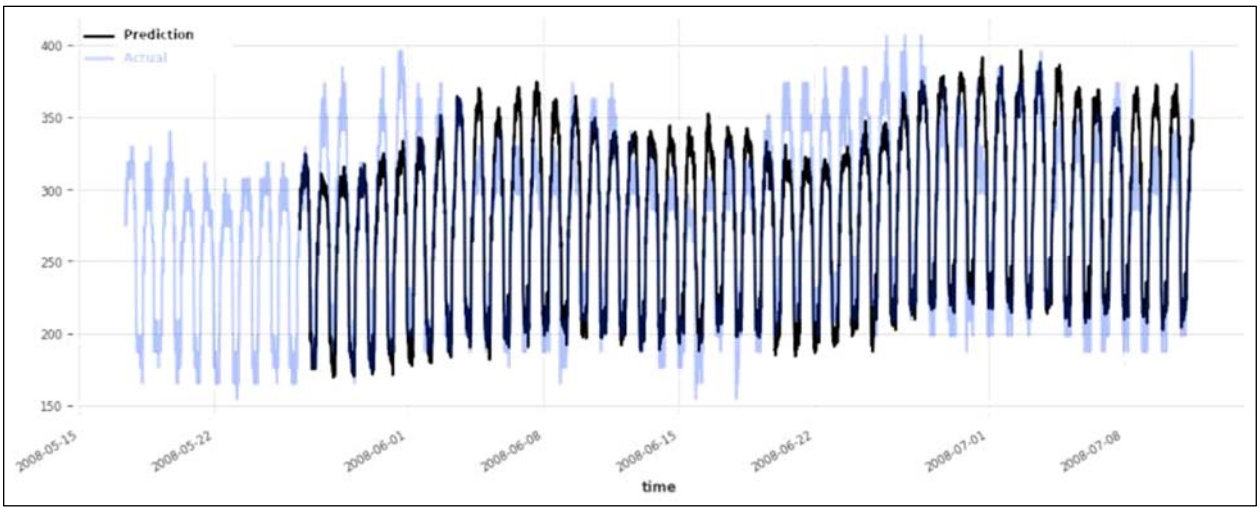

Figure 9. TiDE results (on valid).

None the less, visually inspecting the results can be informative, since we can observe, that even the Linear Regression model is capable of capturing complex time dependent relationships, though the TiDE model has a definitive dominance in being able to adapt to the ebb and flow of demand (which is the expected behavior from such a parameter rich model).

Regarding the foundational model: In the "zero shot prediction" case, where we just simply present a section of the validation data to the Lag-Llama model, it's performance is definitely inferior (see Figure 10), even to the simple linear model case. We can thus conclude, that the long-awaited era of just using "out of the box" time series models to predict without any training seems not to have arrived, at least to our practical retail energy usage use case.

But a more notable, and highly encouraging result can be gleaned from the performance of Lag-Llama after finetuning (see Figure 11). The performance of the finetuned model is remarkably good, and even with it's definitely lower parameter count (5 million vs. 30 million) definitely beats the "dedicated" TiDE model.

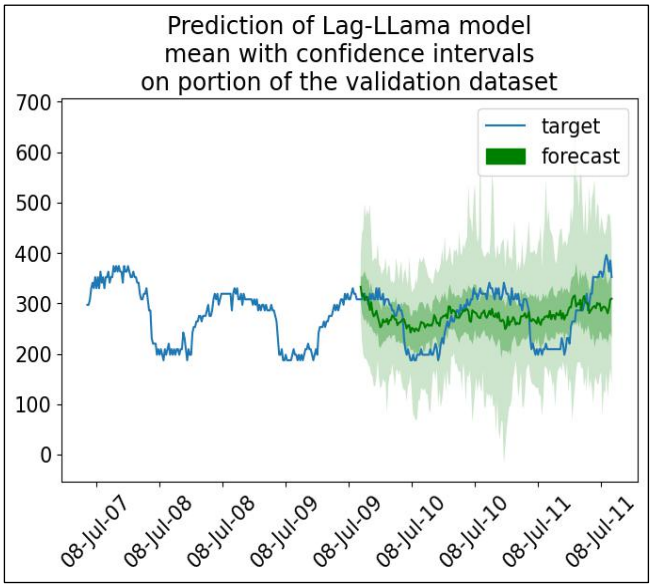

Figure 10. Lad-Llama zero-shot results (on valid).

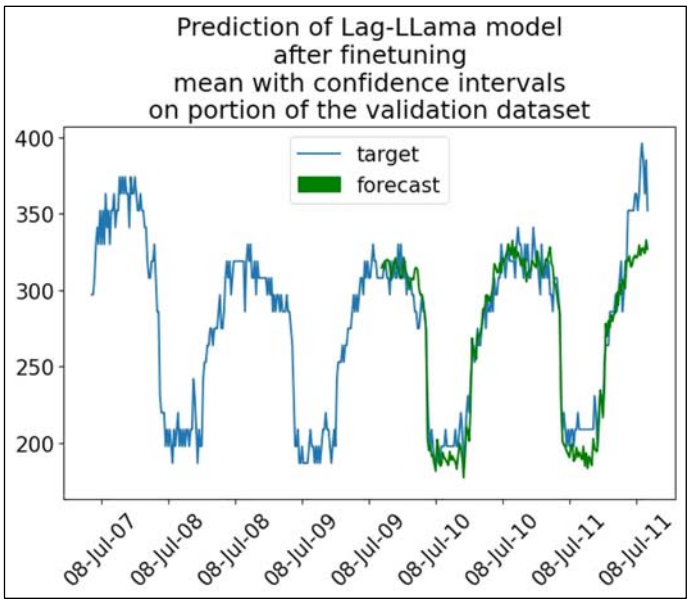

Figure 11. Lad-Llama finetuned results (on valid).

Discussion and Recommendations

Though a more extensive validation on different parts of the dataset, as well as on other data sources is still definitely needed, current results indicate, that the utilization of a pre-training and finetuning regime based on a foundational time series model is definitely competitive even with extensive feature engineering and dedicated deep learning models. This result is all the more encouraging, since in many cases single organizations lack the dataset size to efficiently train even dedicated deep learning models, let alone foundational ones, but they can absolutely be able to collect enough data to fruitfully apply finetuning of pre-trained and openly available models as we did in the context of the Hungarian retail company, thus deep learning, in it's "foundational" form can reasonably contribute to cost saving and mitigation of climate related challenges.

References

Ansari, A. F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S. S., Arango, S. P., Kapoor, S., Zschiegner, J., Maddix, D. C., Wang, H., Mahoney, M. W., Torkkola, K., Wilson, A. G., Bohlke-Schneider, M., & Wang, Y. (2024). Chronos: Learning the language of time series.

Biber, A., Felder, M., Wieland, C., & Spliethoff, H. (2022). Negative price spiral caused by renewables? electricity price prediction on the german market for 2030. *The Electricity Journal*, *35*(8), 107188. https://doi.org/https://doi.org/10.1016/j.tej.2022.107188

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N. S., Chen, A. S., Creel, K. A., Davis, J., Demszky, D., . . . Liang, P. (2021). On the opportunities and risks of foundation models. ArXiv. https://crfm.stanford.edu/assets/report.pdf

Box, G. E. P., & Jenkins, G. M. (1970). Time series analysis: Forecasting and control (1st). Holden-Day. contributors, W. (2024). Makridakis competitions [Accessed: 2024-05-16]. https://en. wikipedia.org/wiki/Makridakis Competitions

Das, A., Kong, W., Leach, A., Mathur, S., Sen, R., & Yu, R. (2023). Long-term forecasting with tide: Time-series dense encoder.

Das, A., Kong, W., Sen, R., & Zhou, Y. (2024). A decoder-only foundation model for time-series forecasting.

Developers, D. (2022a). Autoarima [Accessed: 2024-05-16].

Developers, D. (2022b). Time axes encoders [Accessed: 2024-05-16].

Dickey, D. A., & Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, *74*(366), 427-431. https://doi.org/10.1080/01621459.1979.10482531

Garcia, C., Yang, L., Zhang, M., Chen, Y., Patel, S., & Nguyen, T. (2024). Lag-Llama: Time-series foundation models [Accessed: 2024-05-16].

Garza, A., & Mergenthaler-Canseco, M. (2023). Timegpt-1.

Goswami, M., Szafer, K., Choudhry, A., Cai, Y., Li, S., & Dubrawski, A. (2024). Moment: A family of open time-series foundation models.

Granger, C. (1969). Investigating causal relations by econometric models and cross-spectral methods [Accessed: 2024-03-12]. *Econometrica*, *37*(3), 424-438. https: //doi.org/10.2307/1912791

Herzen, J., Lässig, F., Piazzetta, S. G., Neuer, T., Tafti, L., Raille, G., Pottelbergh, T. V., Pasieka, M., Skrodzki, A., Huguenin, N., Dumonal, M., Kościsz, J., Bader, D., Gusset, F., Benheddi, M., Williamson, C., Kosinski, M., Petrik, M., & Grosch, G. (2022). Darts: User-friendly modern machine learning for time series [Accessed: 2024-05-16]. *Journal of Machine Learning Research*, *23*(124), 1-6. https: //unit8co.github.io/darts/index.html; http: //jmlr.org/papers/v23/21-1177.html

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification.

Kingma, D. P., & Ba, J. (2017). Adam: A method for stochastic optimization.

Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., & Han, J. (2021). On the variance of the adaptive learning rate and beyond.

Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2020). N-beats: Neural basis expansion analysis for interpretable time series forecasting.

Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L. M., Rothchild, D., So, D., Texier, M., & Dean, J. (2021). Carbon emissions and large neural network training.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python [Accessed: 2024-05-16]. *Journal of machine learning research*, *12*(Oct), 2825-2830. https://scikit- learn.org/stable/modules/ generated/sklearn.linear model.LinearRegression.html.

Rasul, K., Ashok, A., Williams, A. R., Ghonia, H., Bhagwatkar, R., Khorasani, A., Bayazi, M. J. D., Adamopoulos, G., Riachi, R., Hassen, N., Bilos,˘ M., Garg, S., Schneider, A., Chapados, N., Drouin, A., Zantedeschi, V., Nevmyvaka, Y., Rish, I. (2024). Lag-Llama: Towards foundation models for probabilistic time series forecasting.

Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., & Liu, Y. (2023). Roformer: Enhanced transformer with rotary position embedding.

Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management Science*, *6*(3), 324-342. https://doi.org/10.1287/mnsc.6.3.324

Woo, G., Liu, C., Kumar, A., Xiong, C., Savarese, S., & Sahoo, D. (2024). Unified training of universal time series forecasting transformers.

Ye, J., Zhang, W., Yi, K., Yu, Y., Li, Z., Li, J., & Tsung, F. (2024). A survey of time series foundation models: Generalizing time series representation with large language model.

Zhang, B., & Sennrich, R. (2019). Root mean square layer normalization.