

Explainable Load Forecasting with Covariate-Informed Time Series Foundation Models

Matthias Hertel*
Karlsruhe Institute of Technology
Karlsruhe, Germany
matthias.hertel@kit.edu

Alexandra Nikoltchovska*
Karlsruhe Institute of Technology
Karlsruhe, Germany
alexandra.nikoltchovska@kit.edu

Sebastian Pütz
Karlsruhe Institute of Technology
Karlsruhe, Germany
sebastian.puetz@kit.edu

Benjamin Schäfer
Karlsruhe Institute of Technology
Karlsruhe, Germany
benjamin.schaefer@kit.edu

Ralf Mikut
Karlsruhe Institute of Technology
Karlsruhe, Germany
ralf.mikut@kit.edu

Veit Hagenmeyer
Karlsruhe Institute of Technology
Karlsruhe, Germany
veit.hagenmeyer@kit.edu

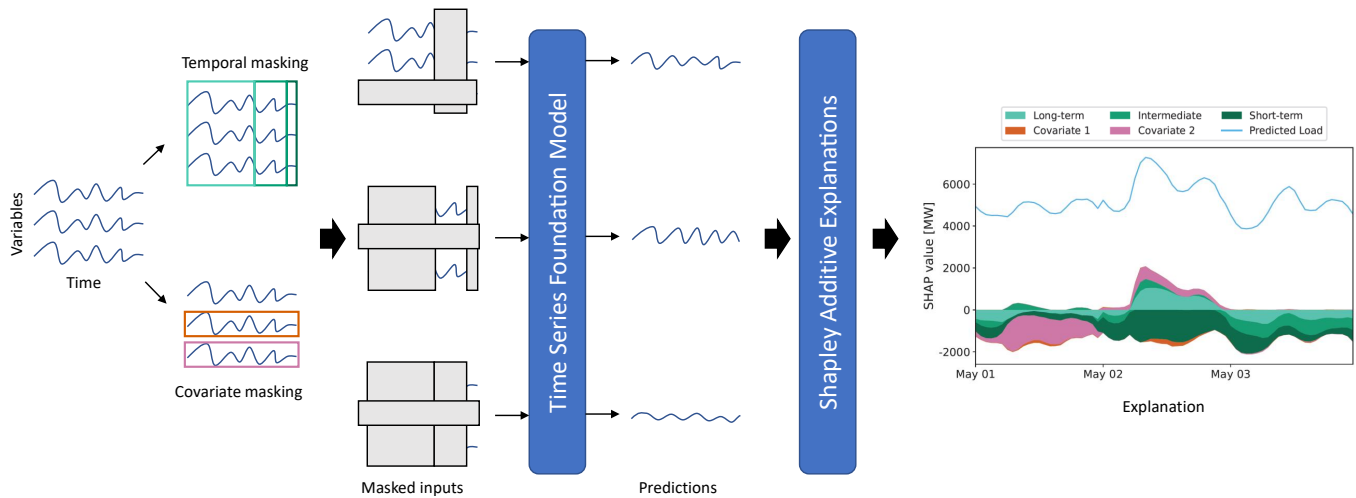


Figure 1: Overview of our approach to efficiently compute exact Shapley Additive Explanations (SHAP) for covariate-informed Time Series Foundation Models (TSFMs). We begin with historical hourly load observations accompanied by weather and calendar covariates. Our masking strategy operates along temporal and covariate dimensions – selectively removing historical time steps or entire covariates to create coalition samples. Each masked sample is then processed independently through Chronos-2 and TabPFN-TS to generate forecasts. Finally, SHAP values are computed by comparing predictions across coalitions, attributing the forecast to specific periods in the context and/or individual covariates.

Abstract

Time Series Foundation Models (TSFMs) have recently emerged as general-purpose forecasting models and show considerable potential for applications in energy systems. However, applications in critical infrastructure like power grids require transparency to ensure trust and reliability and cannot rely on pure black-box models. To enhance the transparency of TSFMs, we propose an efficient algorithm for computing Shapley Additive Explanations (SHAP) tailored to these models. The proposed approach leverages the flexibility of TSFMs with respect to input context length and provided

covariates. This property enables efficient temporal and covariate masking (selectively withholding inputs), allowing for a scalable explanation of model predictions using SHAP. We evaluate two TSFMs – Chronos-2 and TabPFN-TS – on a day-ahead load forecasting task for a transmission system operator (TSO). In a zero-shot setting, both models achieve predictive performance competitive with a Transformer model trained specifically on multiple years of TSO data. The explanations obtained through our proposed approach align with established domain knowledge, particularly as the TSFMs appropriately use weather and calendar information for load prediction. Overall, we demonstrate that TSFMs can serve as transparent and reliable tools for operational energy forecasting.

*Matthias Hertel and Alexandra Nikoltchovska contributed equally.



This work is licensed under a Creative Commons Attribution 4.0 International License.
E-Energy '26, Banff, AB, Canada
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2011-6/26/06
<https://doi.org/10.1145/3744255.3811724>

CCS Concepts

• Computing methodologies → Machine learning; • Applied computing → Forecasting.

Keywords

Load Forecasting, Time Series Foundation Models, Explainable Artificial Intelligence, Shapley Additive Explanations

ACM Reference Format:

Matthias Hertel, Alexandra Nikoltchovska, Sebastian Pütz, Benjamin Schäfer, Ralf Mikut, and Veit Hagenmeyer. 2026. Explainable Load Forecasting with Covariate-Informed Time Series Foundation Models. In *The 17th ACM International Conference on Future and Sustainable Energy Systems (E-Energy '26)*, June 22–25, 2026, Banff, AB, Canada. ACM, New York, NY, USA, 15 pages. <https://doi.org/10.1145/3744255.3811724>

1 Introduction

Accurate forecasting plays a central role in modern energy systems, as it is essential for balancing supply and demand, maintaining grid stability, and anticipating and mitigating grid congestions [16, 23]. Reliable forecasts of electrical load, renewable generation, and energy prices enable a wide range of operational and planning applications, including the scheduling of redispatch measures, demand-side management, storage dispatch, and the operation of energy management systems. In recent years, forecasting performance has been substantially improved through the use of deep learning models. While these models achieve good prediction accuracy, they lack transparency, which limits their acceptance in safety-critical and operational settings [36]. This challenge has become increasingly urgent with the introduction of regulatory frameworks such as the EU AI Act and guidelines for AI deployment in critical infrastructure, which emphasize the need for transparency and accountability in high-stakes applications [13, 48].

Time Series Foundation Models (TSFMs) [30] have recently emerged as a new class of large-scale deep learning models, built predominantly on Transformer architectures and pretrained on large and diverse collections of time series data. Unlike conventional deep learning approaches for forecasting, which are typically trained from scratch on task-specific datasets, TSFMs leverage large-scale pretraining to achieve strong zero-shot and few-shot generalization across tasks and domains. More recent architectures, such as Chronos-2 [2] and TabPFN-TS [25], extend this paradigm by supporting the integration of covariates — additional time-varying features that influence the target (sometimes also referred to as exogenous features or auxiliary information) — which is particularly relevant for energy system applications.

Like previous work [2, 25], we define models as *univariate* when they forecast a single time series, whereas *multivariate* models forecast multiple time series together. Models that incorporate covariates in addition to past values from the target time series are called *covariate-informed*.

Despite their promising performance, covariate-informed TSFMs remain largely opaque, and their increasing complexity exacerbates the challenge of understanding and trusting their predictions. Developing and applying suitable Explainable AI (XAI) methods is therefore a key requirement for the deployment of TSFMs in operational energy system contexts. Post-hoc explainability methods such as Shapley Additive Explanations (SHAP) [35] offer theoretically grounded insights into model behavior, but their computational cost poses a major barrier when applied to large-scale time series models with long input horizons and multiple covariates.

To address this gap, we introduce an efficient SHAP-based explainability algorithm tailored to TSFMs and demonstrate its applicability to load forecasting, a key challenge in the energy domain [28]. We apply our approach to two architecturally distinct TSFMs — Chronos-2 [2] and TabPFN-TS [25] — using operational electrical load data from Baden-Württemberg, Germany. The overall procedure of our approach is illustrated in Figure 1. We make the following key contributions:

- (1) We compare two covariate-informed TSFMs — Chronos-2 and TabPFN-TS — for electrical load forecasting on recent operational data from a specific TSO region, and benchmark their performance against state-of-the-art models trained from scratch.
- (2) We propose an efficient SHAP algorithm for TSFMs, using temporal and covariate masking to enable the estimation of SHAP values without the need for sampling from background data.
- (3) We empirically demonstrate that Chronos-2 and TabPFN-TS make meaningful use of covariates, such as calendar effects and weather variables, and that the resulting explanations are consistent with established domain knowledge.

The remainder of this paper is structured as follows: Section 2 reviews related work on load forecasting, TSFMs, and post-hoc explainability methods. The proposed approach for efficiently computing SHAP-based explanations is introduced in Section 3, followed by a description of the experimental setup in Section 4, including the dataset, preprocessing steps, TSFM configurations and evaluation metrics. Section 5 assesses the performance of TSFMs on recent operational load data. Section 6 presents the explainability analysis, examining TSFM behavior through global feature importance patterns and local explanations of individual predictions. Finally, Section 7 draws conclusions and discusses future research directions.

2 Related work

Machine Learning for Load Forecasting. While traditional load forecasting methods relied primarily on statistical approaches [23], in recent years, there has been a shift towards more complex deep learning architectures [14, 16, 24, 45]. Recent work showed that global models — trained on multiple time series, such as load profiles from multiple buildings or substations — can capture shared patterns across different contexts and achieve superior generalization compared to local models trained on individual time series [15, 18, 19, 39]. In addition, the effectiveness of load forecasting models improves when covariates are incorporated alongside historical load observations. Variables such as temperature, solar irradiation, and calendar information have been shown to improve prediction accuracy by capturing the external factors that drive consumption patterns [16]. It is particularly important to also include covariates for future time steps, which can be either known in advance, such as holidays and calendar information, or can be forecasts themselves, such as weather predictions.

Time Series Foundation Models (TSFMs). TSFMs take the global modeling paradigm further by pretraining on large and diverse datasets that often include synthetic data, enabling them to identify patterns across domains when applied to specific forecasting

tasks [32]. Initial TSFM architectures focused primarily on univariate forecasting, learning temporal patterns from historical values of the target series alone. Chronos [3] introduced the application of language model tokenization to time series, treating forecasting as a sequence-to-sequence task based on quantized observations. Lag-Llama [43] employs a decoder-only transformer architecture pretrained on large-scale univariate time series. TimesFM [10] uses a similar decoder-only design, emphasizing patching mechanisms to capture multi-scale temporal patterns. However, as with load forecasting, many real-world forecasting tasks require the integration of covariates to achieve accurate predictions. Recognizing this need, recent TSFM architectures have been extended to handle covariates. Chronos-2 [2] extends the original Chronos framework to explicitly incorporate covariates during both pretraining and inference while maintaining strong zero-shot capabilities. TabPFN-TS [25] takes a different approach by extending tabular foundation models to time series forecasting, using in-context learning to adapt to new tasks without gradient-based fine-tuning. While Moirai [50] also claims to be able to handle covariates, recent evaluations suggest it does not effectively leverage future covariates in load forecasting applications [31].

We select Chronos-2 and TabPFN-TS, representing state-of-the-art Transformer-based and tabular foundation model paradigms, respectively, as both have shown impressive results on established forecasting benchmarks, specifically on covariate-informed tasks [1, 47]. This comparison allows us to examine whether feature importance patterns are consistent across different modeling paradigms, with extension to further TSFMs left for future work.

Despite their capability to generalize to unseen data, TSFMs have primarily been developed and evaluated on standard benchmark datasets. Recent work by Meyer et al. [37] raises concerns about this paradigm, suggesting that TSFM performance can substantially drop on datasets that differ from those used in the original publications. Initial applications of TSFMs to energy load forecasting have started filling this gap and have demonstrated their potential in this domain. Meyer et al. [38] and Lin et al. [33] showed competitive performance of univariate TSFMs on short-term load forecasting benchmarks, while Kreusel et al. [31] extended these evaluations to covariate-informed settings, highlighting the promise of covariate-informed TSFMs for energy applications. Recent findings question whether TSFMs can deliver on their "one-size-fits-all" promise, showing that zero-shot performance depends heavily on the alignment between pretraining and target data, and that lightweight specialized models can outperform fine-tuned TSFMs under domain shift or concept drift [27]. Given that load forecasting involves region-specific weather dependencies and evolving consumption patterns, continued evaluation of covariate-informed TSFMs on recent operational data from specific TSO regions remains valuable. Therefore, for our first contribution we evaluate Chronos-2 and TabPFN-TS on electrical load data from the German TSO TransnetBW.

Post-hoc Explainability for Time Series Models. While the evaluation of TSFMs on domain-specific data addresses questions of predictive performance, their deployment in critical infrastructure, such as power grids, raises an equally important concern: model transparency. TSFMs, similar to deep learning methods, operate as

black boxes whose learning and decision processes remain hidden from energy domain experts such as system operators. Operational energy forecasting requires not only accurate forecasts, but also validation that models appropriately respond to known physical relationships, such as temperature-load dependence and calendar-driven consumption patterns. Post-hoc explainability techniques have therefore become essential tools for interpreting complex forecasting models without sacrificing their predictive power, enabling domain experts to inspect model behavior and build the operational trust necessary for deployment [4, 36].

Among post-hoc methods, SHAP [35] has been widely adopted due to its theoretical foundations in cooperative game theory and its model-agnostic property. SHAP is a unified framework for explaining model predictions, based on Shapley values from game theory [46], used to fairly distribute the total payoff of a cooperative game between players, when their individual contributions vary and they interact with one another. These values satisfy fairness axioms that make them particularly meaningful for post-hoc interpretability. For example, efficiency ensures that attributions sum to the total prediction shift, and symmetry ensures that features with identical contributions receive identical attributions. In machine learning, features act as players, with the payoff being the difference between prediction and the base value (expected value or the average prediction of the model across the dataset).

Despite these desirable theoretical properties, SHAP is computationally expensive in practice. The method requires many model evaluations based on feature subsets, called coalitions. As most models cannot be evaluated with subsets of the features provided during training, a common approach is to sample absent features multiple times from a background dataset. This results in the need for many model evaluations, which is especially costly for large models such as TSFMs.

To address these computational limitations, several methods improve the efficiency of SHAP for time series. TimeSHAP [5] groups consecutive time steps to reduce the number of coalitions, while WindowSHAP [40] uses sliding windows for temporal localization. ShapTime [51] leverages temporal structure for acceleration, and SHAPformer [21] combines Transformers with temporal grouping. Further speedups are achieved by avoiding sampling, either replacing absent features with a base value [5] or restricting the models access to absent values with attention manipulation [11, 21]. However, these methods either still rely on background sampling or are tied to specific architectures, making them computationally infeasible for general-purpose TSFMs. Our work adopts the temporal grouping idea from TimeSHAP and WindowSHAP, while eliminating background sampling by exploiting the flexibility of TSFMs to handle arbitrary input contexts.

Most TSFMs are based on the Transformer architecture [49], which allows to visualize the attention weights to highlight important inputs as an explanation [20]. However, attention is only a proxy for feature importance [6, 26] and highlights inputs without revealing their directional influence on predictions. Therefore, we use SHAP, which provides directional, quantitative feature attributions.

The intersection of explainability and TSFMs remains largely unexplored. Pandey et al. [42] examined internal representations of TSFMs through synthetic scenarios, but focused on model behavior

rather than real-world applications. Boileau et al. [7] developed inherently interpretable TSFM architectures, which represent a different design philosophy than post-hoc explanation of existing high-performance models. Rundel et al. [44] introduced a method for explaining TabPFN predictions, but this work focuses on static tabular tasks and does not address the temporal dimension inherent in time series forecasting.

Building on these foundations, we address the need for efficient XAI methods tailored to covariate-informed TSFMs, where both temporal dependencies and the effects of covariates must be considered. This motivates our second contribution: leveraging the internal structure of TSFMs for an efficient SHAP adaptation using temporal and covariate masking that eliminates the need for background data sampling. Our third contribution empirically demonstrates that covariate-informed TSFMs meaningfully utilize covariates like calendar effects and weather variables, with explanations that are consistent with established domain knowledge.

3 Methodology

We develop efficient SHAP-based explanations by leveraging the internal structures of Chronos-2 and TabPFN-TS to implement architecture-specific masking strategies, eliminating the need for background data sampling. The methodology is depicted in Figure 1. To establish the foundation for our approach, we first briefly revisit the fundamentals of SHAP. Then, we describe how the TSFMs process input data, and finally outline our feature grouping and masking strategies. Concise summaries of the model architectures are provided in Appendix A, with full technical details in the original publications [2, 25].

SHAP. For a model f trained on a feature set $V = \{v_1, v_2, \dots, v_n\}$ consisting of n features, the SHAP value for feature v_i is calculated as:

$$SHAP(v_i) = \sum_{S \subseteq V \setminus \{v_i\}} \frac{(n-1-|S|)! \cdot |S|!}{n!} \cdot (f(S \cup \{v_i\}) - f(S)), \quad (1)$$

where S represents all possible subsets (coalitions) of features excluding v_i , and $f(S \cup \{v_i\}) - f(S)$ quantifies the marginal contribution of v_i to coalition S .

Computing exact SHAP values requires evaluating all 2^n possible feature coalitions, resulting in exponential computational complexity of $O(2^n)$. Since this becomes intractable even for a moderate number of features, in practice, SHAP values are typically estimated using sampling-based approximations. However, these approximations introduce variance and can require thousands of model evaluations to achieve stable estimates, making them computationally expensive for time series forecasting applications.

An additional challenge is that classical machine learning models require all features they have been trained on for inference. Evaluating $f(S)$ for a subset of features $S \subset V$ then requires handling the missing feature values. In practice, this is usually done by computing the marginal contribution of a feature by marginalizing over the missing features, either by sampling from a background dataset or by integrating over their empirical distribution.

TSFM inputs. In contrast to classical machine learning models, TSFMs are flexible in the context length and covariates provided to the model. This allows to evaluate them on subsets of the data

without sampling absent features from background data. A TSFM evaluated with context length C and forecast horizon H receives the past C values of the target time series as input (denoted as $X \in \mathbb{R}^C$), together with an arbitrary number of covariates Z_i . These covariates can either be available for the past only ($Z_i \in \mathbb{R}^C$), or extend into the forecast horizon ($Z_i \in \mathbb{R}^{C+H}$) when they are known in advance, such as calendar features or weather forecasts.

Feature grouping. Computing exact SHAP values requires to evaluate the TSFM with 2^n feature subsets for n features. As this is not feasible for large amounts of features, we instead group the features into N subsets, with $N \ll n$, and evaluate the TSFM only 2^N times. We incorporate two grouping mechanisms: *Temporal grouping* splits the past load time series into consecutive patches, in order to analyze the effect of the load in different past periods. In particular, we introduce one group for the last day (hours -24 to -1 relative to the time the prediction is made), short-term (hours -168 to -25), intermediate (hours -672 to -169) and long-term (hours < -672) periods. *Covariate grouping* builds one group for each covariate, which is used to analyze the effect of the covariate on the prediction. In total, we analyze four temporal groups and $|Z|$ covariate groups.

Masking for Chronos-2. To evaluate a given feature subset, the groups not belonging to it are *masked*, i.e., withheld from the model input. *Temporal masking* works in two ways. To mask the first k values of the context window, the context length gets reduced to $C-k$, so that the masked values are not contained in the context. To mask intermediate or the most recent values, the load values are set to not-a-number (NaN), which indicates missing values. Internally, Chronos-2 sets missing values to zero and uses a special token to distinguish between missing values and measured zeros. *Covariate masking* works by removing masked covariates entirely from the model input. When all data is masked so that the model input is empty, a base prediction is made. For this, we use the mean load from the previous 48 weeks, which corresponds to the model's maximum input length.

Masking for TabPFN-TS. To mask the first k values of the context window, *temporal masking* for TabPFN-TS works the same way as for Chronos-2. To mask intermediate or the most recent values, corresponding rows are dropped from the model input entirely, due to the inability of TabPFN-TS to handle missing values in the target variable. *Covariate masking* removes masked covariates entirely from the model input, equivalent to the Chronos-2 approach. Similar to Chronos-2, we use the mean load from the previous year as a base prediction when all data is masked.

4 Experimental Setup

Dataset. The dataset comprises the electrical load of the German TSO TransnetBW, downloaded from the ENTSO-E Transparency Platform [12] for the period January 2015 – September 2025, and converted into hourly resolution. The dataset is enriched with weather data from the ERA5 reanalysis model [17] for the DE1 region comprising Baden-Württemberg, which has a typical peak load of roughly 14 GW [29]. From the weather dataset, the solar irradiance and the outside temperature are used as covariates. In practice, the reanalysis data is not available at prediction time, but

it is a common approach to use weather data as perfect forecasts in the absence of historical weather forecasts [16]. A third covariate is used to encode holidays, set to 1 for Sundays and public holidays, and to 0 otherwise. Sundays and holidays are grouped into a single indicator because both exhibit similar reduced-demand patterns, and treating them as separate categories would result in a sparse feature with too few occurrences for reliable generalization [52].

The last year of data (October 2024 – September 2025) is used as test data for the experiments, the year before (October 2023 – September 2024) as validation data and everything before (January 2015 – September 2023) as training data to train models from scratch.

Forecasting task & metrics. To evaluate forecast performance, a forecast is made at every hour in the test set with a horizon of 24 hours. We evaluate the models as point forecasters, taking the median of the predicted distribution as the forecast. Our focus is on evaluating and explaining point predictions, as these are the primary outputs used in operational forecasting decisions. Extending the analysis to probabilistic forecasts is left for future work. We use three different metrics to evaluate the forecast accuracy: mean absolute error (MAE), root-mean squared error (RMSE), and mean absolute percentage error (MAPE), see Appendix B.1 for more details. To explain models, we generate explanations of the forecasts made at midnight, so that each hour in the test set is predicted exactly once and we do not get multiple explanations for the same hour. Note that the concatenation of the explained forecasts comprises the entire year without overlaps.

Chronos-2. We evaluate Chronos-2 with the default hyperparameters in a zero-shot setting. Chronos-2 is a probabilistic model returning quantile predictions, which we turn into point forecasts by taking the median of the predicted distribution. We test different context lengths ranging from one week to the model’s maximum of 8192 hours, and find that longer context lengths improve the forecast accuracy (see Appendix B.2 for the detailed results). We report forecasting results for a univariate Chronos-2 and for Chronos-2 with the three covariates described above, in order to quantify their contribution.

TabPFN-TS. We evaluate TabPFN-TS following the steps proposed in the original publication [25], which include generating a running index, calendar features (day of week, month, hour encoded as cyclic sine/cosine pairs), and automatic seasonal features that identify domain-specific periodicities beyond standard calendar cycles. We construct an ensemble of two models: one trained on normalized targets and one trained on power transformed targets (Box-Cox transformation [8]). Similar to Chronos-2, we test different context lengths (see Appendix B.2) and report results for both univariate TabPFN-TS and covariate-informed TabPFN-TS to quantify the improvement gained with covariates.

Non-TSFM models. We compare the TSFMs to a simple baseline and to Transformer models trained from scratch on the TSO data. The baseline predicts the observed load from the last day of the same type, distinguishing between workday, Saturday and Sunday/holiday as the types. The *Transformer* architecture is inspired by Hertel et al. [20]. It is an encoder-decoder model with two layers each and a hidden dimension of 128 and four attention

heads, trained from scratch with two weeks context length using the AdamW optimizer with a batch size of 128, a learning rate of 0.0001 and early stopping with ten epochs patience. We report results for two Transformer variants with different amounts of training data, namely one year (October 2022 – September 2023) and the full training data comprising 8.75 years (January 2015 – September 2023).

5 Benchmarking TSFMs on Recent Operational Load Data

In this section, we analyze the forecast accuracy and the model runtimes to evaluate the applicability of TSFMs to load forecasting.

Forecast accuracy. Table 1 reports the forecasting results on the test set. All models outperform the baseline across all error metrics. The Transformer trained on the full data achieves the lowest errors, while the same architecture trained on one year of data shows substantially higher errors. Both TSFMs – TabPFN-TS and Chronos-2 – achieve lower errors than the Transformer trained on one year of data. Compared to the one-year Transformer, TabPFN-TS reduces the MAE by 23.5 % and Chronos-2 by 26.8 %. When trained on the full data, the Transformer is better than the TSFMs, but only by a small margin. Between the two TSFMs, Chronos-2 achieves slightly lower errors than TabPFN-TS across all metrics in the covariate-informed setting. For both TSFMs, incorporating covariates leads to better accuracy than the univariate setting, highlighting the importance of incorporating exogenous information. The MAE improvement by using covariates is 27.0 % for Chronos-2 and 31.5 % for TabPFN-TS. Note that all models benefit from ERA5 reanalysis weather inputs, which represent historically reconstructed rather than forecasted conditions, so reported results represent an upper bound on operational accuracy.

Both models benefit from longer contexts, and Chronos-2 outperforms TabPFN-TS for all tested context lengths. Notably, TabPFN-TS needs at least two weeks of context to beat the baseline, whereas Chronos-2 outperforms the baseline already with one week of context. Potentially, this is due to the fact that TabPFN is pretrained as a tabular model, so it needs more context to infer the daily and weekly pattern inherent in the load time series, whereas Chronos-2 is pretrained on time series with various seasonalities. Results for the TSFMs with different context lengths are reported in Appendix B.2.

Despite the strong overall forecasting performance of both models, they encounter difficulties on specific days in proximity to public holidays, including October 4, December 23, January 7, June 20, and June 23. This suggests potential for improvement through feature engineering, such as incorporating covariates for long week-ends, school holidays, or other periods of reduced economic activity. For the complete year of forecasts made daily at midnight, see Figure 7 for Chronos-2 and Figure 8 for TabPFN-TS in Appendix B.4, each evaluated with the best model configuration.

Overall, the results highlight the importance of integrating covariates into TSFMs for accurate load forecasting. TSFMs are especially useful in scenarios with little training data, as they outperform a Transformer trained from scratch on one year of data. When lots of data is available, models trained from scratch can achieve better results than zero-shot TSFMs, but the performance difference is small and the TSFMs have the advantage that they are easy to use

without requiring any training, which makes them appealing in practice.

Table 1: Forecasting results on the test set, with the best result per metric highlighted in bold and second best in italics. The TSFMs are evaluated in both univariate (univ.) and covariate-informed settings.

Model	Training data	MAE [MW]	RMSE [MW]	MAPE [%]
Baseline	—	325.9	451.2	5.24
Transformer	1 year	213.4	318.9	3.32
Transformer	8.75 years	150.2	206.8	2.34
TabPFN-TS (univ.)	—	238.1	398.8	3.74
TabPFN-TS	—	163.2	250.5	2.55
Chronos-2 (univ.)	—	214.0	356.9	3.40
Chronos-2	—	<i>156.2</i>	<i>229.1</i>	<i>2.44</i>

Runtimes. Table 2 reports the runtimes for model training, inference and explanation. Training the Transformer from scratch on the full training dataset takes less than 20 minutes. This computational cost is not needed with TSFMs, which are pretrained on large diverse datasets and applicable zero-shot to new data without the need for further training. Chronos-2 has more parameters than the Transformer trained from scratch, and therefore needs more time to generate a prediction. However, both models are very efficient and allow to predict in a few milliseconds. TabPFN-TS, on the other hand, is much slower, needing multiple seconds to create a prediction. To generate an explanation, the TSFMs are run with all $2^7 = 128$ combinations of the seven temporal and covariate masks. Explaining a forecast takes less than $128 \times$ the prediction time, because masking reduces the context length and the number of covariates in the model input.

Table 2: Training runtime, runtime to generate a single prediction, and runtime to generate a single explanation, with the best result highlighted in bold.

Model	Training data	Training [min]	Prediction [ms]	Explanation [s]
Transformer	8.75 years	19.2	6.8	—
TabPFN-TS	—	—	5521.2	366.84
Chronos-2	—	—	65.6	5.12

6 Explaining TSFMs for Load Forecasting

Having established that TSFMs show competitive performance, we next investigate the SHAP-based global and local explanations of the two TSFMs to understand how the provided data influences the predictions.

6.1 Global explanations

Feature importance. Table 3 presents the feature importance for the two TSFMs, calculated as the percentage of the absolute SHAP values over the test set. The past load is by far the most important feature, with 89 % total importance for Chronos-2 and 87 % for TabPFN-TS. For Chronos-2, the four temporal windows have similar importance, despite their different lengths. This indicates that recent information is important for the prediction, as the last day alone is almost as important as the whole long-term context. TabPFN-TS also emphasizes the last day, but gives more relevance to the long-term context. Potentially, this can be explained by the fact that TabPFN-TS is built on TabPFN, which is not a time series model per se, and therefore requires more context to infer time series patterns. Regarding the covariates, the two TSFMs agree on the order of importance, with holiday being the most important, followed by temperature, and irradiance being the least important. TabPFN-TS gives more importance to the holiday covariate than Chronos-2, whereas the importance of the other two covariates is similar for both TSFMs.

Table 3: Feature importance on the test set.

Feature Group	Chronos-2	TabPFN-TS
Past load		
Long-term (weeks ≤ -5)	23.64 %	29.15 %
Intermediate (weeks -4 to -2)	23.39 %	18.67 %
Short-term (days -7 to -2)	19.88 %	18.63 %
Last day (day -1)	22.30 %	20.60 %
Covariates		
Holiday	4.51 %	6.46 %
Temperature	3.55 %	3.79 %
Irradiance	2.74 %	2.70 %

Covariate dependence. In order to analyze how the TSFMs use covariates, Figure 2 shows the SHAP values of Chronos-2 and TabPFN-TS in dependence of the three covariates. The x-axis shows the feature values at the predicted hour (or the difference to the value 24 hours before) and the y-axis shows the corresponding SHAP values. A positive SHAP value indicates that the feature increased the prediction, and a negative SHAP value indicates a decreased prediction. The dot color is used to show an interacting variable.

- *Holiday dependence:* The SHAP values are shown for six categories of days: Sunday, Monday, a workday following a workday ($\circ/\circ$), a workday following a holiday ($\bullet/\circ$), a holiday following a workday ($\circ/\bullet$), and a holiday following a holiday ($\bullet/\bullet$). When the previous day was a holiday and the next day is no holiday ($\bullet/\circ$), the holiday feature increases the predicted load. When the next day is a holiday ($\circ/\bullet$ and $\bullet/\bullet$), it decreases the load instead. The effect is stronger during the day (dark dots) than during the night (bright dots), where the base load is similar on all weekdays. This aligns well with our expectation that the load is reduced on holidays due to less industrial and economic activity, and jumps back to a normal level after the holiday. Chronos-2 is only slightly affected by the holiday feature on Sundays and

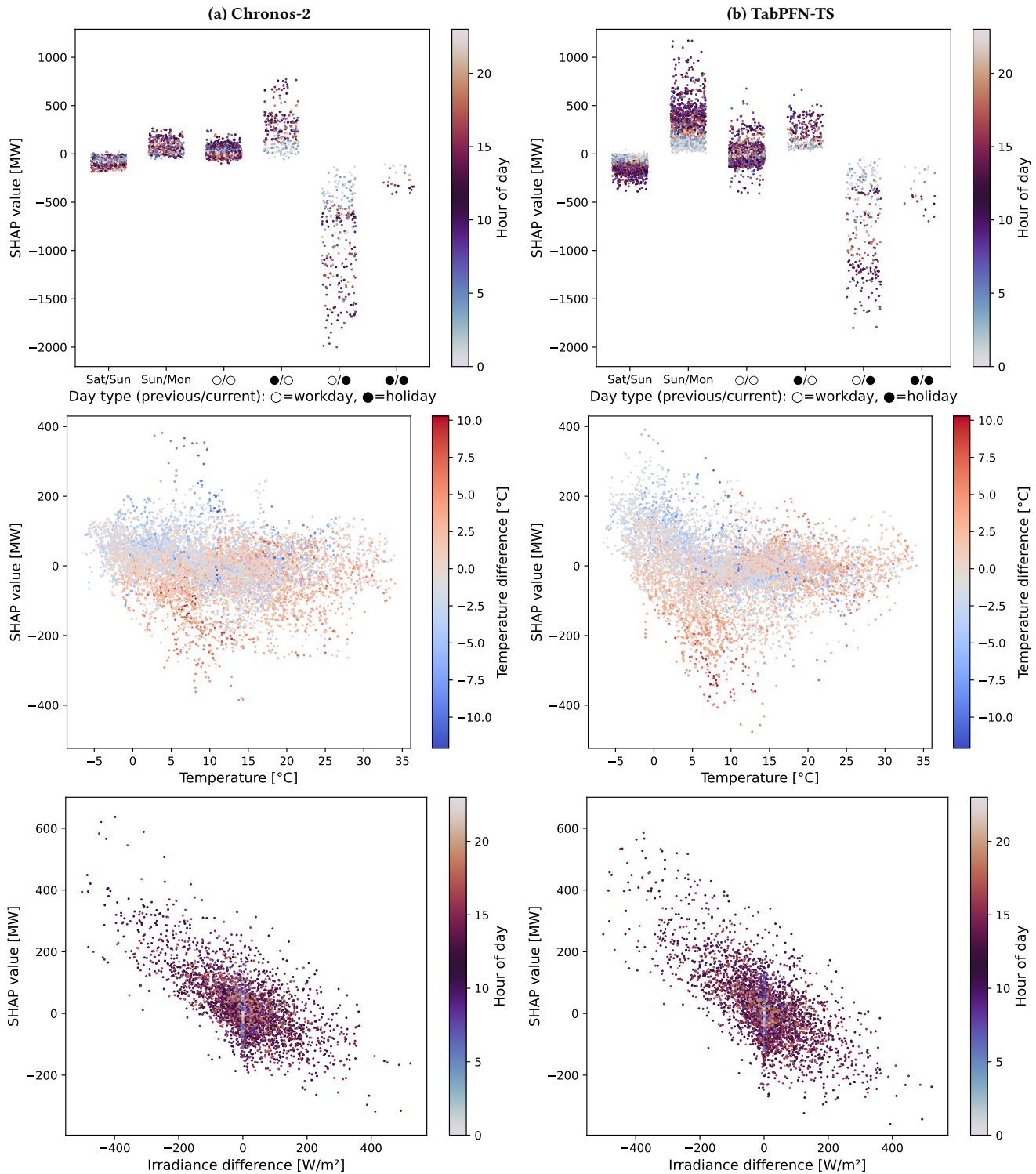


Figure 2: Dependence of Chronos-2 (left) and TabPFN-TS (right) on the holiday covariate (top), the temperature covariate (middle) and the irradiance difference to the previous day (bottom). Both TSFMs make similar use of the covariates and this use is consistent with domain knowledge. The predicted load is decreased on holidays (top), increased when the outside temperature drops on cold days (middle), and increased when PV generation drops due to lower irradiance (bottom).

Mondays, whereas TabPFN-TS shows a pronounced influence of the holiday feature on these days. This signifies that TabPFN-TS uses the holiday feature to predict a lower load on Sundays (where the feature is set to 1) and a higher load again on Mondays, whereas Chronos-2 infers the weekly pattern from the past load.

- *Temperature dependence*: For temperatures below 15 °C, both models react to temperature differences with respect to the previous day. When the temperature increases with respect to the temperature 24 hours ago (red dots), the predicted load decreases, resulting in negative SHAP values. When the temperature decreases (blue dots), the opposite effect is observed, resulting in positive SHAP values. This can be plausibly explained by the increased use of electric heating on cold days. At warmer temperatures, the magnitude of the SHAP values is smaller and the effect of the temperature difference is less pronounced. This temperature dependence is further illustrated by visualizing SHAP values against both current and previous day temperatures, confirming that both TSFMs rely more heavily on this covariate during periods of pronounced change, as shown in Figure 5 of Appendix B.3.
- *Irradiance dependence*: The lower panels show the SHAP values depending on the difference in the irradiance from the day before. Unsurprisingly, the difference is zero at night (bright dots), because the sun does not shine. During the day (dark dots), an almost linear pattern is visible for both models, with the slope directly quantifying how much the predicted load changes per unit change in irradiance. The models use the irradiance feature to react to changing weather conditions – when the irradiance decreases (negative Δ irradiance), the predicted load increases (positive SHAP value), and vice versa. This can be explained by the fact that the TSO load is affected by behind-the-meter photovoltaic, so that a higher irradiance decreases the effective load.

Overall, the covariate dependence of the TSFMs aligns well with our expectations derived from domain knowledge about influences of holidays, temperature and irradiance on the TSO load, indicating that the TSFMs make meaningful use of the covariates.

6.2 Local explanations

Next, we examine the local explanations for both past load windows and covariates to give further insights into when particular covariates are used by the TSFMs. We highlight four representative examples for the Chronos-2 model in Figure 3 that illustrate the models' behavior across different scenarios. The complete set of local explanations for both models for the entire test period is provided in Appendix B.4 (Figure 7 for Chronos-2 and Figure 8 for TabPFN-TS). Since Chronos-2 and TabPFN-TS exhibit similar patterns, Figure 3 shows the effects only based on Chronos-2 (see Figure 6 in the appendix for the same examples with TabPFN-TS):

- *Post-holiday effects* (Figure 3 (a)): On January 6, a public holiday in Baden-Württemberg, the holiday feature shows a negative SHAP value, indicating that the model appropriately reduces the predicted load. The following day (January 7) marks the first workday after an extended holiday period. While the holiday feature still plays a role by contributing positively to increase predicted load (reflecting the return to workdays), the previous day has a negative influence, resulting in an underestimation of the load on this transition day. By January 8, the model generates more accurate predictions by using January 7 as a reference – indicated by the large influence of the last day on January 8. This shows how the model uses the immediate post-holiday day as a marker for the end of the Christmas period.
- *Temperature effects* (Figure 3 (b)): Rising temperatures on February 20–21 produce negative SHAP values for the temperature feature, appropriately reducing the predicted load, which aligns with the inverse temperature-load relationship we would expect from domain knowledge that a part of the heating load is electric.
- *Holiday effects* (Figure 3 (c)): The May 1 example illustrates the impact of holidays. The holiday feature decreases the load on May 1 (a Thursday) and increases it on May 2 to account for the return to normal working patterns the following day. Similar to January 7, the previous day negatively impacts the prediction. This shows that the holiday feature helps the model to predict that the low load on the previous day is not due to a permanent change in consumption levels, but the load will increase again on the next workday.
- *Irradiance effects* (Figure 3 (d)): On August 20, a lower solar irradiance compared to the previous day leads to adjustments in the midday load curve. This indicates that the model captures the dependency between solar generation and load. Reduced behind-the-meter photovoltaic output lowers self-consumption and increases the net load, which is reflected in a higher prediction.

An analysis of the complete test period (Appendix B.4, Figures 7 and 8) reveals several broader patterns in the importance of covariates. Covariates contribute most strongly on specific days rather than uniformly over time. In particular, their importance increases when temperature or irradiance differs substantially from the previous day, or when the preceding or following day is a holiday. Furthermore, the previous day's load is especially influential for forecasts on Tuesdays, while its importance is markedly lower on Mondays and weekends. This pattern indicates that the model captures weekly cyclical patterns in electricity consumption behavior, distinguishing between workdays and weekends.

7 Conclusion & Future Work

The present work demonstrates that Time Series Foundation Models (TSFMs) offer a promising combination of accuracy and efficiency for load forecasting applications, and that they can be meaningfully explained when paired with efficient XAI approaches such as the one proposed here. We show that Chronos-2 and TabPFN-TS achieve competitive performance with Transformers trained from scratch on domain-specific data, while requiring no training data or computational resources for model development – making accurate forecasting accessible even in scenarios where training data is scarce. The flexibility of TSFMs to operate with varying context lengths and covariates enables efficient computation of exact SHAP values, with full explanations generated in approximately 5 seconds for Chronos-2. Our global and local explanation analysis reveals that TSFMs utilize covariates in ways that align with established domain

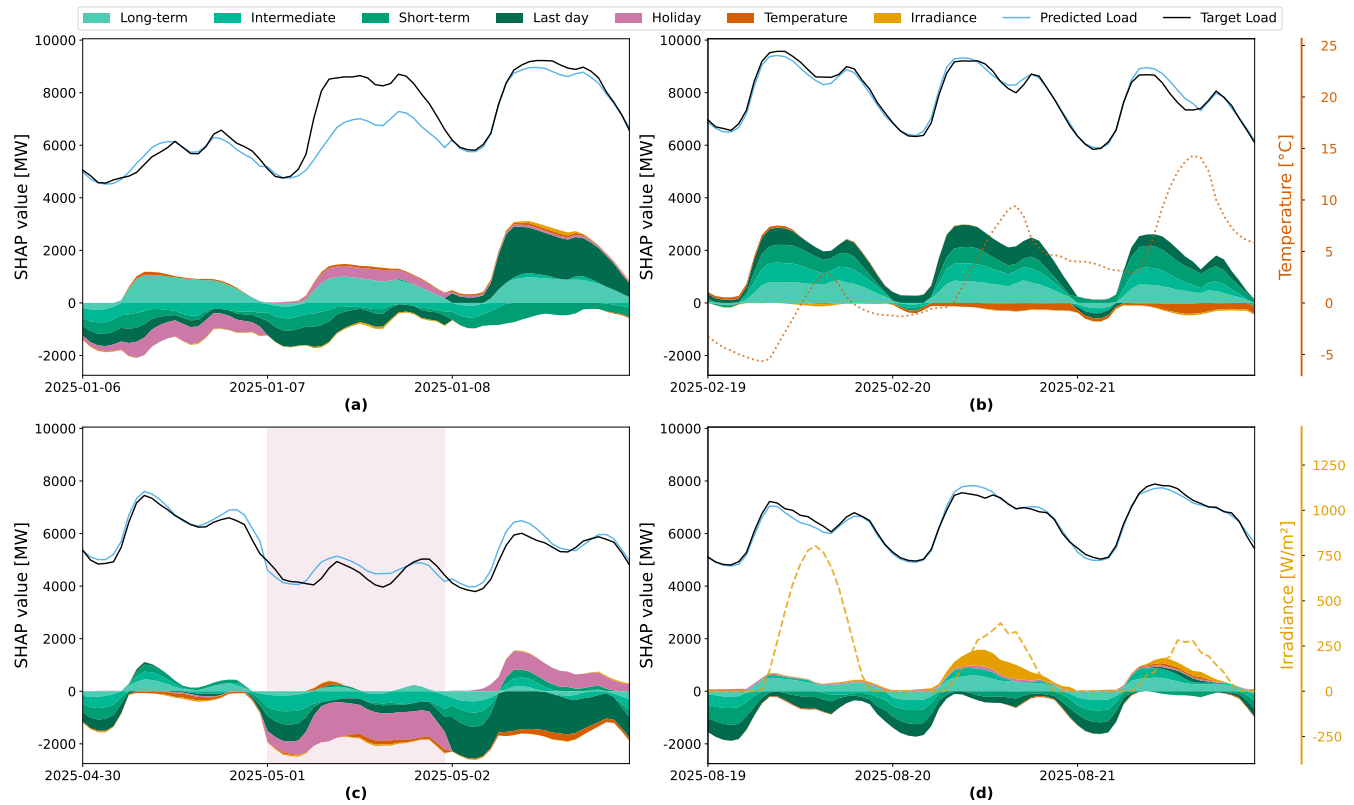


Figure 3: Local SHAP explanations for Chronos-2 predictions on selected days showing feature contributions to load forecasts: (a) post-holiday effects (January 6-8), (b) temperature responses (February 20-21), (c) holiday patterns (May 1-2), and (d) irradiance influence (August 20). Positive (negative) SHAP values correspond to an increase (decrease) in the predicted load.

knowledge of energy consumption patterns, including temperature dependencies, calendar effects, and post-holiday phenomena. Our work demonstrates that TSFMs can satisfy both the accuracy and explainability demands of critical infrastructure applications while preserving their zero-shot advantages.

We identify several promising research directions at the intersection of XAI with TSFMs. Applying our methodology to other energy forecasting tasks – including low-voltage load, price, and renewable generation forecasting – would demonstrate the generalizability of our approach across diverse operational contexts. Since our approach essentially only requires models to accept masked inputs, a capability inherent to most TSFM architectures, it is largely model-agnostic and can be easily generalized to other foundation models. Applying our SHAP adaptation to other TSFMs, both univariate (e.g., [34]) and covariate-informed (e.g., [9]), would therefore provide broader insights into how different architectural choices affect feature attribution patterns in TSFMs. Computational efficiency could be substantially improved by estimating SHAP values from subsets of feature group combinations rather than exhaustively evaluating all 2^N coalitions – enabling finer-grained temporal groupings beyond the current four buckets, which is a particularly important consideration for the slower TabPFN-TS and scenarios

with larger feature sets. Extending our framework to explain probabilistic forecasts and uncertainty quantification would address a critical gap for risk-sensitive operational decisions [41]. Finally, analyzing how fine-tuning affects feature attribution patterns could reveal what domain-specific knowledge is acquired during adaptation and inform best practices for model deployment in specialized forecasting contexts.

Code Availability

The code to reproduce the forecasts and explanations for Chronos-2 and TabPFN-TS are available at <https://github.com/KIT-IAI/SHAP4TSFMs>. The original TSFM implementations and weights are available with the publications [2, 25].

Statement on the Usage of Generative AI

ChatGPT with GPT-5.2 and Claude Sonnet 4.5 were used to improve the fluency of the text. The authors have checked all produced text carefully and take the full responsibility for the submitted manuscript.

Acknowledgments

The authors gratefully acknowledge funding from the Helmholtz Association under the program “Energy System Design” and the

Networking Fund through Helmholtz AI and the HAICORE@KIT partition. We thank Jonathan Kolar for the data preprocessing and Alexander Kreusel for helpful discussions about TSFMs.

References

- [1] Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. 2024. GIFT-Eval: A Benchmark For General Time Series Forecasting Model Evaluation. doi:10.48550/arXiv.2410.10393 arXiv:2410.10393 [cs].
- [2] Abdul Fatir Ansari, Oleksandr Shchur, Jari Kükken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. 2025. Chronos-2: From Univariate to Universal Forecasting. doi:10.48550/ARXIV.2510.15821
- [3] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. 2024. Chronos: Learning the Language of Time Series. doi:10.48550/ARXIV.2403.07815 Version Number: 3.
- [4] Lukas Baur, Konstantin Ditschuneit, Maximilian Schambach, Can Kaymakci, Thomas Wollmann, and Alexander Sauer. 2024. Explainability and Interpretability in Electric Load Forecasting Using Machine Learning Techniques – A Review. *Energy and AI* 16 (May 2024), 100358. doi:10.1016/j.egyai.2024.100358
- [5] João Bento, Pedro Saleiro, André F. Cruz, Mário A.T. Figueiredo, and Pedro Bizarro. 2021. TimeSHAP: Explaining Recurrent Models through Sequence Perturbations. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD '21)*. Association for Computing Machinery, New York, NY, USA, 2565–2573. doi:10.1145/3447548.3467166
- [6] Adrien Bibal, Rémi Cardon, David Alfter, Rodrigo Wilkens, Xiaoo Wang, Thomas François, and Patrick Watrin. 2022. Is Attention Explanation? An Introduction to the Debate. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 3889–3900. doi:10.18653/v1/2022.acl-long.269
- [7] Matthieu Boileau, Philippe Helluy, Jeremy Pawlus, and Svitlana Vyetenko. 2025. Towards Interpretable Time Series Foundation Models. doi:10.48550/arXiv.2507.07439 arXiv:2507.07439 [cs].
- [8] G. E. P. Box and D. R. Cox. 1964. An Analysis of Transformations. *Journal of the Royal Statistical Society: Series B (Methodological)* 26, 2 (July 1964), 211–243. doi:10.1111/j.2517-6161.1964.tb00553.x
- [9] Ben Cohen, Emaad Khwaja, Youssef Doubli, Salahidine Lemaachi, Chris Lettieri, Charles Masson, Hugo Miccinilli, Elise Ramé, Qiqi Ren, Afshin Rostamizadeh, Jean Ogier du Terrail, Anna-Monica Toon, Kan Wang, Stephan Xie, Zongzhe Xu, Viktoriya Zhukova, David Asker, Ameet Talwalkar, and Othmane Abou-Amal. 2025. This Time is Different: An Observability Perspective on Time Series Foundation Models. doi:10.48550/arXiv.2505.14766 arXiv:2505.14766 [cs].
- [10] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. 2024. A decoder-only foundation model for time-series forecasting. <http://arxiv.org/abs/2310.10688> arXiv:2310.10688 [cs].
- [11] Björn Deiseroth, Mayukh Deb, Samuel Weinbach, Manuel Brack, Patrick Schramowski, and Kristian Kersting. 2023. ATMAN: Understanding Transformer Predictions Through Memory Efficient Attention Manipulation. *Advances in Neural Information Processing Systems* 36 (Dec. 2023), 63437–63460. https://proceedings.neurips.cc/paper_files/paper/2023/hash/c83bc020a020cdeb966ed10804619664-Abstract-Conference.html
- [12] ENTOS-E. 2025. Transparency platform. <https://transparency.entsoe.eu>
- [13] European Commission. 2024. Artificial Intelligence Act. https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=OJ:L_202401689#art_13
- [14] Elena Giacomazzi, Felix Haag, and Konstantin Hopf. 2023. Short-Term Electricity Load Forecasting Using the Temporal Fusion Transformer: Effect of Grid Hierarchies and Data Sources. In *Proceedings of the 14th ACM International Conference on Future Energy Systems*. ACM, Orlando FL USA, 353–360. doi:10.1145/3575813.3597345
- [15] Miha Grabner, Yi Wang, Qingsong Wen, Boštjan Blažič, and Vitomir Štruc. 2023. A Global Modeling Framework for Load Forecasting in Distribution Networks. *IEEE Transactions on Smart Grid* 14, 6 (Nov. 2023), 4927–4941. doi:10.1109/TSG.2023.3264525
- [16] Stephen Haben, Siddharth Arora, Georgios Giasemidis, Marcus Voss, and Danica Vukadinović Greetham. 2021. Review of low voltage load forecasting: Methods, applications, and recommendations. *Applied Energy* 304 (Dec. 2021), 117798. doi:10.1016/j.apenergy.2021.117798
- [17] Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, Adrian Simmons, Cornel Soci, Saleh Abdalla, Xavier Abellan, Gianpaolo Balsamo, Peter Bechtold, Gionata Biavati, Jean Bidlot, Massimo Bonavita, Giovanna De Chiara, Per Dahlgren, Dick Dee, Michail Diamantakis, Rossana Dragani, Johannes Flemming, Richard Forbes, Manuel Fuentes, Alan Geer, Leo Haimberger, Sean Healy, Robin J. Hogan, Elias Hölm, Marta Janisková, Sarah Keeley, Patrick Laloyaux, Philippe Lopez, Cristina Lupu, Gabor Radnoti, Patricia de Rosnay, Iryna Rozum, Freja Vamborg, Sebastien Villaume, and Jean-Noël Thépaut. 2020. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society* 146, 730 (2020), 1999–2049. doi:10.1002/qj.3803
- [18] Matthias Hertel, Lara Ambrosius, Manuel Treutlein, Ralf Mikut, and Veit Hagenmeyer. 2025. A comparison of local, cluster-specific and global Transformer models for forecasting electrical loads of individual buildings and substations. In *2025 IEEE Kiel PowerTech*. IEEE, Kiel, Germany, 1–8. doi:10.1109/PowerTech59965.2025.11180482
- [19] Matthias Hertel, Maximilian Beichter, Benedikt Heidrich, Oliver Neumann, Benjamin Schäfer, Ralf Mikut, and Veit Hagenmeyer. 2023. Transformer training strategies for forecasting multiple load time series. *Energy Informatics* 6, 1 (Oct. 2023), 20. doi:10.1186/s42162-023-00278-z
- [20] Matthias Hertel, Simon Ott, Benjamin Schäfer, Ralf Mikut, Veit Hagenmeyer, and Oliver Neumann. 2022. Evaluation of Transformer Architectures for Electrical Load Time-Series Forecasting. In *Proceedings 32. Workshop Computational Intelligence*, Vol. 1. 93. https://library.oapen.org/bitstream/handle/20.500.12657/59840/external_content.pdf?sequence=1#page=105
- [21] Matthias Hertel, Sebastian Pütz, Ralf Mikut, Veit Hagenmeyer, and Benjamin Schäfer. 2025. Explainable time-series forecasting with sampling-free SHAP for Transformers. doi:10.48550/arXiv.2512.20514 arXiv:2512.20514 [cs].
- [22] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. 2023. TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second. doi:10.48550/arXiv.2207.01848 arXiv:2207.01848 [cs].
- [23] Tao Hong. 2014. Energy Forecasting: Past, Present, and Future. *Foresight: The International Journal of Applied Forecasting* 32 (2014), 43–48. <https://ideas.repec.org/a/for/ijafaa/y2014i32p43-48.html>
- [24] Tao Hong, Pierre Pinson, Yi Wang, Rafal Weron, Dazhi Yang, and Hamidreza Zareipour. 2020. Energy Forecasting: A Review and Outlook. *IEEE Open Access Journal of Power and Energy* 7 (2020), 376–388. doi:10.1109/OAJPE.2020.3029979
- [25] Shi Bin Hoo, Samuel Müller, David Salinas, and Frank Hutter. 2025. From Tables to Time: How TabPFN-v2 Outperforms Specialized Time Series Forecasting Models. doi:10.48550/arXiv.2501.02945 arXiv:2501.02945 [cs].
- [26] Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 3543–3556. doi:10.18653/v1/N19-1357
- [27] Nouha Karaoui, Denis Coquenot, Elisa Fromont, Martial Mermillod, and Marina Reyboz. 2025. How Foundational are Foundation Models for Time Series Forecasting? doi:10.48550/arXiv.2510.00742 arXiv:2510.00742 [cs].
- [28] Ahsan Raza Khan, Anzar Mahmood, Awais Saffar, Zafar A. Khan, and Naveed Ahmed Khan. 2016. Load forecasting, dynamic pricing and DSM in smart grid: A review. *Renewable and Sustainable Energy Reviews* 54 (Feb. 2016), 1311–1322. doi:10.1016/j.rser.2015.10.117
- [29] Marian Klobasa, Gerhard Angerer, Arne Lüllmann, Joachim Schleich, Tim Buber, Anna Gruber, Marie Hünecke, and Serafin von Roon. 2014. *Load management as a way of covering peak demand in Southern Germany*. Final Report 040/04-S-2014/EN. Fraunhofer Institute for Systems and Innovation Research (ISI) and Forschungsgesellschaft für Energiewirtschaft mbH. doi:10.24406/publica-fhg-296916
- [30] Siva Rama Krishna Kottapalli, Karthik Hubli, Sandeep Chandrashekhara, Garima Jain, Sunayana Hubli, Gayathri Botla, and Ramesh Doddaiiah. 2025. Foundation Models for Time Series: A Survey. doi:10.48550/arXiv.2504.04011 arXiv:2504.04011 [cs].
- [31] Alexander Kreusel, Matthias Hertel, Moritz Noskiewicz, Heiko Maaß, Ralf Mikut, and Veit Hagenmeyer. 2025. Evaluating Time-Series Foundation Models for Cooling Demand Forecasting with Little Data. In *Proceedings 35. Workshop Computational Intelligence*. Berlin, 301–322.
- [32] Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang, Ming Jin, Dongjin Song, Shirui Pan, and Qingsong Wen. 2024. Foundation Models for Time Series Analysis: A Tutorial and Survey. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 6555–6565. doi:10.1145/3637528.3671451 arXiv:2403.14735 [cs].
- [33] Nan Lin, Dong Yun, Weijie Xia, Peter Palensky, and Pedro P. Vergara. 2025. Comparative Analysis of Zero-Shot Capability of Time-Series Foundation Models in Short-Term Load Prediction. In *2025 IEEE Kiel PowerTech*. IEEE, Kiel, Germany, 1–6. doi:10.1109/PowerTech59965.2025.11180251
- [34] Chenghao Liu, Taha Aksu, Juncheng Liu, Xu Liu, Hanshu Yan, Quang Pham, Silvio Savarese, Doyen Sahoo, Caiming Xiong, and Junnan Li. 2026. Moirai 2.0: When Less Is More for Time Series Forecasting. doi:10.48550/arXiv.2511.11698 arXiv:2511.11698 [cs].

- [35] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, Vol. 30. Canada, 4768 – 4777. <https://dl.acm.org/doi/proceedings/10.5555/3295222>
- [36] R. Machlev, L. Heistrene, M. Perl, K. Y. Levy, J. Belikov, S. Mannor, and Y. Levron. 2022. Explainable Artificial Intelligence (XAI) techniques for energy and power systems: Review, challenges and opportunities. *Energy and AI* 9 (Aug. 2022), 100169. doi:10.1016/j.egyai.2022.100169
- [37] Marcel Meyer, Sascha Kaltenpoth, Kevin Zalipski, and Oliver Müller. 2025. Time Series Foundation Models: Benchmarking Challenges and Requirements. doi:10.48550/arXiv.2510.13654 arXiv:2510.13654 [cs].
- [38] Marcel Meyer, David Zapata, Sascha Kaltenpoth, and Oliver Müller. 2024. Benchmarking Time Series Foundation Models for Short-Term Household Electricity Load Forecasting. doi:10.48550/arXiv.2410.09487 arXiv:2410.09487.
- [39] Pablo Montero-Manso and Rob J. Hyndman. 2021. Principles and algorithms for forecasting groups of time series: Locality and globality. *International Journal of Forecasting* 37, 4 (Oct. 2021), 1632–1653. doi:10.1016/j.ijforecast.2021.03.004
- [40] Amin Nayeibi, Sindhu Tipirneni, Chandan K. Reddy, Brandon Foreman, and Vignesh Subbian. 2023. WindowSHAP: An efficient framework for explaining time-series classifiers based on Shapley values. *Journal of Biomedical Informatics* 144 (Aug. 2023), 104438. doi:10.1016/j.jbi.2023.104438
- [41] Alexandra Nikoltchovska, Sebastian Pütz, Xiao Li, Veit Hagenmeyer, and Benjamin Schäfer. 2025. Probabilistic and Explainable Machine Learning for Tabular Power Grid Data. In *Proceedings of the 16th ACM International Conference on Future and Sustainable Energy Systems (E-Energy '25)*. Association for Computing Machinery, New York, NY, USA, 213–231. doi:10.1145/3679240.3734623
- [42] Atharva Pandey, Abhilash Neog, and Gautam Jajoo. 2025. On the Internal Semantics of Time-Series Foundation Models. doi:10.48550/ARXIV.2511.15324 Version Number: 1.
- [43] Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, Marin Biloš, Sahil Garg, Anderson Schneider, Nicolas Chapados, Alexandre Drouin, Valentina Zantedeschi, Yuriy Nevmyvaka, and Irina Rish. 2023. Lag-Llama: Towards Foundation Models for Probabilistic Time Series Forecasting. doi:10.48550/ARXIV.2310.08278 Version Number: 3.
- [44] David Rundel, Julius Kobiak, Constantin von Crailsheim, Matthias Feurer, Thomas Nagler, and David Rügamer. 2024. Interpretable Machine Learning for TabPFN. In *Explainable Artificial Intelligence*, Luca Longo, Sebastian Lapuschkin, and Christin Seifert (Eds.). Vol. 2154. Springer Nature Switzerland, Cham, 465–476. doi:10.1007/978-3-031-63797-1_23 Series Title: Communications in Computer and Information Science.
- [45] Frederik vom Scheidt, Hana Medinová, Nicole Ludwig, Bent Richter, Philipp Staudt, and Christof Weinhardt. 2020. Data analytics in the electricity sector – A quantitative and qualitative literature review. *Energy and AI* 1 (Aug. 2020), 100009. doi:10.1016/j.egyai.2020.100009
- [46] Shapley, Lloyd S. 1953. A Value for n-person Games. *Contributions to the Theory of Games. Annals of Mathematical Studies. Princeton University Press*. 28 (1953), 307–317. doi:10.1515/9781400881970-018
- [47] Oleksandr Shchur, Abdul Fatir Ansari, Caner Turkmen, Lorenzo Stella, Nick Erickson, Pablo Guerron, Michael Bohlke-Schneider, and Yuyang Wang. 2025. fev-bench: A Realistic Benchmark for Time Series Forecasting. doi:10.48550/ARXIV.2509.26468 Version Number: 1.
- [48] U.S. Department of Homeland Security. 2024. *Roles and responsibilities framework for artificial intelligence in critical infrastructure*. Technical Report. U.S. Department of Homeland Security. https://www.dhs.gov/sites/default/files/2024-11/24_1114_dhs_ai-roles-and-responsibilities-framework-508.pdf
- [49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc., 5998–6008.
- [50] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. 2024. Unified Training of Universal Time Series Forecasting Transformers. doi:10.48550/arXiv.2402.02592 arXiv:2402.02592 [cs].
- [51] Yuyi Zhang, Qiushi Sun, Dongfang Qi, Jing Liu, Ruimin Ma, and Ovanes Petrosian. 2024. ShapTime: A General XAI Approach for Explainable Time Series Forecasting. In *Intelligent Systems and Applications*, Kohei Arai (Ed.). Springer Nature Switzerland, Cham, 659–673. doi:10.1007/978-3-031-47721-8_45
- [52] Florian Ziel. 2018. Modeling public holidays in load forecasting: a German case study. *Journal of Modern Power Systems and Clean Energy* 6, 2 (March 2018), 191–207. doi:10.1007/s40565-018-0385-5

A Extended Methodology

Chronos-2. Chronos-2 [2] is a pretrained foundation model for time series forecasting capable of handling univariate, multivariate, and covariate-informed forecasting tasks. The model architecture tokenizes time series data, enabling it to process temporal sequences

through a transformer-based framework. For covariate-informed forecasting, Chronos-2 builds patches for each variable and uses group attention to incorporate information from the covariates. This design allows the model to learn cross-channel dependencies while leveraging its pretrained representations for zero-shot and few-shot forecasting scenarios.

TabPFN-TS. TabPFN-TS [25] extends TabPFN [22], a foundation model originally designed for classification and regression for tabular data, to the time series forecasting domain. Rather than treating forecasting as a sequential modeling problem, TabPFN-TS reframes it as a tabular supervised learning task where historical observations and future time steps are represented in a tabular format. This approach is particularly effective for small-to-medium sized datasets common in energy forecasting applications. The model explicitly handles covariates as separate input features within the tabular representation. This enables direct integration of weather variables, calendar information, and other contextual data alongside historical load patterns.

B Extended Results

B.1 Evaluation Metrics

We evaluate forecasting performance using three standard regression metrics. Let y_i denote the actual load at time step i , $\hat{y}_i$ the predicted load, and n the number of predictions.

Mean absolute error (MAE) measures the average absolute deviation between predictions and actual values:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2)$$

Root mean squared error (RMSE) penalizes larger errors more heavily due to the quadratic term:

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

Mean absolute percentage error (MAPE) expresses errors as percentages relative to actual values:

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|. \quad (4)$$

B.2 Forecasting Performance with Different Context Lengths

Figure 4 illustrates how forecasting performance varies with context length for both TSFMs. TabPFN-TS shows substantial improvement as context increases, with MAE decreasing from 549 MW at one week to 163.2 MW at 52 weeks of context. In contrast, Chronos-2 achieves near-optimal performance already at 16 weeks of context, plateauing around 156 MW MAE. Both models surpass the persistence baseline (325.9 MW MAE) across all context lengths, with Chronos-2 demonstrating superior performance and greater sample efficiency in utilizing historical information.

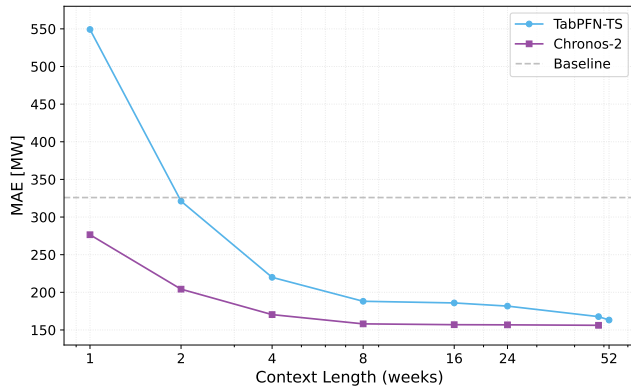


Figure 4: Performance comparison of the TabPFN-TS and Chronos-2 models across different context lengths.

B.3 Global Explanations - Dependence on Temperature Difference

Figure 5 presents the SHAP values for the temperature covariate depending on the temperature of the predicted day and the temperature of the day before. The black line indicates that the temperature is equal to the day before. Points above the line indicate a temperature increase, and points below the line indicate a temperature decrease. Chronos-2 decreases the predicted load with increasing temperatures, shown by the negative SHAP values above the line. TabPFN-TS does so only for temperatures below 15 °C.

B.4 Local Explanations - TabPFN-TS Examples

Figure 6 provides detailed local explanations for TabPFN-TS across representative forecasting scenarios. These examples demonstrate how feature contributions vary with specific operational contexts: post-holiday load recovery (January 6–8), temperature-driven demand variations (February 20–21), holiday effects (May 1–2), and solar irradiance influence on summer demand (August 20). The explanations reveal consistent patterns aligned with energy domain knowledge, including the model’s ability to distinguish between different types of calendar effects and its appropriate weighting of weather variables under varying seasonal conditions.

B.5 Local Explanations - SHAP values of one year

Figures 7 and 8 present local SHAP explanations across the complete test set for Chronos-2 and TabPFN-TS, respectively and complement the focused examples in Section 6. Each figure displays twelve monthly panels (October 2024–September 2025) showing model predictions, SHAP feature attributions, and actual covariate values.

The visualizations reveal how feature importance varies across seasonal cycles, major holidays, and weather events throughout the year. Temperature attributions show strong seasonal patterns, with negative contributions during summer months (reduced heating demand) and positive contributions in winter. Irradiance attributions peak during summer, reflecting increased cooling loads on sunny days. The holiday covariate captures substantial load reductions

during public holidays, especially around major holidays such as New Year’s Day, Easter, and Christmas.

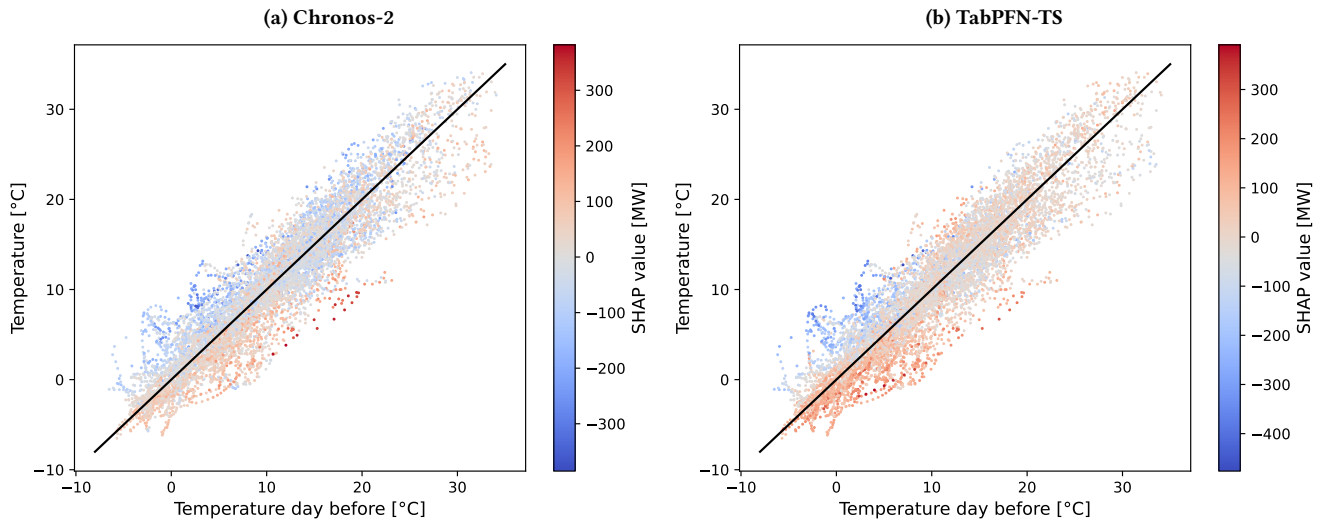


Figure 5: SHAP dependence plots of temperature difference between prediction day and previous day. The back line indicates points where the temperature is equal to the previous day. Both models decrease the predicted load with increasing temperatures (points above the black line), and vice versa (points below the black line).

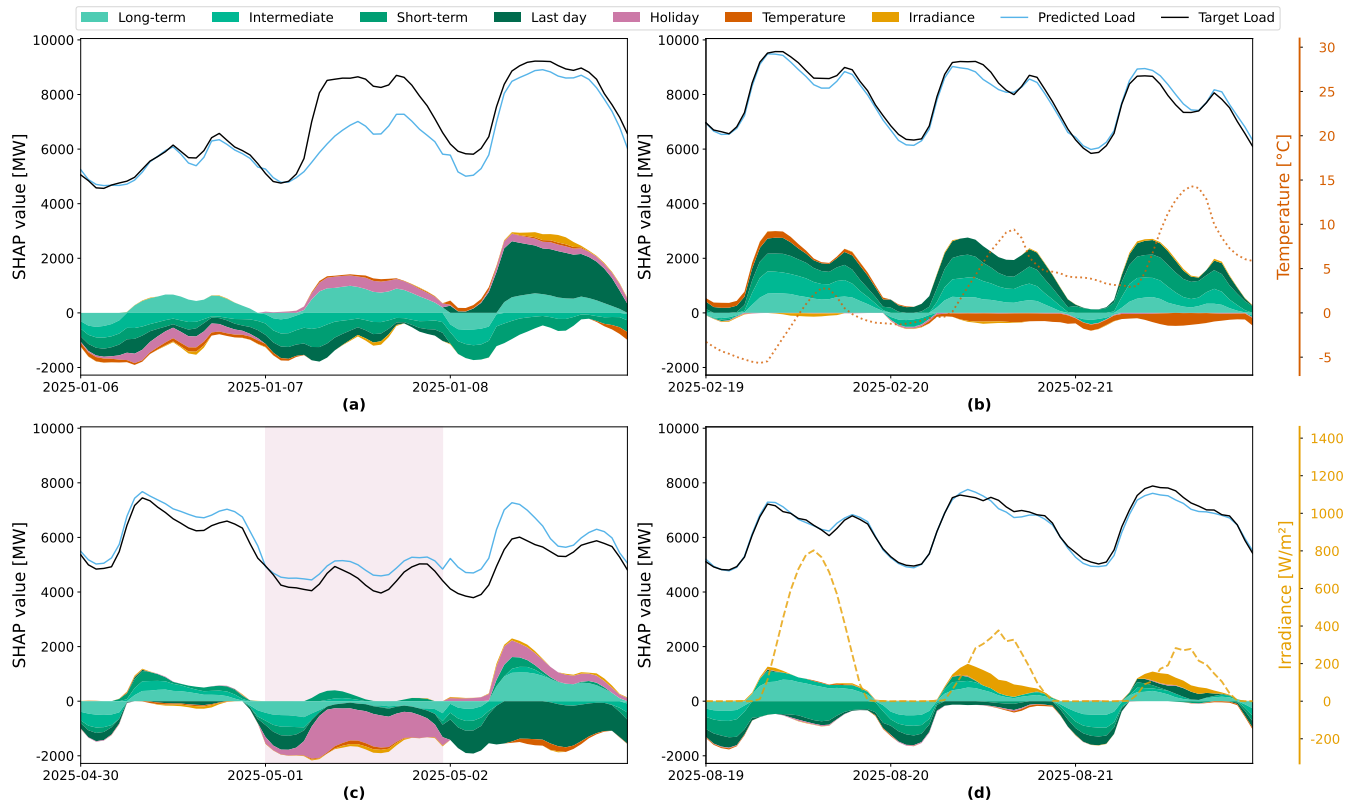


Figure 6: Local SHAP explanations for TabPFN-TS predictions on selected days showing feature contributions to load forecasts: (a) post-holiday effects (January 6-8), (b) temperature responses (February 20-21), (c) holiday patterns (May 1-2), and (d) irradiance influence (August 20). Positive (negative) SHAP values correspond to an increase (decrease) in the predicted load.

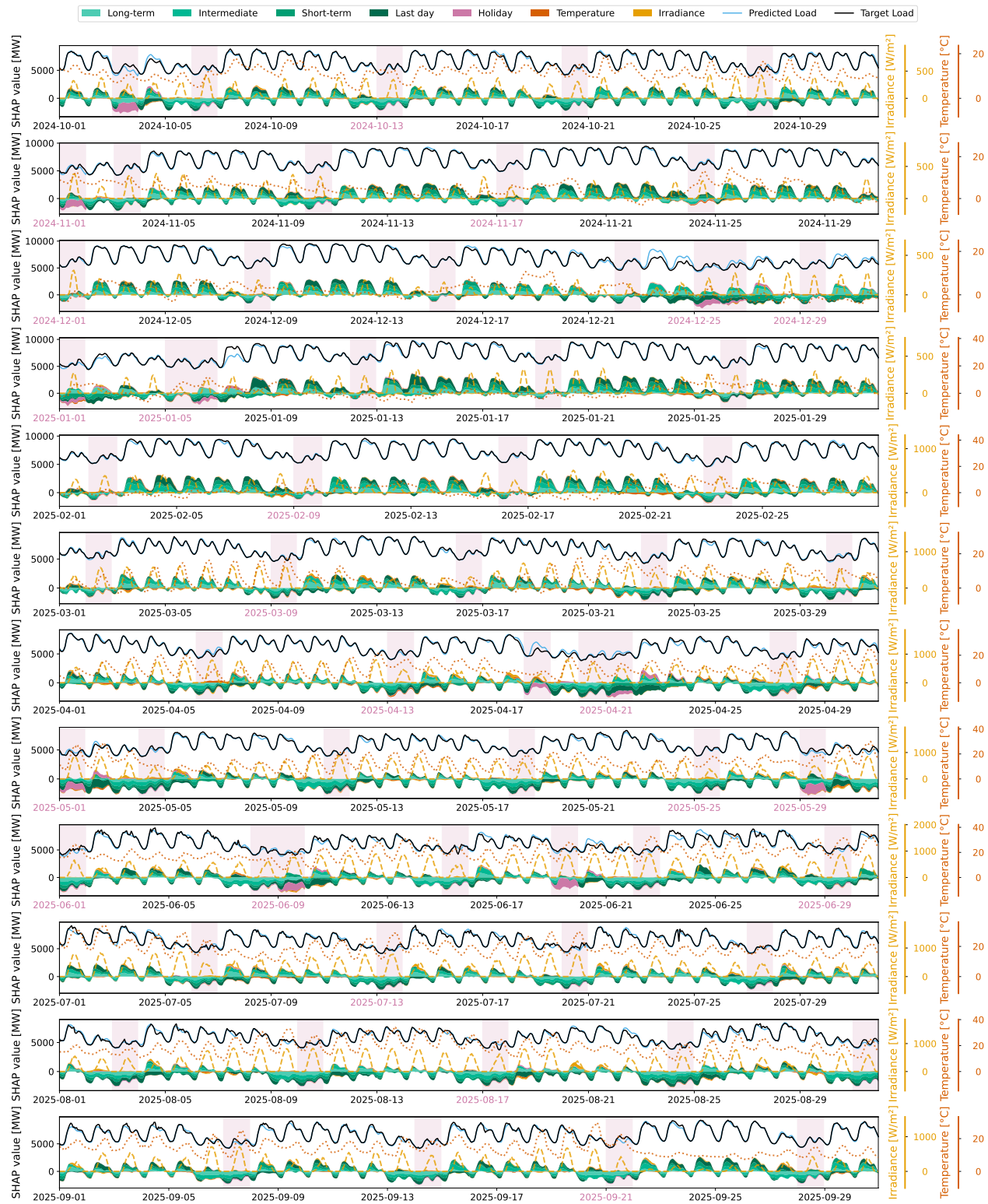


Figure 7: Chronos-2 predictions and local SHAP feature attributions across the full test set, with one panel per month. Each panel shows: actual load (black), model predictions (blue); SHAP values indicating feature contributions to load forecasts, where positive (negative) values correspond to increased (decreased) predicted load; actual covariate values for temperature and irradiance (right y-axis), with Sundays and holidays marked in pink.

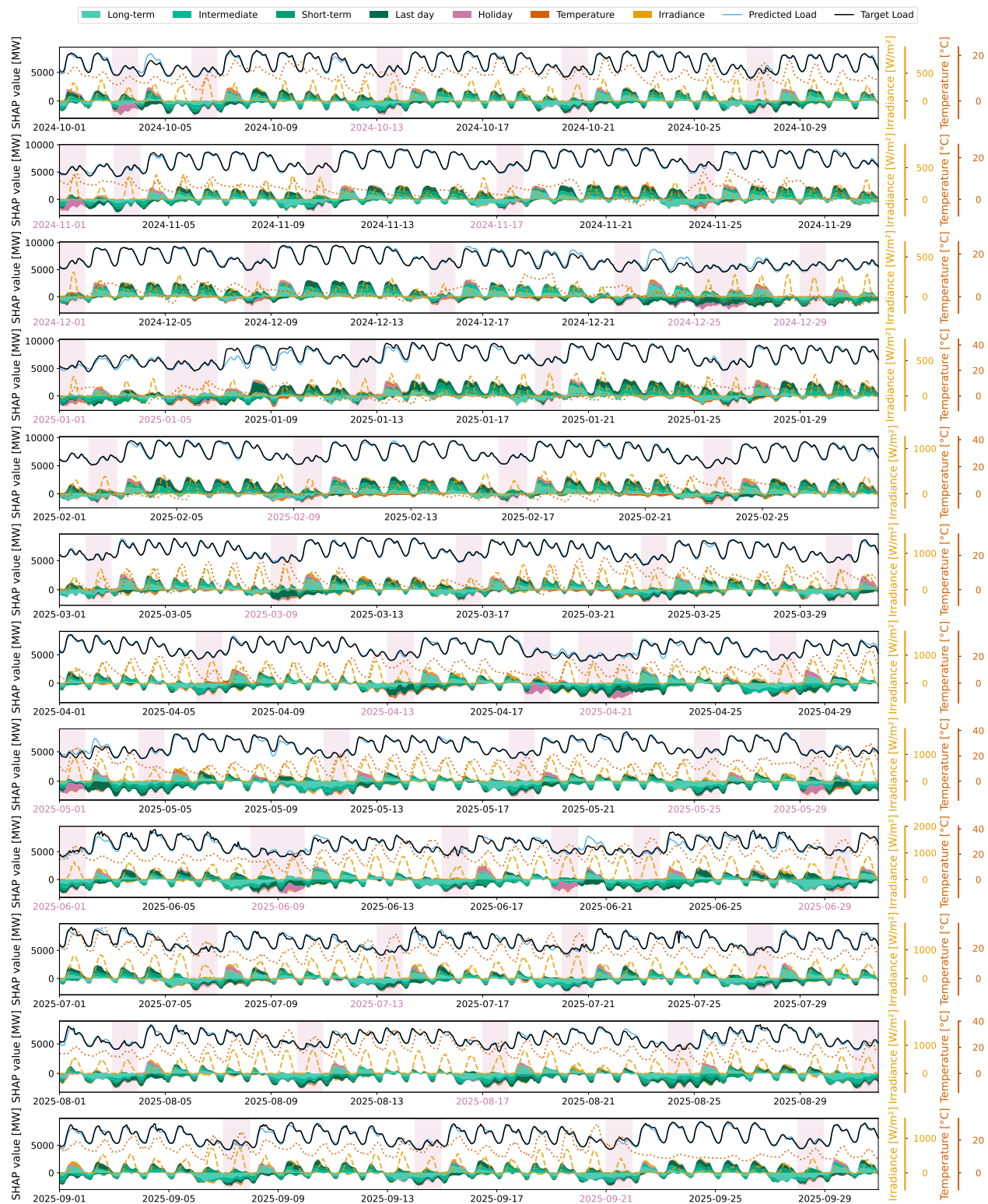


Figure 8: TabPFN-TS predictions and local SHAP feature attributions across the full test set, with one panel per month. Each panel shows: actual load (black), model predictions (blue); SHAP values indicating feature contributions to load forecasts, where positive (negative) values correspond to increased (decreased) predicted load; actual covariate values for temperature and irradiance (right y-axis), with Sundays and holidays marked in pink.