Advanced Intelligent Systems

ADVANCED INTELLIGENT SYSTEMS Open Access

www.advintellsyst.com

RESEARCH ARTICLE OPEN ACCESS

Interpretable Short-Term Electric Load Forecasting

Alessandro Nicola[1] | Giorgia Ghione[2] | Vincenzo Randazzo[2] | Vincenzo Franzitta[3] | Marco Badami[1] | Eros Pasero[2] | Giansalvo Cirrincione[4]

[1]Department of Energy, Politecnico di Torino, Turin, Italy | [2]Department of Electronics and Telecommunications, Politecnico di Torino, Turin, Italy | [3]Department of Engineering, University of Palermo, Palermo, Italy | [4]Lab. LTI, University of Picardie Jules Verne, Amiens, France

Correspondence: Giorgia Ghione (giorgia.ghione@polito.it) | Vincenzo Randazzo (vincenzo.randazzo@polito.it)

Received: 20 January 2026 | **Revised:** 22 June 2026 | **Accepted:** 25 June 2026

Keywords: attention patterns | electrical load | interpretability | short-term load forecasting | temporal fusion transformer | time series forecasting | variable importance

ABSTRACT

The reliable and efficient operation of power systems hinges on accurate electrical load forecasts over various horizons and objectives, ranging from long-term grid adequacy estimation to day-ahead system scheduling and real-time dispatching. In this context, deep learning (DL) models outperform conventional statistical approaches at the expense of interpretability, making the research on interpretable algorithms essential. Among these methods, the temporal fusion transformer (TFT) has proven its validity across multiple sectors, while its applicability to short-term load forecasting (STLF) lacks sufficient corroboration. This study evaluates the TFT within a multivariate STLF task concerning a department building of an Italian university, focusing on interpretability. The performance of the model is evaluated against established benchmarks: The TFT outscores all competitors, yielding a mean absolute percentage error of 7.40% on the test subset, 25% lower than the competitors. Jointly, the interpretability of the model is studied: Variable selection weights identify the most important features, a comprehensive investigation of attention patterns highlights the most critical timesteps for generating forecasts, and the Comaniciu distance helps detect regime shifts and significant events. Overall, this study presents an effective TFT-based model for day-ahead STLF in terms of interpretability and forecasting performance.

1 | Introduction

Electrical demand forecasting has been studied since 1966 to enable the efficient operation of energy systems [1]. Reliable forecasting approaches support improved investment decisions, planning, and regulation of power systems, which in turn facilitate smoother electricity generation, transmission, distribution, and storage [2–4]. Minimizing forecast errors contributes to both energy and financial savings. For instance, one study found that lowering forecast errors by just 1% could save as much as 1.6 million dollars annually for a 10,000 MW electric utility [5]. This motivates researchers to design forecasting models capable of delivering more accurate predictions, especially as electricity demand continues to rise each year. Indeed, according to the US Energy Information, energy final uses are projected to increase by 28% between 2015 and 2040 [6]; together with the increasing share of non-dispatchable renewable energy sources in electricity generation, which in turn necessitates more efficient load management strategies, this makes accurate energy demand forecasting a critical factor in the transition to a low-carbon economy.

According to the literature, load forecasting can be categorized into four main groups: long term, medium term, short term, and very short term [7]. As outlined by [3], long-term forecasting is typically applied to investment and network planning over horizons of 1 year or longer; medium-term forecasting generally covers periods from 1 month up to a year and is mainly employed for capacity, policy and investment planning; short-term forecasting usually considers horizons from 1 day to a week, supporting decisions such as generator unit scheduling and economic dispatch; very short-term forecasting, on the other hand, focuses on predictions within minutes to a few hours and is often used to regulate real-time operations like grid stability. Of all these, short-term load forecasting (STLF) has attracted significant interest due to its practical role in economic dispatch as well as load management and scheduling of system operation. For this reason, the present work is focused on STLF.

Conventional approaches to STLF were univariate, as they relied solely on the observed demand to generate predictions, and were based on statistical models, prioritizing interpretability and computational inexpensiveness over predictive accuracy. Progress in deep learning (DL) algorithms has made DL architectures based on multivariate inputs the state-of-the-art approach [8], achieving unprecedented accuracy. However, trust in their predictions is often undermined by the absence of robust interpretability because of their black box nature, making interpretable DL models a growing research area [9]. Among these, attention-based algorithms, such as temporal fusion transformer (TFT) [10] and ETSformer [11], have been proposed. Although their superior performance and interpretability compared to traditional methods have been reported across multiple sectors, including the prediction of freeway traffic speed [12], solar irradiation [13], photovoltaic generation [14], and bus exhaust emissions [15], the study of their applicability to STLF has been limited.

This study aims to fill this gap by assessing the forecasting performance of the TFT against multiple traditional and attention-based time series forecasting models, considering a real-world case study and with a focus on interpretability.

The main contributions of this work are the following:

- The development and validation of an interpretable DL framework for STLF: A TFT model was implemented and adapted to the forecasting of a real-world university building electrical load, attaining a mean absolute percentage error (MAPE) of 7.40% on the test subset.
- The systematic comparison of the TFT model with a classical statistical model (a Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors model, known as SARIMAX), an established DL architecture (a long–short-term memory-based recurrent neural network, known as LSTM-based RNN or, in brief, LSTM), and an innovative transformer-based DL model (ETSformer).
- The preliminary evaluation of the generalization capability of each forecasting model using unseen data from two additional department buildings within the same university campus.
- A systematic interpretability-driven evaluation on TFT: Variable selection weights (VSWs) and attention weight patterns were analyzed to investigate how the model captures both short-term dependencies and longer-term seasonal patterns. Additionally, the Comaniciu distance metric was employed to examine the model's behavior during atypical periods.

1.1 | Related Work

Various STLF methods have been proposed since the earliest developments of the electricity industry [16], evolving alongside data availability and computational advancements. Early statistical techniques, like ARMA [17], are efficient and interpretable but struggle with highly nonstationary data and complex nonlinear relationships [16, 18]. Conversely, classical machine learning (ML) algorithms better capture nonlinearities and handle noisy datasets, though they require careful feature engineering, accurate hyperparameter tuning, and are prone to overfitting [18]. Recent applications highlight diverse ML and feature selection strategies. Ref. [19] utilized Cubist, an ensemble rule-based model tree, to balance predictive power with interpretability, but its piecewise linear structure limits its ability to capture complex, nonlinear temporal dynamics. Ref. [20] demonstrated that combining cross-domain datasets with advanced feature selection reveals critical consumption drivers often missed in single-domain studies. Ref. [21] proposed a stacking ensemble model to generate joint probabilistic forecasts of industrial loads, but the stacking architecture offers limited insight into temporal patterns.

DL algorithms gained momentum in the 1980s and 1990s following the theorization of feedforward neural networks (FNNs) and, later, the backpropagation algorithm [8]. FNNs bridged traditional ML and modern DL algorithms; for example [22], utilized a deep FNN stacking ensemble for day-ahead electricity forecasting. Yet FNN-based architectures lack explicit mechanisms for modeling temporal dependencies. Recently, the availability of extensive datasets and increased computational power has established DL models like RNNs (especially LSTMs and Gated Recurrent Units, i.e., GRUs) and convolutional neural networks (CNNs) as benchmarks for STLF [8, 18, 23]. In [24], a simple LSTM-based model was proposed: It achieves a mean absolute percentage error (MAPE) of 1.49% and 1.52% over 2 years, surpassing an exponential smoothing average and a fully connected FNN baseline, but its architecture remains opaque. RNN and CNN architectures excel at modeling complex, nonlinear, and long-term dependencies driving the evolution of the electric load [8, 23], albeit with increased computational intensity and reduced interpretability.

Hybrid models are also highly prevalent in recent literature to further improve forecasting performance. Some combine statistical methods and early ML algorithms, such as pairing dynamic mode decomposition with Gaussian process regression [25], or utilizing statistical filtering to enhance traditional STLF [26]. However, these models are constrained by computational complexity, linear approximations, and normality assumptions. Other models combine DL architectures [27]: merged the feature extraction capabilities of CNNs with the temporal modeling capabilities of bidirectional RNNs, achieving a MAPE of 0.36%, with a 21.7% relative improvement compared with LSTM and GRU models [28], and framed load forecasting as an image inpainting task via a 3D convolution-LSTM-convolution stack, which surpassed the performance of multiple conventional models but primarily considering a longer horizon (1 week). Diverse model combinations are common, including Prophet with LSTM [29], a CNN-BiLSTM with Sparrow Search optimization [30], and a joint XGBoost and Robust Random Cut Forest framework for simultaneous forecasting and anomaly detection [31]. Despite their diversity, these hybrid approaches share a common limitation: The gains in predictive accuracy come at the expense of model transparency, and none studies interpretability in operational settings.

As models become more complex, they increasingly function as black boxes, which limits their operational trust in energy management systems, where interpretability is required for the safety and efficiency of energy systems. Current solutions to this problem include utilizing inherently interpretable, low-complexity models [19], whose predictive abilities are limited when dealing with high volatility, or leveraging time series decomposition [25] or post hoc methods like Shapely Additive Explanations (SHAP) [32], which often introduce additional computational overhead and fail to provide inherent insights into the decision-making process of the model. Attention-based and transformer-inspired architectures [33] constitute a growing research area in multiple sectors, including aerospace [34], biomedical engineering [35], and energy, specifically, electrical load forecasting [9]. In this sector, a variant of the attention mechanism was successfully integrated into RNNs with importance weighting in [36], but no study of the interpretability of the model is provided.

Among the transformer variants proposed for time series forecasting, including the Informer [37] (which reduces self-attention complexity for long sequences) and the ETSformer [11] (which integrates exponential smoothing into the Transformer architecture), most prioritize predictive performance while offering limited or no built-in interpretability mechanisms, making them difficult to deploy in energy systems, where safety is critical. In contrast, the TFT [10] is notable for its uncertainty-aware, multi-horizon forecasting capabilities and inherent interpretability blocks for feature selection (variable selection weights, i.e., VSWs) and temporal processing (attention weights). Although the TFT has proven to significantly outperform a wide variety of benchmarks across diverse time series datasets, a significant research gap remains in the quantitative validation of its interpretability tools in real-world STLF, to the best of the authors' knowledge. Many studies employing the TFT either omit interpretability analyses [38], rely heavily on post-hoc explanations [39], or limit their scope solely to feature importance [40, 41]. Other works offer only preliminary, qualitative discussions of attention weights without extensive analysis [42, 43]. Notably, previous research [44] found that inherent feature importance scores can yield conflicting results compared to feature permutation, a finding this article also corroborates. Finally, while [45] provides an interpretability analysis alongside benchmark comparisons, it focuses on a custom architecture without analyzing the behavior of the model during atypical load periods. Additionally, to the best of the authors' knowledge, current STLF literature lacks robust metrics, such as the Comaniciu distance, a measure of dissimilarity between discrete probability distributions used in [10], to quantify how attention shifts during these periods. Therefore, providing a complete analysis of the inherent interpretability of TFT constitutes a primary contribution of this work.

To address these gaps, this work proposes a framework that concurrently achieves high predictive performance and interpretability in real-world STLF. Unlike methods that have limited capacity to capture nonlinear and long-range temporal patterns or have limited interpretability, often in the form of basic feature importance, the core innovation of this work is a systematic exploration of the performance and inherent interpretability mechanisms of the TFT. Specifically, attention patterns are analyzed, and the Comaniciu distance metric is leveraged to measure and analyze shifts in attention patterns during atypical load periods, moving beyond traditional visual inspections of attention weights. This framework is robustly validated against SARIMAX, LSTM, and ETSformer baselines, assessing its performance across multiple buildings. Broader validation across different building types, geographic climates, and varied operational contexts remains an important avenue for future work. Ultimately, this framework advances the operational viability of DL models in energy management systems.

2 | Methods

This section describes the model used in this work, the case study, and the preprocessing and evaluation stages. Figure 1 shows a complete workflow of the proposed framework, including the data preprocessing step, hyperparameter tuning and model training, and, finally, the performance study and interpretability analysis steps.

All code associated with the current submission is available on GitHub at TFT-for-STLF.

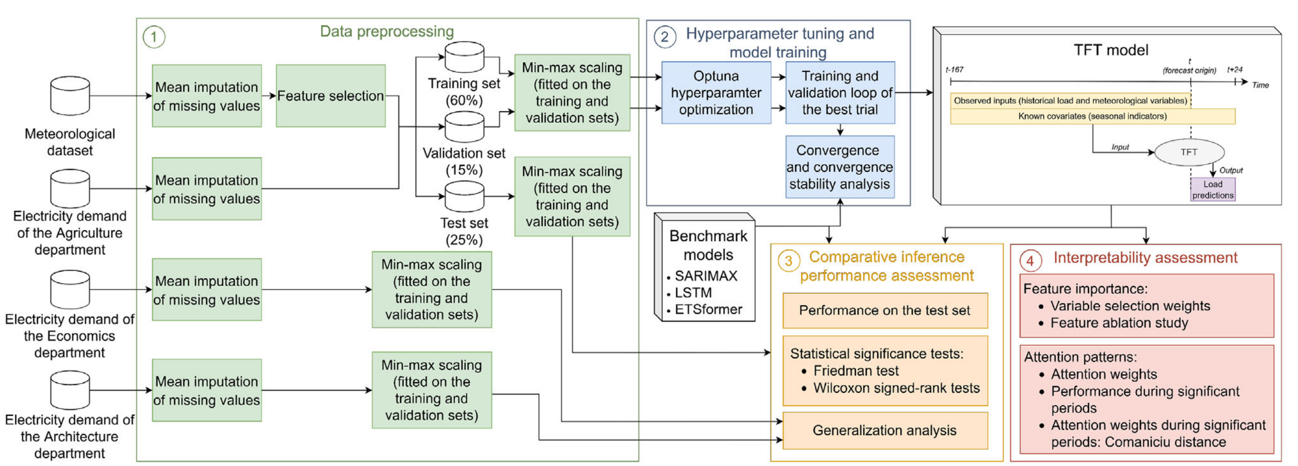


FIGURE 1 | Workflow of the proposed framework.

2.1 | Temporal Fusion Transformer

The TFT, originally introduced in [10], is a state-of-the-art Transformer-inspired direct model for multihorizon time series forecasting. Alongside the dedicated processing for each data channel (time-dependent variables, observed or known a priori, and static covariates), which valorizes all types of time series input data, and the vast use of residual connections and gating layers, which ensures flexible network depth and degree of nonlinear processing, the architectural elements of TFT provide interpretability insights on the rationale behind model predictions.

In any data channel, at each processing step, raw inputs are mapped to embeddings, which are further processed and linearly combined into a single feature relevance-weighted embedding according to the VSWs from the channel-specific VSN.

A many-to-many encoder-decoder LSTM-based unidirectional RNN, initialized with static information, provides the model with the ordering of time-dependent inputs and integrates short-term temporal dependencies.

Each output activation is enriched with static information. The resulting features constitute keys ($\mathbf{K}$), queries ($\mathbf{Q}$), and values ($\mathbf{V}$) of the interpretable multihead attention mechanism (*IMH*) over the temporal dimension, responsible for enhancing the feature of any horizon with the information encoded in the features of the most relevant previous timesteps for describing the temporal dynamics of the same horizon. *IMH* novelty is to share value weights ($\mathbf{W}_{\mathrm{V}}$) across all heads: let m_{H} be the number of attention heads, $\mathbf{W}_{\cdot}^{(\mathrm{h})}$ be head-specific weights, $\mathbf{W}_{\mathrm{H}}$ represent a final linear transformation, and $\mathrm{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{Softmax}(\mathbf{Q}\mathbf{K}^{\top}/\sqrt{d_{\mathrm{k}}})\mathbf{V}$ denote the attention mechanism of vanilla Transformers (d_{k} is the key dimensionality), the contextualized representations for all horizons are retrieved as in Equation (1). Decoder masking prevents any horizon from paying attention to timesteps after itself.

$$IMH(\mathbf{Q},\mathbf{K},\mathbf{V}) = \left[\frac{1}{m_{\mathrm{H}}}\sum_{h=1}^{m_{\mathrm{h}}}\mathrm{Attention}\left(\mathbf{Q}\mathbf{W}_{\mathrm{Q}}^{(\mathrm{h})},\mathbf{K}\mathbf{W}_{\mathrm{K}}^{(\mathrm{h})},\mathbf{V}\mathbf{W}_{\mathrm{V}}\right)\right]\mathbf{W}_{\mathrm{H}} \tag{1}$$

Finally, the contextualized representation of each horizon is further processed and fed to independent linear units whose outputs are used to perform a multi-conditional quantile regression task.

Lim et al. [10] also proposed a metric to detect abrupt variations in the temporal attention pattern captured by the model (e.g., regime shifts or significant events): $\mathrm{dist}(t)$ quantifies the horizon-averaged dissimilarity between the heads-averaged attention pattern $\boldsymbol{\alpha}(t,\tau)$ of a horizon in a given sequence, and the sequences-averaged and heads-averaged attention pattern $\overline{\boldsymbol{\alpha}}(\tau)$ of that same horizon. As shown in Equations (2)–(4), $\mathrm{dist}(t)$ is based on the Comaniciu distance $\kappa(\cdot)$ [46], which measures the dissimilarity between probability vectors ($\boldsymbol{p}$ and $\boldsymbol{q}$, p_j and q_j being their j-th element) by taking the complement of the Bhattacharyya coefficient $\rho(\cdot)$ [47].

$$\rho(\boldsymbol{p},\boldsymbol{q}) = \sum_j \sqrt{p_j q_j} \tag{2}$$

$$\kappa(\boldsymbol{p},\boldsymbol{q}) = \sqrt{1-\rho(\boldsymbol{p},\boldsymbol{q})} \tag{3}$$

$$\mathrm{dist}(t) = \frac{1}{\tau_{\max}}\sum_{\tau=1}^{\tau_{\max}}\kappa(\overline{\boldsymbol{\alpha}}(\tau),\boldsymbol{\alpha}(t,\tau)) \tag{4}$$

In this work, $\mathrm{dist}(t)$ will be denoted as the Comaniciu distance between attention patterns or, in brief, the Comaniciu distance.

2.2 | Case Study

The dataset employed in this work represents a multivariate extension of the one adopted in the earlier univariate study that inspired the present research [48]. It consists of data obtained through direct measurements in three departmental buildings of the University of Palermo, Italy (Agriculture, Architecture, and Economics). These measurements were aggregated to an hourly resolution and subsequently made available for the purposes of this study. Overall, the dataset covers the period from 1 July 2019 to 30 June 2023, inclusive, amounting to 35,064 hourly observations for each of the 24 features. The variables are organized into four main categories.

- **Endogenous variables**: These are the target time series, namely the hourly electricity demand of the three departmental buildings, considered separately. Their average weekly load profiles are shown in Figure S1. All models were trained (or fitted) exclusively on the load data from the Agriculture building, whereas data from the Architecture and Economics departments were employed to assess the generalization performance.
- **Exogenous variables**: These are time series that indirectly influence the dynamics of the endogenous variables. They are shared across all departments and provide crucial contextual information for the forecasting process. They include:
  - **Meteorological variables**: mean air temperature, precipitation, air pressure, mean air humidity, solar irradiation, solar irradiance, and wind data (direction, mean and maximum speed, and frequency of zero-speed events) at 2- and 10-meter heights.
  - **Seasonal indicators**: season, month, day of the week, day of the month, hour of the day, and day of the year. Season, month, and day of the week were represented using ordinal encoding.

While the case study is geographically limited, the focus of this work is on interpretability mechanisms rather than absolute forecasting accuracy.

2.3 | Preprocessing

A preprocessing phase was performed to prepare the raw data for model training, as illustrated in Step 1 of the workflow in Figure 1.

First, missing values were imputed with a mean-based approach.

To reduce the dimensionality of the model's input space while retaining most of the information contained in the original meteorological dataset, a feature selection step was performed. In the dataset used in this study, several of the initial features carried redundant information, as indicated by the correlation matrix in Figure S2. As a result, the following variables were excluded: solar irradiation, maximum wind speed, frequency of null wind speed measurements at heights of 2 and 10 m, mean wind speed at 10 m, and the day of the year. The day of the month and the hour of the day were excluded from the dataset to further reduce its dimensionality.

Moreover, based on domain knowledge, the wind direction was deemed unlikely to provide particularly relevant information, and this feature was therefore removed from the dataset.

An overview of the final dataset, comprising the meteorological variables and the historical load of the Agriculture department, is provided in Figure S3.

Then, the dataset was split into a train subset and a test subset. The train subset included data from the first 3 years (1 July 2019–30 June 2022), corresponding to 26,304 samples, while the test subset comprised the subsequent period (1 July 2022–30 June 2023), containing 8,760 samples. Due to the size of the lookback window, i.e., 1 week (168 h), it is not possible to generate predictions for the first week; therefore, the forecasting period extends from 00:00 on 8 July 2022 to 23:00 on 30 June 2023, both limits included.

Precursor matrices for train and test inputs were extracted from the respective preprocessed subsets of the department of Agriculture: The first 80% of rows of the former matrix were used as the training set, and the remainder as the validation set.

In order to prevent training instability and decrease the sensitivity of learning to weight initialization, the variables were scaled with a min-max scaling approach, fitting the scaler on the training and validation sets.

To further assess the generalizability of the model, the historical load data of the Architecture and Economics departments were preprocessed as well: Missing values were estimated through mean imputation and then the data were min-max scaled (using the previously described scaler).

2.4 | Evaluation

The performance of the tested models was quantitatively assessed and compared using the following metrics: MAE, Mean Squared Error (MSE), RMSE, MAPE, and coefficient of determination (R^2).

MAPE was selected as the primary evaluation metric in this study since it is easily interpretable and can be used to compare forecasts on different scales.

3 | Experimental Setup

The implemented TFT was built with the PyTorch Forecasting library [49]. The hyperparameter tuning and model training process is schematized in Step 2 of the workflow in Figure 1.

Internally, the recurrent backbone used by the TFT was configured with a single LSTM layer. The standard variable selection and gated residual blocks, the attention component, and the output head provided by the library were employed.

The datasets were structured based on feature types and processed by the model accordingly. Known covariates comprised the month of the year and day of the week, while unknown covariates included the target variable together with meteorological variables. Encoder and decoder window lengths defined the historical input and the forecast horizon (i.e., 168 and 24, respectively) and were used by the PyTorch Forecasting framework to automatically generate rolling windows for model training, validation, and testing with unit stride, resulting in repeated predictions for the same timestamp.

Training used a quantile loss as the objective, Adam as the optimization algorithm, and mini-batches of size 64 with shuffling at every epoch. The learning rate was halved whenever the validation loss did not improve for four consecutive epochs, and global norm-based gradient clipping was enabled.

Details on the hyperparameter tuning procedure are available in the supplement. The resulting model features 313,007 trainable parameters and is characterized by a hidden size of 66, a hidden continuous size of 34, and a single attention head; its training was based on a dropout rate of 0.295, a gradient clip value of 0.71, and a learning rate of approximately 0.0089.

3.1 | Benchmark Models

This section presents the implementation and the hyperparameter tuning of benchmark models considered in this study.

3.1.1 | *Seasonal AutoRegressive Integrated Moving Average With eXogenous Regressors*

The SARIMAX model was implemented using instances of the class ARIMA from the statsmodels library [50] in which the electricity consumption represented the endogenous process and the selected seasonal indicators constituted the exogenous variables (as meteorological features are unknown in forecast horizons); the trend polynomial was not included. An automatic tuning procedure was implemented to determine the most suitable orders for SARIMAX (order of the autoregressive polynomial, of differencing, and of the moving average polynomial, both for non-seasonal effects and seasonal effects), as described in the supplement. The best model considers seasons made up of 168 observations (weekly seasonality); the seasonal autoregressive polynomial order, the seasonal moving average polynomial order, and all non-seasonal orders are null, while inputs undergo a first-order seasonal differencing.

3.1.2 | *Long Short-Term Memory*

The LSTM model, implemented using Tensorflow [51], consists of a unidirectional encoder deep RNN composed of two stacked LSTM layers, followed by two fully connected linear layers, providing a balance between network depth and computational efficiency. The output activations of the first LSTM layer served as inputs to the second one. Precursor vectors for train and test labels were built through a rolling-window process on the input and label precursors. Each input sequence contains 168 consecutive rows of its precursor, while each label sequence consists of the 24 consecutive rows following the previous ones, but from its own precursor. A stride of 24 rows between consecutive sequences was applied to avoid overlapping predictions for the same timestamp.

MSE was the loss function, and Adam was the selected optimization algorithm. To encourage steady improvement during training, gradient clipping was enabled (according to a global norm of 1.0), and the learning rate was decreased by a factor of 10 each time the validation loss failed to show improvement over 4 successive epochs.

Further details on hyperparameter tuning are presented in the supplement. The best instance of the model has 38,680 trainable parameters (a reduction of 87.64% compared with the TFT), and it was trained using a mini-batch size of 16, a dropout rate of 0.12, and a learning rate of 0.0021; architecture-wise, its first LSTM layer groups 64 LSTM cells, its second LSTM layer is formed by 32 LSTM cells, and its first fully connected layer is composed of 128 linear units.

3.1.3 | *ETSformer*

Originally proposed by [11], the ETSformer is an attempt to tailor vanilla Transformers to time series data by exploiting the principles of exponential smoothing and frequency domain analysis to generate inductive biases for modeling interpretable time series components (level, growth, and season). It is a direct model based on the residual learning of a stack of encoder layers, each extracting time series components via novel multihead exponential smoothing attention and frequency attention mechanisms. These components are then fed into the corresponding decoder layer for their extrapolation into the forecast horizon.

The ETSformer implementation follows the original codebase [11] and adopts the standard encoder-decoder layout with temporal convolutional token mixing and spectral attention. Unless otherwise specified, default architectural settings are used. The network employs a temporal convolutional kernel size of 3, a dropout rate of 0.2, and a feed-forward subnetwork with sigmoid activation. Inputs to the encoder comprise all meteorological features and the historical load (a total of seven variables), as the aforementioned mechanisms do not rely on seasonal indicators.

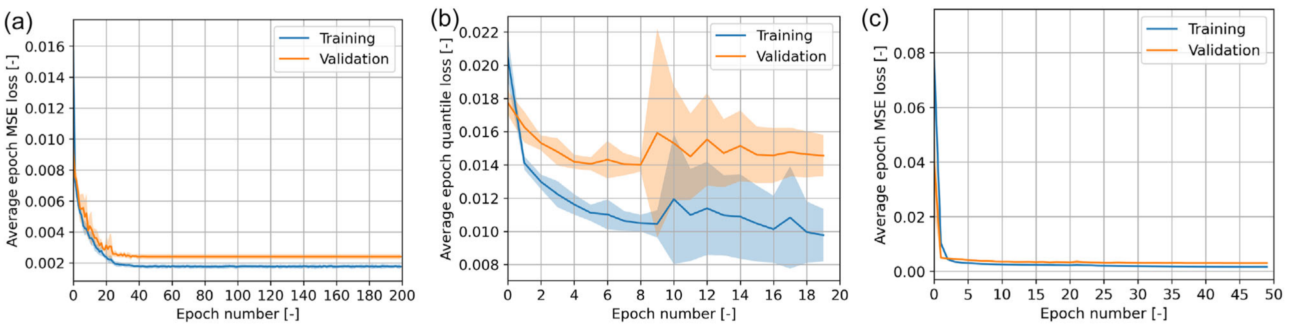
The mechanisms to define time sequences are the same as described for the LSTM, but with a unitary stride as for the TFT. Mini-batches of size 64 are sampled with shuffling at every epoch.

Training minimized the MSE loss using the Adam optimizer. The learning rate schedule follows the ETSformer default: a linear warm-up followed by cosine annealing for all parameters except for smoothing and damping parameters, featuring a constant learning rate 100 times larger than the post-warm-up value. Gradients were clipped with a global norm value of 1.0, and early stopping monitored the epoch validation loss with a patience of 8 epochs.

More details on hyperparameter tuning are presented in the supplement. The best model instance was trained with a learning rate of 0.0026 and features four paired encoder and decoder layers, a latent space dimensionality of 252, and 64 units in each feed-forward module; moreover, nine attention heads extract and extrapolate the latent growth component while the three highest amplitude Fourier bases form the latent seasonal component. The tuned model features 666,479 trainable parameters, which is more than double the parameters of the TFT and a 1,623% increase compared with the LSTM, resulting in the most complex model considered in this study; nonetheless, the TFT remains the most expensive model in terms of inference time.

3.2 | Convergence Analysis and Convergence Stability Analysis

Following the convergence analysis proposed by [52], the convergence dynamics of the benchmarked trainable models are evaluated across 10 independent training runs for each model to assess the stability and reproducibility of the training process. This analysis is part of Step 2 depicted in the workflow in Figure 1. Figure 2 illustrates the resulting average learning curves. All models exhibit robust learning dynamics. The learning curve of the TFT shows a sharp decrease in quantile loss during the initial epochs, followed by increased variability, as reflected by a higher standard deviation after epoch 8. This variability is likely attributable to the complexity of the attention-based architecture and its increased susceptibility to overfitting compared with the



LSTM TFT ETSformer

FIGURE 2 | Average learning curves of the tuned (trainable) models (10 runs per model); shaded regions indicate ±1 standard deviation.

LSTM and ETSformer models. Consequently, training was limited to 20 epochs.

To quantitatively assess the convergence stability of the TFT and the trainable benchmark models as in [52], Table 1 reports the minimum, maximum, mean, and standard deviation of the training loss at the final epoch, computed across the 10 independent training runs for each model. The ETSformer demonstrates greater stability than the LSTM, while a direct comparison with the TFT is not appropriate since the TFT employs a quantile loss rather than the MSE loss; nevertheless, all models exhibit robust training performance.

The learning curves of the training run used for performance evaluation are provided in Section S2.

4 | Results and Discussion

This section analyzes the performance of the tuned models. First, a comparative assessment of inference performance is reported, as depicted in Step 3 of the workflow in Figure 1: The first subsection evaluates their accuracy in predicting the electricity consumption of the Department of Agriculture on the test subset, demonstrating the superiority of the TFT through statistical tests; the second subsection examines their ability to generalize to other buildings with potentially different data distributions. The final subsection outlines the interpretability tools available in the TFT, which are summarized in Step 4 of the workflow in Figure 1.

Evaluation metrics are based on the forecasted conditional mean for SARIMAX, LSTM, and ETSformer and on the forecasted conditional median for the TFT. Moreover, only the sequences whose forecast horizon starts at midnight are considered, ignoring duplicate predictions for the same timestamp.

4.1 | Performance on the Test Subset

Table 2 reports the evaluation metrics obtained from the predictions of each tuned model on the test subset of the Department of Agriculture.

The TFT achieves the smallest MAE, MAPE, MSE, and RMSE values, resulting in the highest R^2. Among the remaining models, the LSTM ranks second in all evaluation metrics, with a 27.04% increase in MAE, a 30.57% increase in MSE, a 14.26% increase in RMSE, a 33.38% increase in MAPE, and a 2.49% decrease in R^2. Both SARIMAX and ETSformer perform considerably worse. When using TFT as the reference, SARIMAX shows relative increases greater than 78% in errors, with a much larger increase of 218.47% in MSE, which is exceptionally high compared to the other models. These results imply that SARIMAX likely generates a higher number of large absolute errors compared with the other approaches, suggesting that it may have difficulty capturing sudden spikes or drops in electricity demand. By contrast, ETSformer presents a smaller and more consistent degradation across all metrics, but with the worst MAPE overall, higher than the TFT one by 85.27%. This indicates that, on average, the ETSformer exhibits notably higher absolute errors than the SARIMAX when the electricity consumption of the Department of Agriculture is at moderate levels, suggesting that the ETSformer performs less effectively than the other models during periods of low electricity demand.

TABLE 2 | Performance of the tuned models on the sequences whose forecast horizon starts at midnight of the Department of Agriculture in the test subset.

Model	MAE, kWh	MSE, kWh^2	RMSE, kWh	MAPE, %	R^2, –
SARIMAX	15.78	806.75	28.40	13.46	0.756
LSTM	10.90	330.77	18.19	9.87	0.900
TFT	**8.58**	**253.32**	**15.92**	**7.40**	**0.923**
ETSformer	14.26	507.94	22.54	13.71	0.846

Note: The best performance for each metric is in bold characters.

To support the claim that the TFT outperforms the benchmark models with statistical significance, a Friedman test at the 5% significance level was conducted on the absolute forecast errors of all models, suggesting that at least one of them performs differently from the others. Subsequently, pairwise Wilcoxon signed-rank tests were performed on the differences between the absolute forecast errors of the TFT and those of each benchmark model, controlling the family-wise error rate at 5% (Holm–Bonferroni correction) and considering as the alternative hypothesis that the differences follow a distribution stochastically less than one symmetric about zero. The latter tests strongly supported the superiority of the TFT over the benchmark models. The results of all statistical tests are reported in Table 3.

Evaluating the practical trade-off between accuracy, interpretability, and computational cost is crucial; hence, model execution times are reported in Table 4. All computations were conducted using Python 3.12.8 on a local machine equipped with an AMD Ryzen 9 5900HX CPU, an NVIDIA GeForce RTX 3050 Ti Laptop GPU with 4 GB of VRAM, and 16 GB of RAM. Among trainable models, the LSTM is the fastest one on average during both training and inference. Compared to the TFT, the ETSformer is 25% faster during inference due to the lower complexity of its attention mechanism ($\mathcal{O}(L \log L)$ versus $\mathcal{O}(L^2)$, where L is the

TABLE 1 | Summary statistics of the training loss at the final epoch (10 runs per model).

Model	Loss type	Minimum, –	Maximum, –	Mean, –	Standard deviation, –
LSTM	MSE	1.50×10^{-3}	1.95×10^{-3}	1.76×10^{-3}	1.33×10^{-4}
TFT	Quantile	8.20×10^{-3}	1.38×10^{-2}	9.77×10^{-3}	1.58×10^{-3}
ETSformer	MSE	1.49×10^{-3}	1.79×10^{-3}	1.61×10^{-3}	9.78×10^{-5}

TABLE 3 | Results of the Friedman test (5% significance level) and the pairwise Wilcoxon signed-rank tests (Holm–Bonferroni correction, 5% family-wise error rate).

Test	Compared models	(Adjusted) *p* value, –
Friedman	SARIMAX, LSTM, TFT, ETSformer	≪0.05
Wilcoxon signed-rank	TFT, SARIMAX	≪0.05
	TFT, LSTM	≪0.05
	TFT, ETSformer	≪0.05

TABLE 4 | Execution times for the tuned models (values indicate mean ± standard deviation).

Model	Training time per epoch, s	Inference time per test sequence, ms	Fit-and-forecast time per test sequence, s
SARIMAX	—	—	1.14 ± 0.04
LSTM	4.87 ± 0.17	1.26 ± 0.37	—
TFT	53.02 ± 0.14	26.35 ± 9.94	—
ETSformer	58.88 ± 0.20	19.60 ± 5.99	—

sequence length), but its training phase is more intensive as the ETSformer requires over twice the parameters. Finally, SARIMAX is the slowest model, as it must refit coefficients for every input sequence. In summary, the selection of the model depends on operational priorities: the TFT is ideal for mission-critical scenarios requiring both accuracy and interpretability. Conversely, LSTM offers a trade-off between accuracy and efficiency suited for resource-constrained edge computing. Finally, SARIMAX provides a simple solution when load profiles remain highly stable.

To build an understanding of the behavior of the models, the interval from 00:00 on Monday, 12 December 2022, to 23:00 on Sunday, 15 January 2023 (5 weeks, denoted as W1-W5) was selected to compare the predictions generated by all the tuned models with the observed targets. Figure 3 displays the targets of each week together with the forecasts for the subsequent week.

This period is expected to be difficult to forecast because it encompasses several regime transitions and festivities that require the models to adapt dynamically to abrupt variations in the distribution of the observed data across consecutive lookback windows. With the exception of particular cases, none of the models is able to capture festivities accurately. This limitation will be examined for the TFT in Section 4.3.3. SARIMAX operates as a weekly seasonal naïve model, where forecasts replicate the last observed value for the same day and hour in the previous week. As a result, it performs poorly during regime shifts or when sudden variations in the electricity consumption occur between corresponding days of consecutive weeks (e.g., all working days in W3 compared with W2, see Figure 3b). Accurate forecasts are obtained mainly on weekends or when demand patterns remain unchanged across weeks, confirming the preliminary findings in Table 2.

The ETSformer provides accurate forecasts not only when load curves are similar across consecutive weeks, but also during subtle regime shifts, as shown by the well-predicted drop in electricity use from the first days of W2 compared with W1. In contrast, predictions for W3, W4, and W5 are less accurate because these weeks belong to significantly different regimes from the preceding ones. Errors are largest in the first days of each new regime (e.g., Monday 26 December 2022 and Monday 02 January 2023), when the lookback window lacks relevant data, but tend to improve over subsequent days.

Although the LSTM lacks inherent interpretability tools, its performance indicates that its complexity is sufficient to capture rich latent representations of the lookback window features and map them to accurate predictions. Week-by-week analysis shows that in W2 the model easily handles subtle regime shifts, while in later weeks, it consistently detects more pronounced ones. Poor predictions mainly occur on the first day after a regime shift, but the model rapidly adjusts its outputs from the second day.

The TFT architecture is inherently the most flexible design as far as feature processing is concerned: It can handle each input channel separately, including deterministic information from the forecast horizon (which is not part of the inputs in the LSTM and ETSformer), and it can leverage its self-attention mechanism to identify temporal patterns. The TFT appears more capable of anticipating regime shifts than the other models. However, the wide spread of predicted conditional percentiles on the first day of the week indicates limited confidence in the load dynamics, as exemplified in Figure 4 for W4. This confidence improves noticeably from the second day of the new regime and remains stable for the rest of the week. Even when regime changes are not effectively anticipated, as between W1–W2 and W4–W5, a similar behavior is observed.

Overall, the primary benefit of the LSTM and the TFT over the other models lies in their capacity to rapidly adapt to regime changes. In the worst case, they tend to produce inaccurate forecasts for only 1 day rather than for several consecutive days, as observed with the SARIMAX and the ETSformer.

4.2 | Generalization Performance

To further assess the ability of the tuned models to generalize the knowledge acquired during training to unseen data, the respective sequences extraction techniques of trained models described in Section 3 were applied on the entire preprocessed dataset (comprising both the training and the test period) for two additional departments within the same university campus (Architecture and Economics) so as to retrieve sequences for model testing. Additionally, the sliding-window fit and forecast procedure described in the supplement for SARIMAX were extended to the new departments as well. Finally, starting from the entire preprocessed dataset for the Department of Agriculture, reference sequences for testing over the entire 4-year period were obtained. Table 5 reports the evaluation metrics for each model applied to the electrical load of the Departments of Architecture and Economics over the entire dataset, with the results for the Department of Agriculture in the same time frame that serve as a baseline.

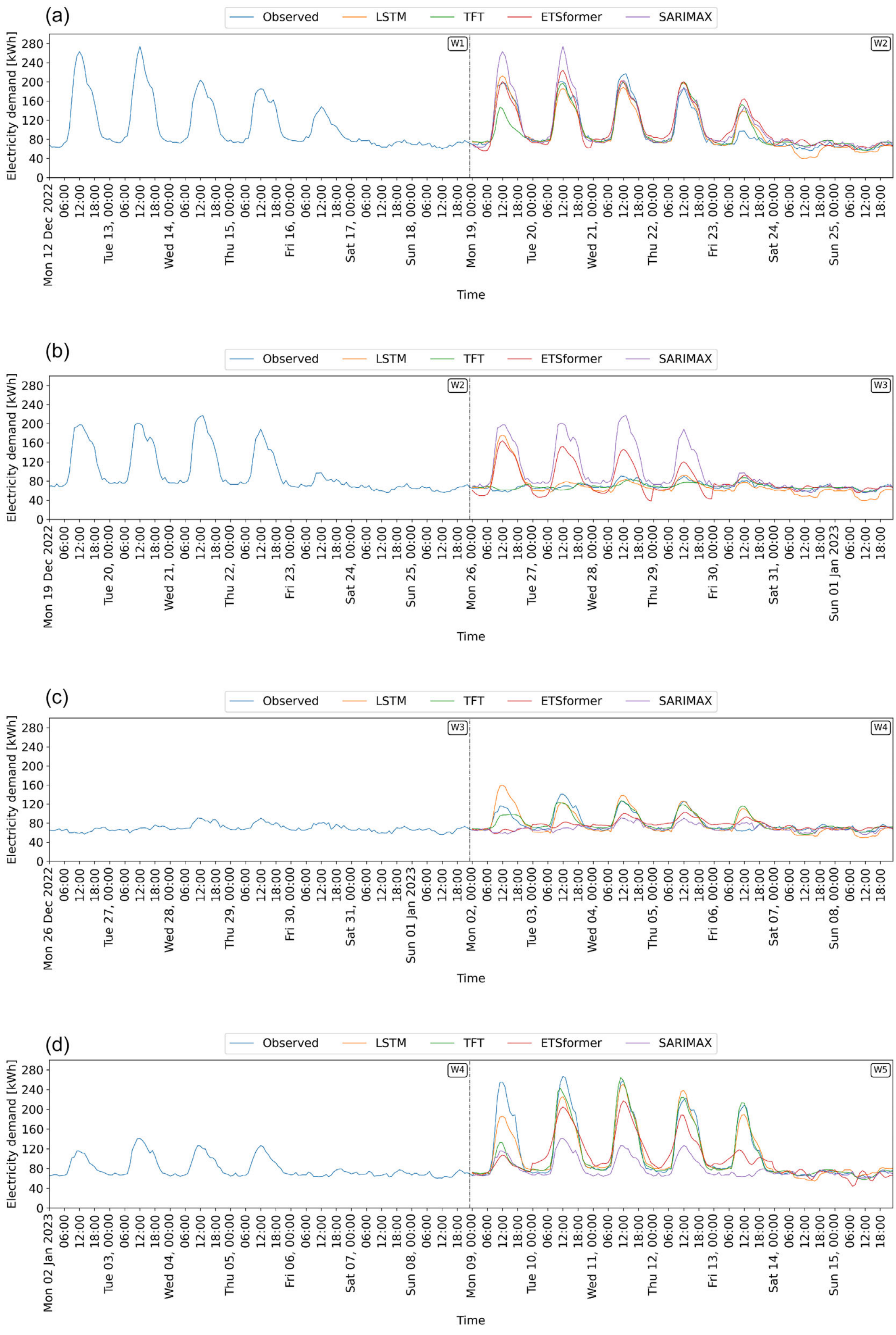


FIGURE 3 | Observed and forecasted electrical load during W2 (a), W3 (b), W4 (c), and W5 (d).

For the Department of Architecture, no single tuned model consistently surpasses the others across all performance metrics. The TFT achieves the smallest MAE and the smallest MAPE, representing the most suitable architecture for capturing the evolution of electricity demand within the Department of Architecture. In contrast, the LSTM attains the lowest MSE and therefore the highest R^2, but scores the highest MAE and MAPE. Since absolute errors usually scale with the level of the target and the LSTM rarely yields large regime-shift errors, a high MAE likely indicates poorer average predictions during high consumption periods. The ETSformer shows performance similar to the LSTM, with a comparable MAE (−0.48%) and MAPE (−5.13%) and only

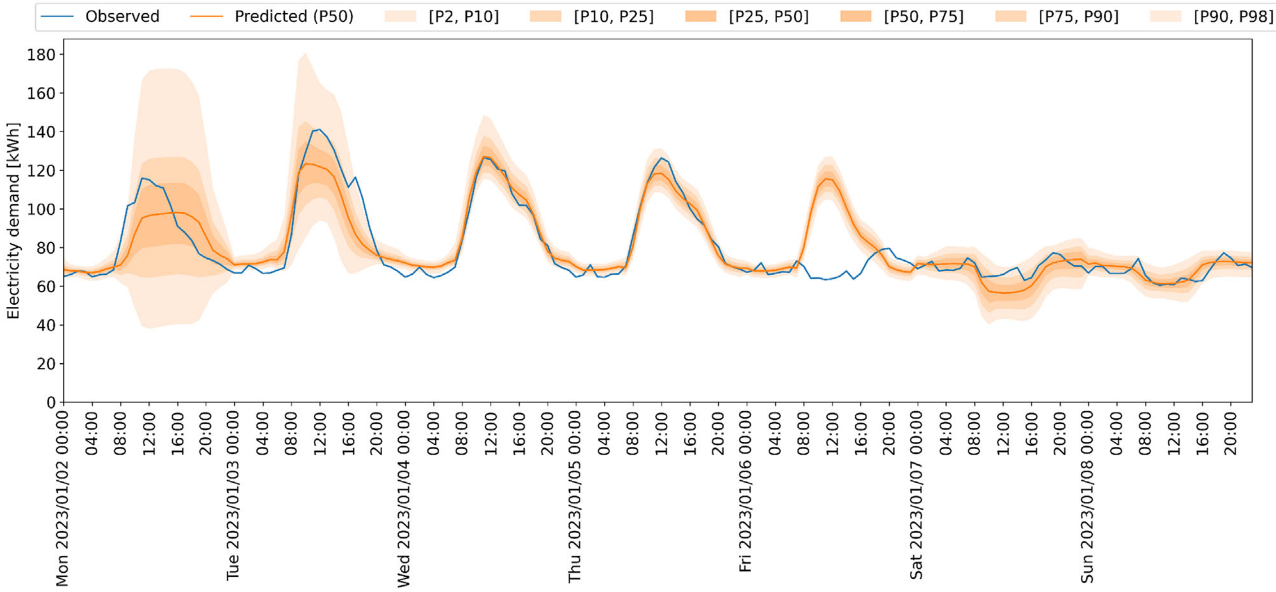


FIGURE 4 | Actual electricity consumption and conditional percentiles (indicated with the letter P) predicted by the TFT during W4.

TABLE 5 | Generalization performance of the pretrained models to the forecasting of the electrical load of the Architecture and the Economics Departments considering the whole dataset.

Department	Model	MAE, kWh	MSE, kWh^2	RMSE, kWh	MAPE, %	R^2, –
Agriculture	SARIMAX	14.18	646.54	25.43	12.45	0.779
	LSTM	9.90	273.72	16.54	9.27	0.906
	TFT	**7.00**	**176.70**	**13.29**	**6.39**	**0.940**
	ETSformer	11.68	352.12	18.76	11.43	0.880
Architecture	SARIMAX	24.95	2,399.60	48.99	38.55	0.603
	LSTM	25.05	**1,295.48**	**35.99**	49.52	**0.786**
	TFT	**18.04**	1,386.86	37.24	**25.99**	0.770
	ETSformer	24.93	1,555.07	39.43	46.98	0.743
Economics	SARIMAX	7.17	149.99	12.25	27.30	0.594
	LSTM	18.99	458.40	21.41	103.47	−0.242
	TFT	**4.90**	**80.00**	**8.94**	**18.46**	**0.783**
	ETSformer	12.66	251.71	15.87	61.55	0.318

Note: The best performance for each metric is in bold characters.

a slightly higher MSE (+20.04%). Since the SARIMAX attains a nearly equal MAE to LSTM (−0.40%) but with much larger MSE (+85.23%), this suggests that it struggles less during high consumption periods compared with the latter model (excluding usual difficulties during regime transitions). In contrast, during stable moderate consumption, the SARIMAX outperforms both ETSformer and LSTM, lowering MAPE by 22.15% relative to the latter.

In the case of the Economics Department, Table 5 displays a well-defined hierarchy among the models. The TFT consistently performs best across all evaluation metrics, the SARIMAX ranks second, followed by the ETSformer, while the LSTM yields the weakest results. This may be due to the substantial decrease in amplitude of the load profile of the Economics Department compared with the Agriculture one (see Figure S1).

Overall, the TFT consistently delivers the best results. The LSTM and ETSformer show a performance degradation in both generalization cases, with only a slight improvement over SARIMAX.

4.3 | Interpretability Analysis

Beyond quantitative performance metrics, it is essential to investigate how the models internally detect and represent the temporal patterns underlying the load time series. This section presents the interpretability study for the TFT. VSWs are first analyzed to assess the relative contribution of each input variable to model predictions. Subsequently, the Attention Patterns learned by the temporal attention layer are studied both over the whole test period and during significant periods such as holidays, where the Comaniciu distance metric is employed to quantify differences in attention distributions. All these analyses are carried out over the test dataset of the Agriculture department.

4.3.1 | *Variable Selection Weights of the TFT*

This section presents the interpretation of the VSWs obtained for all input sequences from the Agriculture test subset. For each feature in a data channel (lookback inputs for the encoder and future inputs for the decoder), the average of its VSWs across all timesteps and sequences (including the ones with some labels overlapped) is chosen as the aggregate metric to represent the importance of the same feature in the entire dataset, as shown in Figure 5.

The measured past load leads the ranking, while most of the meteorological information seems to be condensed in the solar irradiance. Regarding the seasonal indicators, both data channels appear to prioritize the day of the week over the month of the year. This behavior is expected, since meteorological variables implicitly capture monthly patterns: Their average evolution is independent of the day of the week, but their magnitude varies significantly across months.

To verify whether the cumulative importance curve in Figure 5 reflects the actual impact of features on performance, a feature ablation study was carried out. All input features except for seasonal indicators (as they are inexpensive to obtain) were progressively removed in ascending importance order and the resulting datasets were used to train the TFT. The MAPE obtained in each ablation case is shown in Figure 6. The removal of a given share of the cumulative importance (see Figure 5) does not lead to a proportional drop in performance, except when the electricity demand is removed (i.e., the number of features in the encoder is 2): In this instance, a sharp peak in MAPE occurs, as expected. This suggests that training a TFT model with only the seasonal indicators and the target variable as encoder inputs could yield results comparable to those of the original TFT, while requiring fewer computational resources.

4.3.2 | *Attention Patterns of the TFT*

Figure 7 reports the line chart of the multi-head attention weights (AWs, denoted by α) of all timesteps $n \in [-167, 24]$ for four representative forecast horizons τ, i.e., 6, 12, 18, and 24, averaged over sequences whose forecast horizon starts at midnight. The positional indices n and τ express the temporal distance from the forecast origin t ($n = 0$). The AWs for a given horizon indicate the relative contribution of each timestep to the latent representation used for forecasting that specific horizon. Periodic peaks in these patterns can thus be interpreted as indicators of the types of seasonality captured by the model on average. In this case, distinct daily-seasonal peaks emerge,

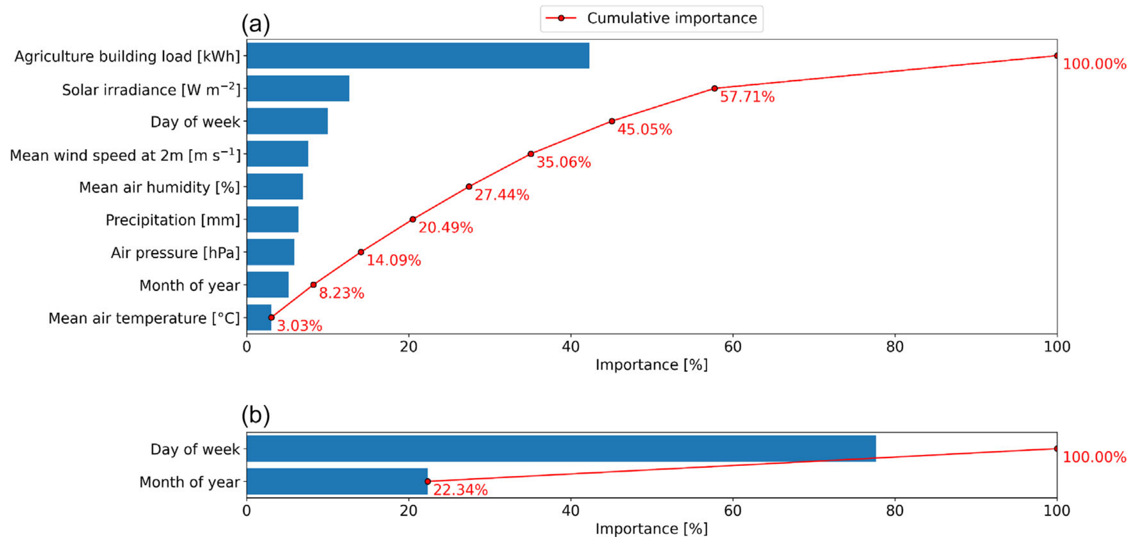


FIGURE 5 | Importance of the input features for the lookback input channel (a) and the future input channel (b) of the tuned TFT.

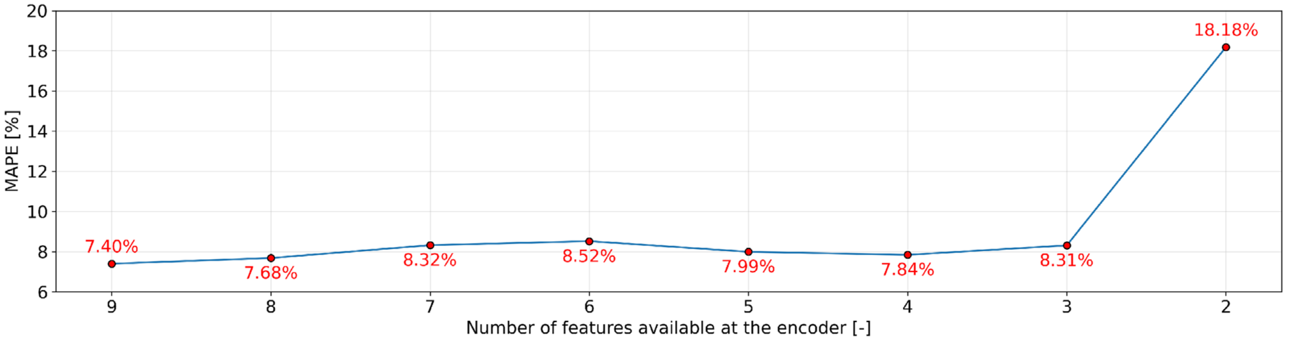


FIGURE 6 | TFT feature ablation study.

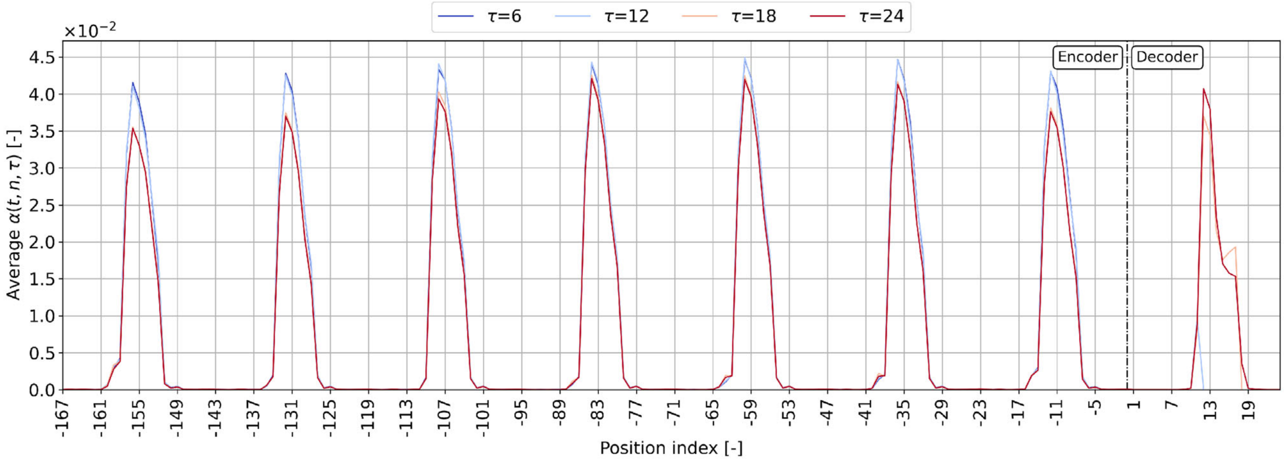


FIGURE 7 | Line chart of the AWs for four representative τ values, averaged over sequences whose first predicted timestep is midnight.

exhibiting a bell-shaped pattern peaking around 11:00–12:00. Interestingly, the position of these peaks remains unchanged as τ varies, i.e., the most attended hours are consistent across all horizons. These hours coincide with the typical daily load peak of the department (see Figure S1a), as well as with the daily solar irradiance peak, suggesting that attention primarily contributes to estimating the amplitude of this daily peak rather than encoding horizon-specific temporal information. It is therefore hypothesized that the differentiation of latent representations for each horizon in the output layer is mainly performed by the matrix $\mathbf{W}_\mathrm{H}$ downstream of the attention layer.

To further investigate this behavior, Figure 8 shows the same AW profiles aggregated by day of the week for a weekday (Monday,

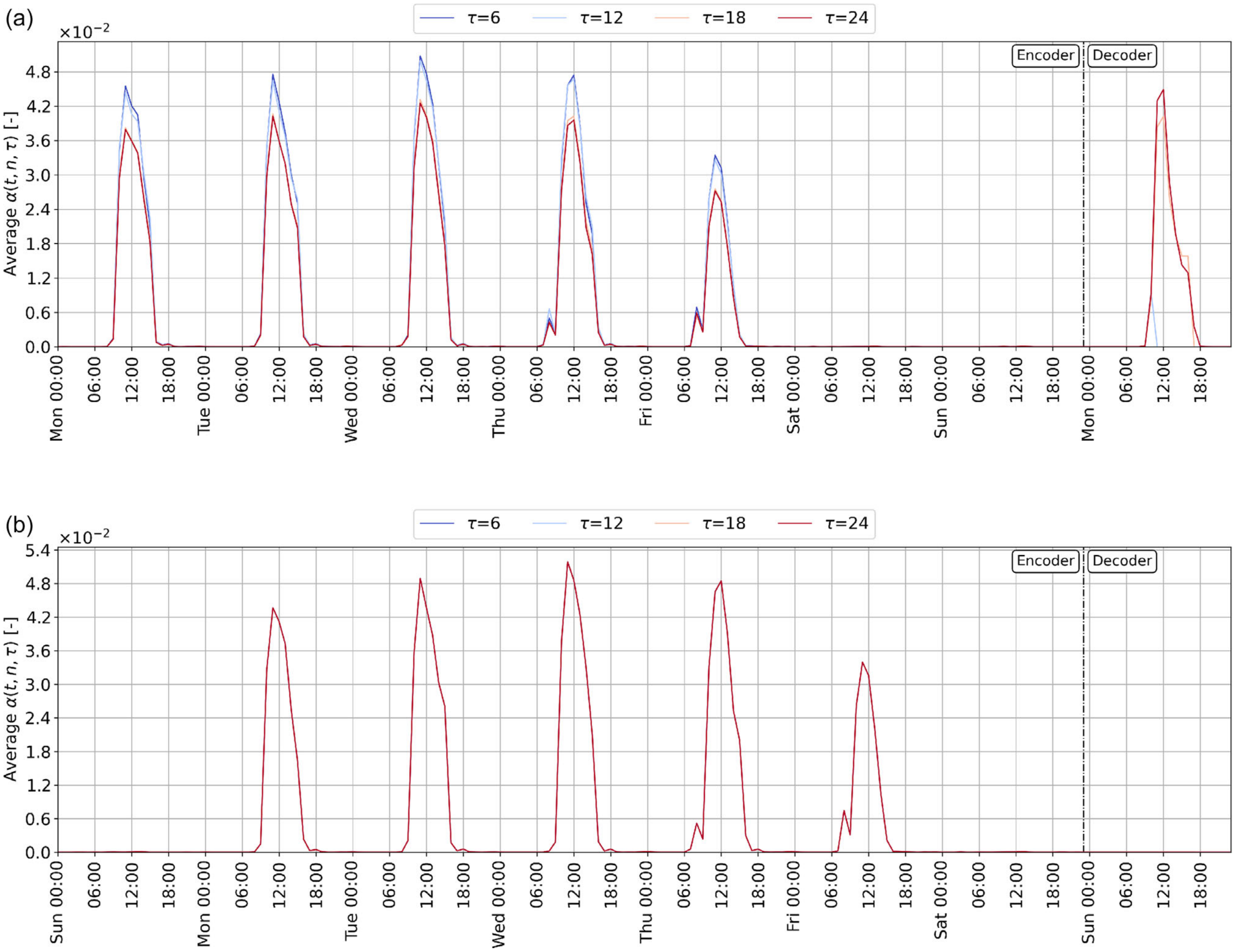


FIGURE 8 | Line chart of the AWs averaged by day of the week for four representative τ values, i.e., 6, 12, 18, and 24: (a) average AWs of a weekday (Monday) and (b) average AWs of a weekend day (Sunday).

Figure 8a) and a weekend day (Sunday, Figure 8b). The Monday chart confirms the bell-shaped, horizon-invariant attention pattern and suggests that attention gaps occur during baseline consumption hours; this naturally makes the model distinguish working from nonworking days. Friday tends to receive lower attention than other weekdays, possibly because its average load dynamics are different from the other weekdays (see Figure S1a). The Sunday plot corroborates the findings from the Monday plot even when a weekend day constitutes the forecast horizon. This behavior can be interpreted as an indication that the model leverages information from high-variance load and weather dynamics of typical weekdays to infer the baseline load conditions of the weekend, rather than relying directly on recent weekend observations. This interpretation reinforces the hypothesis that attention in this context serves primarily as a mechanism for identifying a globally informative temporal context rather than for capturing day-specific or horizon-specific dependencies, while subsequent layers are responsible for adjusting the prediction magnitude to the specific day type and horizon.

4.3.3 | *Attention Patterns of the TFT During Significant Periods and the Comaniciu Distance*

The selected seasonal indicators do not supply the model with information on festivities. Therefore, analyzing the performance of the model during these periods can provide interesting insights into the TFT learning mechanism.

Figure 9 presents the percentage relative differences between the yearly MAPE values and the MAPE values computed during national and regional holidays for the Agriculture Department, only considering the sequences whose forecast horizon starts at midnight. The model underperforms during Epiphany, Easter Monday, and Saint Rosalia across all years, whereas it consistently performs well on Easter. For the other holidays, the model shows a few notable drops in performance, with Republic Day, All Saints' Day, and Immaculate Conception emerging as the most challenging holidays. Most of the holidays where good performance occurs (e.g., Easter) fall during the weekend, as the load evolution in weekends is similar to the one during holidays. On the contrary, most of the holidays characterized by underperformance fall during weekdays (e.g., Easter Monday).

To demonstrate the benefit of a holiday indicator in assisting the model with identifying festive periods, a TFT variant that incorporates a binary indicator for holidays as an additional input feature was trained. This variant will be denoted as TFT-H. Figure 10 presents its performance during holidays, highlighting a substantial reduction in the magnitude of the MAPE relative differences. Additionally, out of 52 holidays in total, the number of holidays characterized by underperformance decreased to 15 (17 fewer than the baseline) and the number of holidays characterized by overperformance not on weekends nearly tripled. These results clearly show that the use of holiday indicators can improve model performance during festivities.

Another interpretability tool to assess whether the model can identify festive periods is the Comaniciu distance between attention patterns dist(t), as defined in Equation (4). Figure 11 illustrates the evolution of dist(t) for each sequence whose forecast horizon starts at midnight from the test dataset, together with the observed electrical load. The 85th percentile (P85) of dist(t) (i.e., 0.51) appears to represent a natural threshold to distinguish periods characterized by atypical attention patterns. Visually, the periods where dist(t) > P85 largely overlap with special periods within the academic calendar of the university, such as the Summer Break 2022 (July 18–August 31, 2022), the First Semester Midterm Exam Session 2022 (November 7–November 11, 2022), the Winter Break 2022 (December 24, 2022–January 8, 2023), and the Second Semester Midterm Exam Session 2023 (April 11–19, 2023), as well as national holidays.

To further investigate how the attention pattern varies during this atypical regime, Figure 12 displays the observed load, the predicted load, and the average attention pattern for $\tau = 24$ of the sequence with the highest dist(t). This week contains several anomalous days due to the Christmas festivities. Different from the reference pattern shown in Figure 7, attention is concentrated almost exclusively on the few non-base-load hours of the days that deviate from the baseline (namely, December 22,

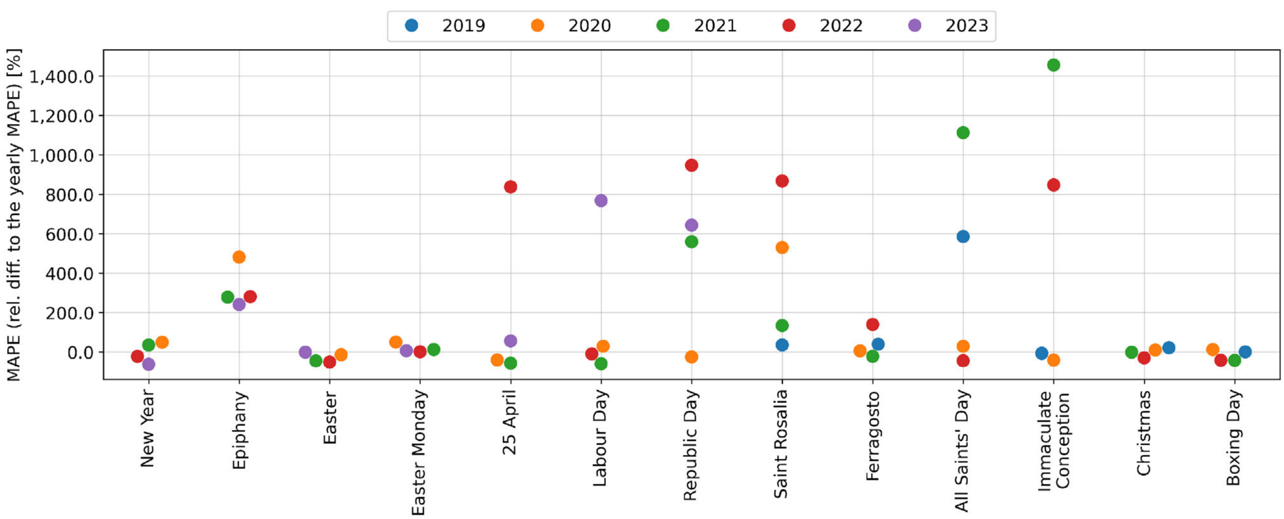


FIGURE 9 | Performance of the tuned TFT during festivities with year stratification. Swarmplots were employed to ensure complete readability of the results; rel. diff. stands for relative difference.

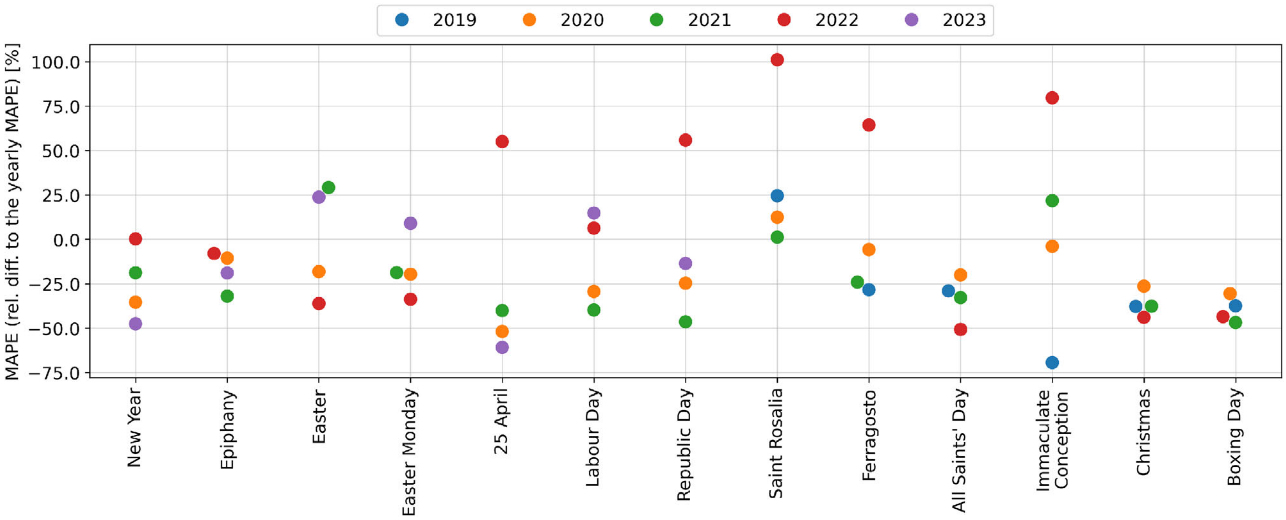


FIGURE 10 | Performance of TFT-H during festivities with year stratification; rel. diff. stands for relative difference.

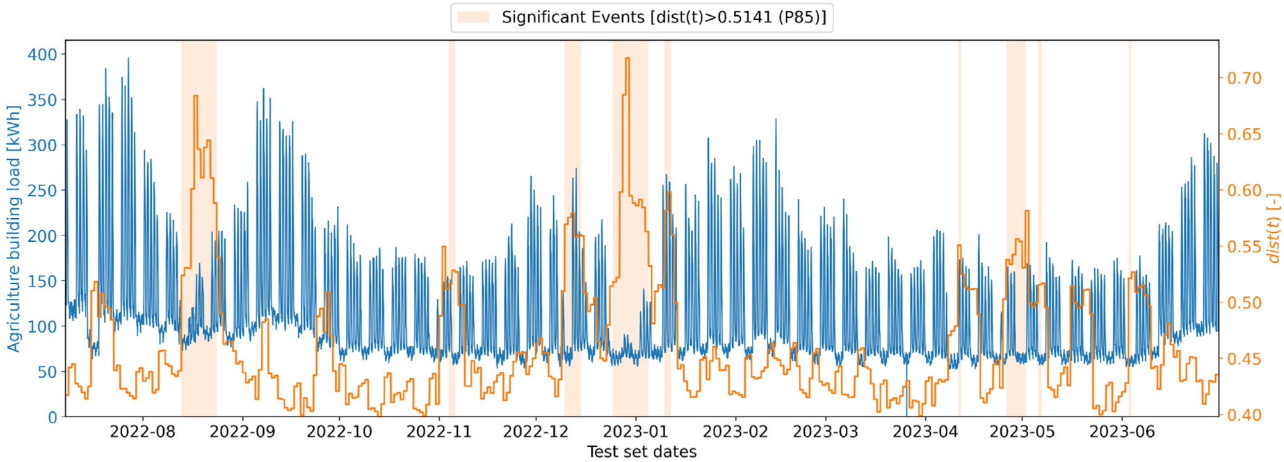


FIGURE 11 | Comaniciu distance between attention patterns for each sequence whose forecast horizon starts at midnight, together with the observed electrical load.

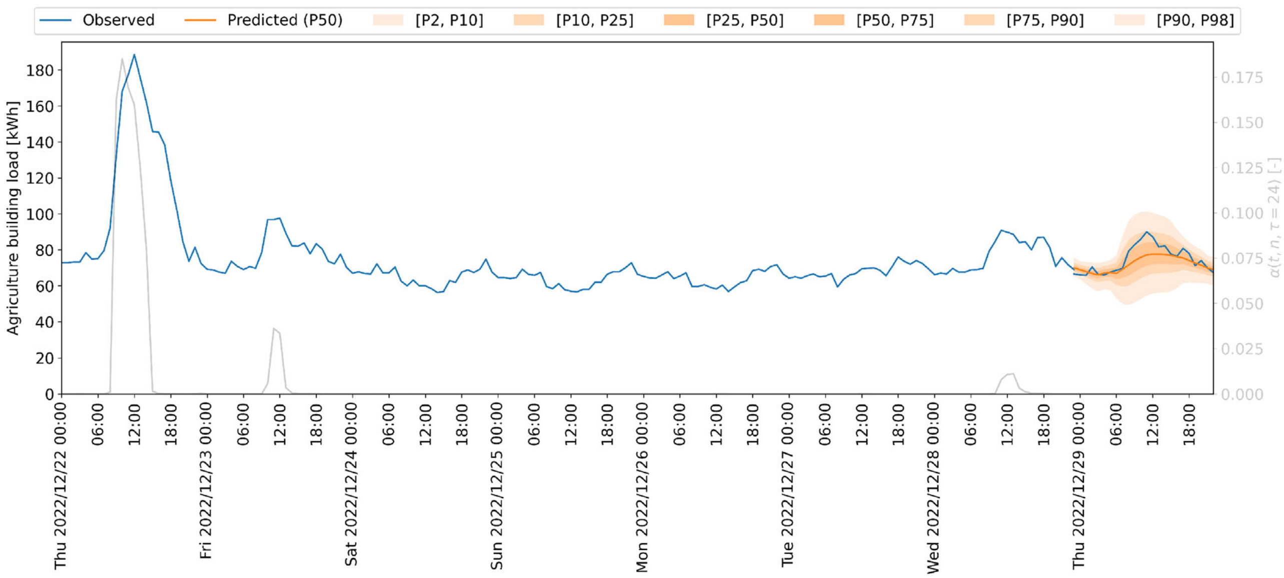


FIGURE 12 | The observed load, the predicted load, and the average attention pattern for $\tau = 24$ of the sequence with the highest dist(t).

23, and 28), while the remaining timesteps receive nearly zero attention, explaining the higher dist(t). Overall, these results suggest that the model dynamically adjusts its AWs to adapt to regime shifts in the target variable and the dist(t) metric may effectively help identify significant events.

5 | Conclusion

STLF is essential in energy management to plan and optimize operational strategies for multi-energy systems. Currently, DL-based models are used to develop more accurate and efficient STLF solutions. However, they often function as black boxes. In this work, a TFT model was applied to the real STLF case study of a department building at an Italian university. The model was compared with a statistical model benchmark (SARIMAX), a conventional RNN model (LSTM), and an alternative Transformer-based model (ETSformer).

The TFT achieves superior forecasting performance overall, exhibiting a MAPE equal to 7.40% over the test dataset, with an improvement of 25.03%, 45.02%, and 46.02% compared to LSTM, SARIMAX, and ETSformer, respectively. It is followed by the LSTM model, which yields a MAPE equal to 9.87%. Although the TFT attains slightly higher accuracy, this improvement comes at the cost of substantially greater architectural complexity: it requires 709% more parameters than the LSTM, which achieves comparable performance at a fraction of the computational cost. The latter therefore offers the best trade-off between performance and computational cost.

A preliminary analysis of the generalizability of the models was carried out by evaluating their performance on unseen data from two different department buildings within the same university campus: The TFT consistently delivers the best results (with a MAPE of 25.99% over the Architecture Department dataset and 18.46% over the Economics Department dataset). On the other hand, none of the benchmark models appears reliable for this same task, with the performance of trained benchmark models even becoming worse than that of SARIMAX.

Then, a comprehensive interpretability analysis of the TFT was carried out. First, VSWs were analyzed, showing the significance of temporal variables. Next, the study of the AWs provides insights into the ability of the model to detect both short-term and periodic patterns. Finally, the effectiveness of the Comaniciu distance metric between attention patterns for detecting anomalous events was evaluated, examining the forecasting capabilities during national and regional holidays.

To conclude, this study confirms that LSTM represents an effective state-of-the-art approach for STLF, offering a favorable balance between performance and computational efficiency due to its simple and well-established architecture. However, its limited generalizability and interpretability constrain its broader applicability. In contrast, the TFT provides enhanced interpretability and promising adaptability, albeit at the cost of greater model complexity and computational demand. Therefore, this work establishes a foundation for applying the TFT to STLF within a broader energy management framework, particularly in scenarios where interpretability is critical for ensuring safe and reliable operation of systems such as microgrids.

A limitation of this work is that these findings are bound to a specific geographic location and institutional context. Future efforts will aim to validate the models across diverse building types, climates, and operational contexts. Future work will also focus on enriching the data representation by integrating information on known special events that influence energy consumption patterns. The workflow and findings of this study will be validated against higher resolution data (e.g., quarter-hourly data). Finally, the model will be integrated into the energy management system of the university.

Acknowledgments

This publication is part of the project PNRR-NGEU which has received funding from the MUR - DM 352/2022 (38-033-32-DOT1332092-3353). This research was also co-funded by Trigenia Srl. Dr. Randazzo acknowledges funding from the research contract no. 32-G-13427-2 (DM 1062/2021) funded within the Programma Operativo Nazionale (PON) Ricerca e Innovazione of the Italian Ministry of University and Research.

Funding

This study was supported by the Ministero dell'Università e della Ricerca (38-033-32-DOT1332092-3353 and 32-G-13427-2).

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

Restrictions apply to the data that support the findings of this study due to strict institutional intellectual property regulations. Data are available from the corresponding authors upon reasonable request.

References

1. G. Heinemann, D. Nordmian, and E. Plant, "The Relationship Between Summer Weather and Summer Loads - A Regression Analysis," *IEEE Transactions on Power Apparatus and Systems* 11 (1966): 1144.

2. M. Shahid, K. Ullah, K. Imran, A. Mahmood, and M. Arentsen, "LEAP Simulated Economic Evaluation of Sustainable Scenarios to Fulfill the Regional Electricity Demand in Pakistan," *Sustainable Energy Technologies and Assessments* 46 (2021): 101292.

3. B. Biswal, S. Deb, S. Datta, T. S. Ustun, and U. Cali, "Review on Smart Grid Load Forecasting for Smart Energy Management Using Machine Learning and Deep Learning Techniques," *Energy Reports* 12 (2024): 3654.

4. R. Ahmed, V. Sreeram, Y. Mishra, and M. Arif, "A Review and Evaluation of the State-of-the-Art in PV Solar Power Forecasting: Techniques and Optimization," *Renewable and Sustainable Energy Reviews* 124 (2020): 109792.

5. B. F. Hobbs, S. Jitprapaikulsarn, S. Konda, V. Chankong, K. A. Loparo, and D. J. Maratukulam, "Analysis of the Value for Unit Commitment of Improved Load Forecasts," *IEEE Transactions on Power Systems* 14, no. 4 (1999): 1342.

6. A. Sieminski, International Energy Outlook, (U.S. Energy Information Administration (EIA), 2014), Vol. 18, 2.

7. T. Hussain, F. U. M. Ullah, K. Muhammad, et al., "Smart and Intelligent Energy Monitoring Systems: A Comprehensive Literature Survey and Future Research Guidelines," *International Journal of Energy Research* 45, no. 3 (2021): 3590.

8. Y. Eren and I. Küçükdemiral, “A Comprehensive Review on Deep Learning Approaches for Short-Term Load Forecasting,” *Renewable and Sustainable Energy Reviews* 189 (2024): 114031.

9. L. Baur, K. Ditschuneit, M. Schambach, C. Kaymakci, T. Wollmann, and A. Sauer, “Explainability and Interpretability in Electric Load Forecasting Using Machine Learning Techniques – A Review,” *Energy and AI* 16 (2024): 100358.

10. B. Lim, S. O. Arık, N. Loeff, and T. Pfister, “Temporal Fusion Transformers for Interpretable Multi-Horizon Time Series Forecasting,” *International Journal of Forecasting* 37, no. 4 (2021): 1748.

11. G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. C. H. Hoi, “Etsformer: Exponential Smoothing Transformers for Time-Series Forecasting,” arXiv preprint 2022, arXiv:2202.01381.

12. H. Zhang, Y. Zou, X. Yang, and H. Yang, “A Temporal Fusion Transformer for Short-Term Freeway Traffic Speed Multistep Prediction,” *Neurocomputing* 500 (2022): 329.

13. X. Liao, M. S. Wong, and R. Zhu, “Dual-Gate Temporal Fusion Transformer for Estimating Large-Scale Land Surface Solar Irradiation,” *Renewable and Sustainable Energy Reviews* 214 (2025): 115510.

14. M. L. Santos, X. García-Santiago, F. E. Camarero, G. B. Gil, and P. C. Ortega, “Application of Temporal Fusion Transformer for Day-Ahead PV Power Forecasting,” *Energies* 15 (2022): 14.

15. Q. Liu, L. Zhang, C. Lv, H. Gao, Y. Cai, and L. Chen, “New Perspective of Environmental Impact Research: Predicting Bus Exhaust Emissions Using the ETSformer Based on Collaborative Perception,” *Sustainable Horizons* 11 (2024): 100105.

16. M. Hasan, Z. Mifta, S. J. Papiya, et al., “A State-of-the-Art Comparative Review of Load Forecasting Methods: Characteristics, Perspectives, and Applications,” *Energy Conversion and Management: X* 26 (2025): 100922.

17. L. Alhmoud and Q. Nawafleh, “Short-Term Load Forecasting for Jordan Power System Based on NARX-ELMAN Neural Network and ARMA Model,” *IEEE Canadian Journal of Electrical and Computer Engineering* 44, no. 3 (2021): 356.

18. K. Ullah, M. Ahsan, S. M. Hasanat, et al., “Short-Term Load Forecasting: A Comprehensive Review and Simulation Study With CNN-LSTM Hybrids Approach,” *IEEE Access* 12 (2024): 111858.

19. J. Moon, S. Park, S. Rho, and E. Hwang, *Computational Intelligence and Neuroscience* 2022 (2022): 6892995.

20. H. M. Shakeel, S. Iram, H. M. A. Farid, and R. Hill, “Linear and Tree-Based Intelligent Investigation of Cross-Domain Housing Features to Enhance Energy Efficiency,” *Advanced Intelligent Systems* 7, no. 8 (2025): 2400939.

21. A. Bracale, P. Caramia, P. De Falco, and T. Hong, “A Multivariate Approach to Probabilistic Industrial Load Forecasting,” *Electric Power Systems Research* 187 (2020): 106430.

22. J. Moon, S. Jung, J. Rew, S. Rho, and E. Hwang, “Combination of Short-Term Load Forecasting Models Based on a Stacking Ensemble Approach,” *Energy and Buildings* 216 (2020): 109921.

23. R. Chandrasekaran and S. K. Paramasivan, “Advances in Deep Learning Techniques for Short-term Energy Load Forecasting Applications: A Review,” *Archives of Computational Methods in Engineering*, 32, no. 2 (2025): 663.

24. B.-S. Kwon, R.-J. Park, and K.-B. Song, “Short-Term Load Forecasting Based on Deep Neural Networks Using LSTM Layer,” *Journal of Electrical Engineering & Technology* 1501 (2020): 15.

25. D. Dylewsky, D. Barajas-Solano, T. Ma, A. M. Tartakovsky, and J. N. Kutz, “Stochastically Forced Ensemble Dynamic Mode Decomposition for Forecasting and Analysis of Near-Periodic Systems,” *IEEE Access* 10 (2022): 33440.

26. D. M. Bui, P. D. Le, T. M. Cao, H. Nguyen, T. T. Pham, and D. A. Pham, “A Statistical Data-Filtering Method Proposed for Short-Term Load Forecasting Models,” *Journal of Electrical Engineering & Technology* 15 (2020): 1947.

27. H. Eskandari, M. Imani, and M. P. Moghaddam, “Convolutional and Recurrent Neural Network Based Model for Short-Term Load Forecasting,” *Electric Power Systems Research* 195 (2021): 107173.

28. Y. Liu, S. Dutta, A. W. K. Kong, and C. K. Yeo, “An Image Inpainting Approach to Short-Term Load Forecasting,” *IEEE Transactions on Power Systems* 38, no. 1 (2023): 177.

29. T. Bashir, C. Haoyong, M. F. Tahir, and Z. Liqiang, “Short Term Electricity Load Forecasting Using Hybrid Prophet-LSTM Model Optimized by BPNN,” *Energy Reports* 8 (2022): 1678.

30. L.-L. Peng, X.-Y. Yang, X.-X. Hua, et al., “Short-Term Electric Load Forecasting Based on Deep Learning and Multi-Modal Fusion,” *Electric Power Systems Research* 254 (2026): 112593.

31. H. T. Choi, H. Y. Park, T. Kim, and J. H. Kim, “A Novel Anomaly Forecasting in Time-Series Data: Feedback Connection between Forecasting and Detecting Algorithms with Applications to Power Systems,” *Advanced Intelligent Systems* 7, no. 5 (2025): 2401141.

32. K. Wu, J. Gu, L. Meng, H. Wen, and J. Ma, ”An Explainable Framework for Load Forecasting of a Regional Integrated Energy System Based on Coupled Features and Multi-task Learning,” *Protection and Control of Modern Power Systems* 7, no. 2 (2022): 1.

33. A. Vaswani, N. Shazeer, N. Parmar, et al., *Advances in Neural Information Processing Systems*, 30 (Curran Associates, Inc, 2017).

34. X. Wang, P. Wang, Y. Song, Q. Xiang, and J. Li, “Self-Regulated Self-Healing Robotic Gripper for Resilient and Adaptive Grasping,” *Advanced Intelligent Systems* 5, no. 11 (2023): 2300286.

35. W. Huang, N. Fan, W. Wang, J. Wang, X. Qi, and S. Zhang, “Improving Long-Term Glucose Prediction Accuracy with Uncertainty-Estimated ProbSparse-Transformer,” *Advanced Intelligent Systems* 7, no. 12 (2025): e202500235.

36. M. Zhang, Z. Yu, and Z. Xu, “Short-Term Load Forecasting Using Recurrent Neural Networks With Input Attention Mechanism and Hidden Connection Mechanism,” *IEEE Access* 8 (2020): 186514.

37. H. Zhou, S. Zhang, J. Peng, et al., “Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting,” *Proceedings of the AAAI Conference on Artificial Intelligence* 35, no. 12 (2021): 11106,

38. P. C. Huy, N. Q. Minh, N. D. Tien, and T. T. Q. Anh, “Short-Term Electricity Load Forecasting Based on Temporal Fusion Transformer Model,” *IEEE Access* 10 (2022): 106296.

39. S. Pelekis, I.-K. Seisopoulos, E. Spiliotis, et al., “A Comparative Assessment of Deep Learning Models for Day-Ahead Load Forecasting: Investigating Key Accuracy Drivers,” *Sustainable Energy, Grids and Networks* 36 (2023): 101171.

40. D. Li, Y. Tan, Y. Zhang, S. Miao, and S. He, “Probabilistic Forecasting Method for Mid-Term Hourly Load Time Series Based on an Improved Temporal Fusion Transformer Model,” *International Journal of Electrical Power & Energy Systems* 146 (2023): 108743.

41. N. Amin, Y.-W. Kim, C. Kang, and Y.-C. Byun, “Electricity Load Forecasting Using Hybrid Prophet–TFT Model Incorporating Weather Data,” *Energy Reports* 15 (2026): 108846.

42. A. B. Ferreira, J. B. Leite, and D. H. Salvadeo, “Power Substation Load Forecasting Using Interpretable Transformer-Based Temporal Fusion Neural Networks,” *Electric Power Systems Research* 238 (2025): 111169.

43. M. L. Santos, S. D. García, X. García-Santiago, et al., “Deep Learning and Transfer Learning Techniques Applied to Short-Term Load Forecasting of Data-Poor Buildings in Local Energy Communities,” *Energy and Buildings* 292 (2023): 113164.

44. A. Mohammadi, G. Nápoles, and Y. Salgueiro, “Deep Learning for Time-Series Forecasting With Exogenous Variables in Energy Consumption: A Performance and Interpretability Analysis,” *IEEE Access* 13 (2025): 86746.

45. Z. Wang, Z. Zhu, G. Xiao, B. Bai, and Y. Zhang, “A Transformer-Based Multi-Entity Load Forecasting Method for Integrated Energy Systems,” *Frontiers in Energy Research* 10 (2022): 2022.

46. D. Comaniciu, V. Ramesh, and P. Meer, “*Kernel-based object tracking*,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25, no. 5 (2003): 564.

47. T. Kailath, “The Divergence and Bhattacharyya Distance Measures in Signal Selection,” *IEEE Transactions on Communication Technology* 15, no. 1 (1967): 52.

48. G. Ghione, M. A. Judge, M. Badami, E. Pasero, V. Franzitta, and G. Cirrincione, 2024 IEEE. 22nd Mediterranean Electrotechnical Conference (MELECON), (IEEE, 2024), 662–667.

49. J. Beitner, “PyTorch Forecasting: Time Series Forecasting with PyTorch, Version: 1.5.0,” Accessed July 4, 2026, https://github.com/jdb78/pytorch-forecasting.

50. S. Seabold and J. Perktold, “Proceedings of the 9th Python in Science Conference,” Statsmodels: Econometric and Statistical Modeling with Python (2010).

51. TensorFlow Developers, “TensorFlow (Zenodo, 2021).” aufgerufen am 17, 2023, https://www.tensorflow.org/.

52. Z.-Y. Yang, X. Cao, R.-Z. Xu, W.-C. Hong, and S.-L. Sun, “Applications of chaotic quantum adaptive satin bower bird optimizer algorithm in berth-tugboat-quay crane allocation optimization,” *Expert Systems with Applications* 237 (2024): 121471.

Supporting Information

Additional supporting information can be found online in the Supporting Information section.