



**CENTRO UNIVERSITARIO
DE LA DEFENSA
EN LA ESCUELA NAVAL MILITAR**



This document is the Accepted Manuscript version of a Published Work that appeared in final form in *Energy and Buildings*, copyright © 2023 Elsevier Ltd, after peer review and technical editing by the publisher. To access the final edited and published work see: <https://doi.org/10.1016/j.enbuild.2023.113164>

Copyright © 2023 Elsevier Ltd

Deep learning and transfer learning techniques applied to short-term load forecasting of data-poor buildings in local energy communities

Miguel López Santos ^a, Saúl Díaz García ^a, Xela García-Santiago ^a, Ana Ogando-Martínez ^a, Fernando Echevarría Camarero ^a, Gonzalo Blázquez Gil ^a and Pablo Carrasco Ortega ^{a*}

^a Galicia Institute of Technology (ITG), A Coruña, Spain;

* Corresponding author: pcarrasco@itg.es; Tel.: +34-981-173-206

Abstract

The use of deep learning for electrical demand forecasting has shown great potential in generating accurate results, but requires a large amount of data to train the models. However, the limited availability of electricity consumption information for new users or recently monitoring buildings in local energy communities make it difficult to achieve these good results. To address this challenge, this research study proposes a novel methodology based on the combination of the transfer learning (TL) concept and the temporal fusion transformer (TFT) architecture. This is an innovative approach used to transfer knowledge from a general pre-trained model based on the historical data of the community's buildings to new users, reducing the needs for large amounts of data for each new building. The results show that TFT provides more accurate load estimates than other state-of-the-art methods by reducing RMSE by more than 11%. Moreover, the TL approach improves the prediction of load demand of buildings with limited availability of historical data, reducing CV_RMSE, SMAPE, and WQLoss by more than 40% compared to models that do not use knowledge transfer. This is a significant improvement that facilitates the incorporation of new users to the community achieving accurate load estimations even with lack of data. Moreover, this approach leads to an optimized and cost-efficient planning and management of energy distribution in local energy communities

Abbreviations: AI, artificial intelligence; ANN, artificial neural network; ARIMA, auto regressive integrated moving average; ARMA, autoregressive moving average; CNN, convolution neural network; CV_RMSE, coefficient of variance of the root mean square error; DL, deep learning; DNN, deep neural network; DR, demand respond; KNN, K-Nearest Neighbors; LSTM, long short-term memory; MAR, mean absolute error; ML, machine learning; MLP, multilayer perceptron; MLR, multiple linear regression; NNI, neural network intelligence; RMSE, root mean square error; RNN, recurrent neural network; SMAPE, mean absolute percentage error; SVM, support vector machine; TFT, temporal fusion transformer; TL, transfer learning; WQLoss, weighted quantile loss.

Keywords

Short-term load forecasting; Limited data availability; Local energy communities; TFT; Transfer learning.

1. Introduction

In line with the Paris Agreement, the European Union has set a goal to reduce its greenhouse gas emissions to be climate neutral by 2050. To achieve this goal, the energy system will undergo a profound transformation, characterized by decentralization, electrification and large-scale deployment of renewable energies. This transition will require an intelligent and flexible system managed through digitalization and based on customer participation, interconnectivity, storage, and demand response (DR) [1]. According to recent European regulation [2], all consumers will be able to consume, store and/or sell self-generated electricity and participate in all electricity markets, providing flexibility to the system, for example, through energy storage, demand response or energy efficiency systems.

Technological innovations will contribute to system's flexibility and the massive deployment of renewable energies and will enable the entry of new market participants such as prosumers and aggregators [3]. Prosumers, active energy consumers, are changing the dynamics of the sector, making it possible to manage demand by the end user. Thanks to digitization, aggregators, actors representing multiple clients in the market, will be able to manage large volumes of data and optimize systems with multiple small units.

Due to the variability of renewable energies, the stochastic nature of demand and the need to integrate different energy vectors, the energy system of the future will be characterized by its great flexibility. Thanks to new technologies, consumers will be able to monitor and control their energy assets in real time and respond automatically to the variable needs of the grid. Artificial intelligence (AI) has become a key technology used to address different challenges such as load forecasting, price forecasting, and energy asset planning and control, among others [4].

Electricity demand forecasting has become essential for planning and allocating available resources. This has gained special relevance due to the growing use of renewable energies, whose intermittent generation hinders the balance between electricity supply and demand [5]. It is also a significant resource for the proper operation of DR programs, which use load forecasting data to stimulate changes in users' electricity consumption through financial incentives [6].

In the field of local energy communities and markets, load forecasting is necessary to program the energy consumption of the community, to control flexible systems, such as controllable loads and batteries, and to enable prosumers to participate in peer-to-peer energy trading and flexibility markets [7]. Moreover, it also facilitates decision-making

related to energy management and enables demand control through strategies such as peak load reduction or shifting and scheduling energy use [8].

Load forecasting is a challenge because it is influenced by multiple factors such as weather, consumption habits of users, characteristics of the building, etc. [8]. In the context of smart buildings, given the large number of installed sensors and monitoring data available, data-driven forecasting models are the most widely used nowadays [5,9,10]. Data-driven forecasting techniques can be classified into statistical models [11–17] and artificial intelligence-based models [8,18–24], although hybrid models are also relevant in the literature [25–27]. Many authors have used statistical models in forecasting. Some examples are regression models [11,12], the auto regressive integrated moving average (ARIMA) model [13,14], the autoregressive moving average (ARMA) model [13,15] or exponential smoothing methods [15,16]. Amber et al. [12] predicted the daily electricity consumption of an administration building in the UK through a multiple linear regression (MLR) model and a genetic programming model. Chujai et al. [13] compared ARIMA and ARMA in their research, concluding that the ARIMA model was the best approach to forecast monthly electricity consumption of a building whereas ARMA was more suitable for weekly and daily period. Statistical models are classical methods and provide excellent results for linear problems. However, for nonlinear problems, the use of other methods such as artificial intelligence improves the accuracy of data-driven models by providing the ability to recognize complex relationships and patterns in the data. Over the last decade, the use of different machine learning (ML) techniques in load forecasting such as decision trees, neural networks, random forest, K-Nearest Neighbors (KNN) or support vector machines (SVM) have become widespread due to their ability to deal with non-linear variations in the electrical load [5,9]. Walker et al. [18] evaluated different ML algorithms for predicting hourly electricity demand in 47 commercial buildings, concluding that boosted-tree, random forest, and artificial neural network (ANN) provide the best outcomes. In the study conducted by Dagdougui [8], several ANN architectures and different input data sets were analyzed. However, deep learning (DL) methods are the most prominent in the recent literature. A variety of deep neural networks (DNN), including recurrent neural networks (RNNs) and convolution neural networks (CNNs), have been applied in numerous studies related to load forecasting [19–23]. The study by Rahman et al. [19] showed the potential of RNN models in predicting electricity consumption over medium and long-term time horizons. Cai et al. [20] evaluated the performance of RNNs and CNNs against traditional methods in commercial buildings. Chitalia et al. [21] compared 9 different models of DL for day-ahead

and hour-ahead load forecasting. These studies converge that long short-term memory (LSTM) networks, a type of RNN with the capability to learn long-term dependencies, are the most accurate for forecasting electricity demand.

This research aims to validate the use of a more complex DL architecture for short-term load forecasting, the temporal fusion transformer (TFT). TFT [28] is an attention-based deep neural network that can predict the behavior of a target variable at different future points in time. This network also includes LSTM within its architecture. TFT stands out in terms of interpretability due to the use of specialized components to identify temporal patterns, significant events and variables of major importance. This method uses a temporal self-attention decoder for learning long-term dependencies, static covariate encoders for contextual learning, a sequence-to-sequence layer for processing known inputs, and dependent variable selection mechanisms to mitigate irrelevant contributions.

The use of DL techniques requires a considerable amount of historical data, which is not always possible in newly constructed or recently monitored buildings [29,30]. In this work, the transfer learning (TL) techniques are used to solve this issue, using data from similar buildings in the training process. TL aims to take the knowledge gained on one task and apply it to a different task (e.g., similarities between the time series data of houses A and B enable transfer learning). This learning mechanism has been adopted in multiple research areas and problems such as image classification [31], voice processing [32], object detection [33], speech and language processing [34] or text sentiment analysis [35]. In recent years, the use of TL has been extended to the building electricity sector to solve problems for which limited data is available [36–41]. Jung et al. [36] developed a monthly electric load forecasting method based on knowledge transfer. Zeng et al. [37] used TL to predict daily load during the holidays. The study conducted by Wu [38] shows that for different forecasting models, the error can be significantly reduced by using knowledge learned from other residential dwellings.

This paper addresses the forecasting of energy demand in buildings with lack of data to improve the energy management of local energy communities. To achieve this goal, the TL concept and the TFT architecture are combined to forecast building hourly electricity consumption in buildings with limited availability of historical data. The application of both techniques is a novelty in this field of study since nothing similar has been found in the literature. The pre-training is done using data from existing buildings within the community, allowing knowledge to be transferred to new buildings, which reduces the training time and improves the accuracy of the results. The main objective of this methodology is to facilitate the management of new users to the community by eliminating the need

for long time data history. This study also analyzes different TL strategies to identify the optimal one for the developed methodology and compares the TFT algorithm with other state-of-the-art methods.

2. Methodology

In this work, a forecasting methodology is proposed to estimate the hourly day-ahead consumption of buildings belonging to the same local energy community using historical load data and weather and calendar data. The methodology of TL is applied following these steps: (1) Preprocess data to transform raw data into the appropriate form for the models. (2) Select source task, which involves selecting a related predictive modelling problem with abundance of data. (3) Develop source model. (4) Reuse source model in target domain following the selected TL strategies. (5) Evaluate the results. TFT models were trained using the implementation in Pytorch Forecasting.

2.1. Data transformation

Data transformations are applied to calendar and continuous variables. Cyclical calendar features are transformed to achieve sequential data. For example, December and January are separated by 1 month, even though they will appear to be 11 months apart. This is avoided by deriving the cosine and the sine of the initial calendar variable.

In addition, continuous variables are normalized to avoid different scales and ensure every data point has the same importance. In this case, standardization was used to rescale the data. Using this technique, the final values are centered around zero with a unit standard deviation. It is applied by subtracting the mean of the series from each value and dividing the result by the standard deviation. Although it is also a continuous variable, the transformations applied to the target variable are different. They are scaled by subtracting the median of the series from each value and dividing the result by half of the interquartile range.

2.2. Model pre-training in the source domain

TFT is the DL model selected to predict the 24 h ahead hourly electricity load, being pre-trained in the source domain and then transferred to the target domain according to the TL strategies selected.

2.2.1. TFT architecture

TFT is an attention-based deep learning model for forecasting of multi-horizon time series, which includes interpretable insights into temporal dynamics [28]. In particular, TFT facilitates the interpretability throughout the identification of the persistent temporal patterns, the globally-important features, and the meaningful events for the

forecasting task. This interpretation ability allows to understand how and why the model arrives at its prediction, in contrast with “black box” models, where the internal mechanisms are not easy to interpret, and it is difficult to determine the importance of each input.

To achieve high prediction performance on a broad range of tasks, TFT uses canonical elements in its architecture to efficiently address different input types: static variables, past-observed, and future known time varying inputs. Thus, the principal components of TFT are: (1) Gating module to filter out any unnecessary components of the architecture. This enables a more concise structure of the model, making it applicable to both simple and complex scenarios. (2) Variable selection network that allows to identify the most important input variables at each time step and ignore the less relevant ones. (3) Static covariate encoders that are used to integrate static information, which is helpful to improve prediction accuracy. (4) Quantile forecasting to calculate prediction intervals at each temporal horizon. (5) Temporal processing to learn both long and short-term temporal patterns. Local information is processed using a sequence-to-sequence layer, whereas a temporal self-attention decoder is employed to learn long term patterns. This self-attention decoder is the Masked Interpretable Multi-Head Attention layer. This layer builds on the Masked Multi-Head [42] to selectively attend to relevant parts of the input sequence while ignoring irrelevant parts, making it an effective tool for capturing long-term temporal patterns. The "masked" component ensures that each position can only attend to previous positions in the sequence, preventing the model from cheating by accessing future information during training. What sets this layer apart is its interpretability, the attention weights produced by this mechanism are interpretable, allowing users to gain insight into which parts of the input sequence are most important for the model's predictions.

Overall, the combination of these architectural components enables TFT to effectively handle a wide range of input types and capture complex temporal dependencies, resulting in high prediction performance on a variety of forecasting tasks. In fact, attention-based forecasting models are widely used to solve load forecasting problems [43,44]. The overall architecture of TFT can be seen in [28] and in Figure 2.

2.2.2. Pre-training

The pre-trained TFT does not include a series identification for each building in the source domain, allowing to learn general knowledge amongst series, and not specific patterns of a certain building. However, the model was also trained with a series identifier to know its performance on each building if traditional machine learning were considered.

TFT was also used to guide feature selection through Backward Elimination [45], since it helps this step through the identification of globally-important variables. Initially the TFT model was trained with all the features considered. Then, the variable with less importance was removed, and the accuracy of both models compared. This process was repeated until no improvement is observed.

The initial network parameters of the TFT are set by default, and then optimized using Neural Network Intelligence (NNI) Toolkit [46]. NNI facilitates the definition of the search space, which is the set of all possible values that the hyperparameters can take. The user can define the range and type of values for each hyperparameter, and NNI will explore every possible combination of these hyperparameters to find the optimal configuration for the model.

During the hyperparameter optimization process, NNI trains the TFT model with different hyperparameter configurations, and evaluates its performance. NNI uses a variety of algorithms, such as Grid Search, Random Search, and Bayesian Optimization, to efficiently search the hyperparameter space and find the best configuration.

Once the optimal hyperparameter configuration is found, it is saved and transferred to the target domains via different Transfer Learning (TL) methods.

2.3. Transfer learning to the target domain

Once the model is pre-trained in the source domain, TL technique is used to transfer the knowledge to the target domain. For this purpose, different TL strategies are analyzed.

2.3.1. Principle of transfer learning (TL)

Traditional ML models attempt to learn tasks independently, based only on the corresponding learning domain. However, TL focuses on storing knowledge from other existing domains (source domains) and applying it to a different but connected task (target domain). This is especially useful when the target domain has fewer training data. Since the use of DL techniques requires a considerable amount of data, the use of TL can help to achieve better performance with a smaller dataset. Both strategies are outlined in Figure 1.

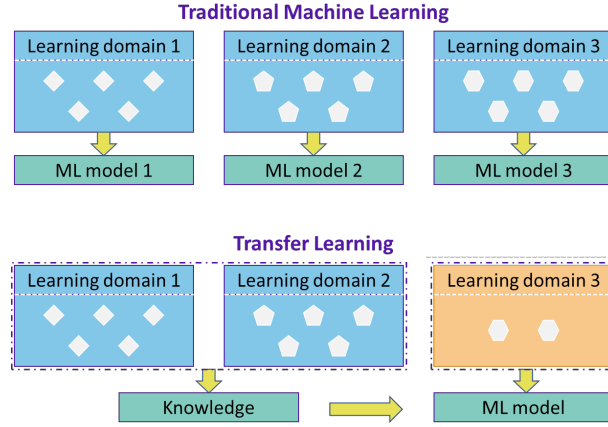


Figure 1. Traditional learning versus transfer learning.

As described in [47], the framework of TL can be proposed as follows. A domain D comprises two components as $D = \{X, P(X)\}$, where X represents the set of variables and $P(X)$ their marginal probability. On the other hand, a task T is also defined by two components: $T = \{Y, f(\cdot)\}$, being Y and $f(\cdot)$ the label space and the objective predictive function, respectively. Given a target domain D_T and learning task T_T , a source domain D_S and a learning task T_S , the objective of transfer learning is to enhance the learning of $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S , where $D_S \neq D_T$, or/and $T_S \neq T_T$.

In this study, $D_S \neq D_T$ and $T_S = T_T$. Since both source and target domains share the same variables as inputs, the condition $D_S \neq D_T$ indicates that the marginal probability distributions are different. Moreover, $T_S = T_T$ implies that the predictive function $f_S(\cdot)$ learned from the source domain can be used in the target domain after fine-tuning. The learning task for both domains is to predict 24 h ahead the building electricity consumption from meteorological and calendar data.

2.3.2. TL strategies

Network-based TL is a category of transfer learning proposed to solve the challenge of developing complicated deep learning models with few data [48]. The main idea is to partially reuse the network pre-trained in the source domain, including its network structure and connection parameters. The structure of the best pre-trained model is transferred to the target domain, and then, the model weights are adjusted according to the TL strategy selected. It is a kind of parameter-based TL [47], which assumes that models developed for relevant tasks would share some model parameters. The network-based transfer can be implemented following two strategies [49].

192 The first strategy is to utilize the pre-trained model as a feature extraction. In this, the whole set of parameters of the
193 pre-trained model are transferred, and the parameters in some layers of the new model are frozen, while the rest are
194 fine-tuned using target data. In case A, the whole pre-trained model is fixed except for the output layer, while in case
195 B only the layers previous to the Static Enrichment are fixed. Figure 2 shows the layers fixed in each case over the
196 architecture of the TFT.

197 The second implementation strategy is weight initialization. In this case, the weights are used to initialize the training
198 process and they are then fine-tuned with the training data available in the target domain.

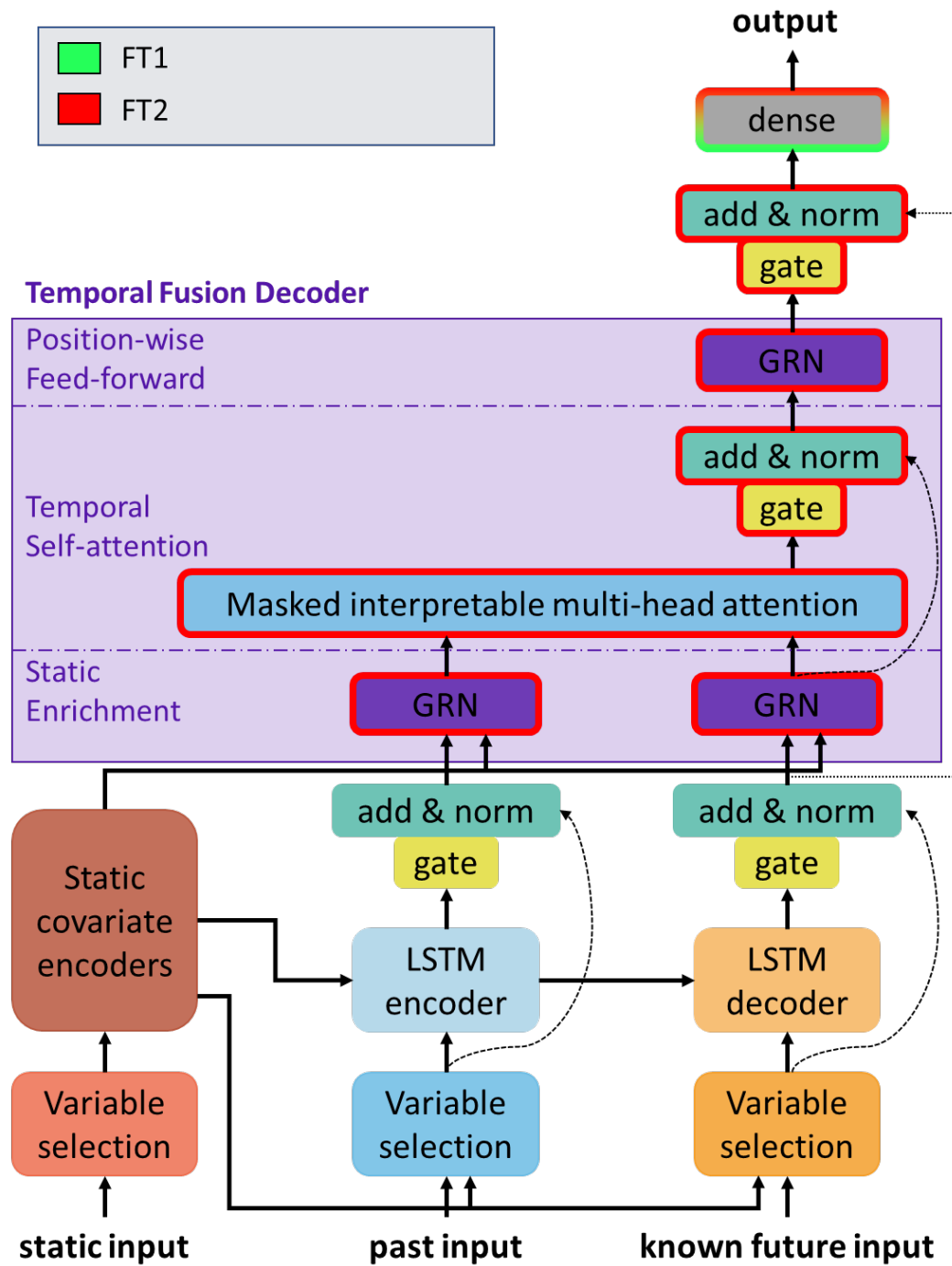


Figure 2. TFT architecture showing layers frozen in each TL strategy.

2.4. Data splitting

The dataset is divided into three sets for training purposes (training, validation, and testing). The training set usually contains the bulk of the data and is used to fit the model, while the validation set is used to provide an unbiased

evaluation of the model while tuning model hyperparameters. Finally, the test set is used to provide an unbiased evaluation of a final model fit on the training set.

The method used to assign each sample to these 3 sets is explained within the next steps: (1) The last samples of each series are assigned to the test set, depending on the number of samples of the selected test size. (2) The remaining samples are divided into a number of subseries, equal to the parameter number of folds. (3) Each of these subseries is then divided into training and validation samples. These validation samples are the last 20% of the samples in each subseries and the training samples are the remaining ones.

Since the samples in a timeseries problem are created as a rolling window over each timeseries, the first samples of the training set of one of the subseries carry information from the previous subseries validation set within its input data. Therefore, to ensure that the validation data cannot be known during model fitting, every subseries after the first one drops the first n samples, n being the input length of the model, in this case 168.

The number of folds is 10 in the source domain, while in the target domain it is 1, because each subseries after the first one loses 168 samples, and the target domain does not have enough samples to warrant that. In addition, the test size is 10% of all the samples in the source domain, whereas, in the target domain, it is 240 samples, which allows trainings with different length series to be compared using the same test set.

2.5. Evaluation metrics

To evaluate and compare the forecasting performances, different statistical error metrics were employed: root mean square error (RMSE), mean absolute error (MAE), and symmetric mean absolute percentage error (SMAPE). Since RMSE depends on the scale and on the sample size, CV_RMSE was also employed. It represents the ratio between RMSE and the mean of the observations. Moreover, weighted quantile loss (WQLoss) was the metric selected to train the different TFT implementations. These evaluation metrics are defined as:

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

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

$$SMAPE = \frac{1}{N} \sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{\frac{(y_i + \hat{y}_i)}{2}} \quad (3)$$

$$CV_RMSE = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|^2}}{\frac{1}{N} \sum_{i=1}^N y_i} \quad (4)$$

$$WQLoss = \frac{\sum_{i=1}^N \sum_{q \in Q} \max((q(y_i - \hat{y}_i), (q - 1)(y_i - \hat{y}_i))}{\sum_{i=1}^N y_i} \quad (5)$$

where $\hat{y}_i$ and y_i represent the forecasted and real values respectively, and Q is the set of quantile values to be fitted, in this study $Q = \{0.02, 0.1, 0.25, 0.5, 0.75, 0.9, 0.98\}$. This definition of SMAPE provides a value between 0 and 2.

3. Data description

The dataset used contains electricity consumption, meteorological variables and calendar variables from 6 different buildings from the tertiary sector located in Galicia, Spain. The historical electricity consumption was gathered from smart meters with 15-minute resolution, whereas the meteorological variables were collected from the nearest weather station from MeteoGalicia [50], and include temperature, relative humidity, and dew temperature. Finally, calendar data include variables that can represent daily, weekly or seasonal patterns, such as hour, day of week, month, and day type. The day type feature is categorized as holidays, before or after holidays, and work days, which include any day that is not considered a holiday, even weekends.

The dataset was filtered to avoid periods with missing or invalid values on the electricity consumption series, and converted to hourly values. As a result, four buildings were selected for source domain and two for target domain. The length of each time series can be seen in Table 1. For source domain, it varies from 1 year to nearly 1.5 years, while for target domain the length is 2 months.

Building	Length	Building	Length
BS_1	365 days	BS_4	373 days
BS_2	531 days	BT_1	60 days
BS_3	424 days	BT_2	60 days

Table 1. Length of time series for the different buildings.

The six tertiary buildings are located in the same geographic area, and therefore, share the same relative weather conditions, although they present diverse load patterns. As shown in Figure 3, all buildings show a regular pattern during weekdays, with clear differences during the weekend for all buildings except BS_3. For weekdays, the daily power consumption starts to rise between 6 a.m. and 8 a.m., and usually have a peak around midday and the lowest value during the night. BS_1 and BS_4 have a low and flat load during weekends, where BS_2 profile shows a high decrease only on Sundays. With respect to the variations between summer and winter, there are some buildings where the hourly load is higher in winter than in summer, such as BS_2, while others like BS_1 present a similar pattern throughout the year.

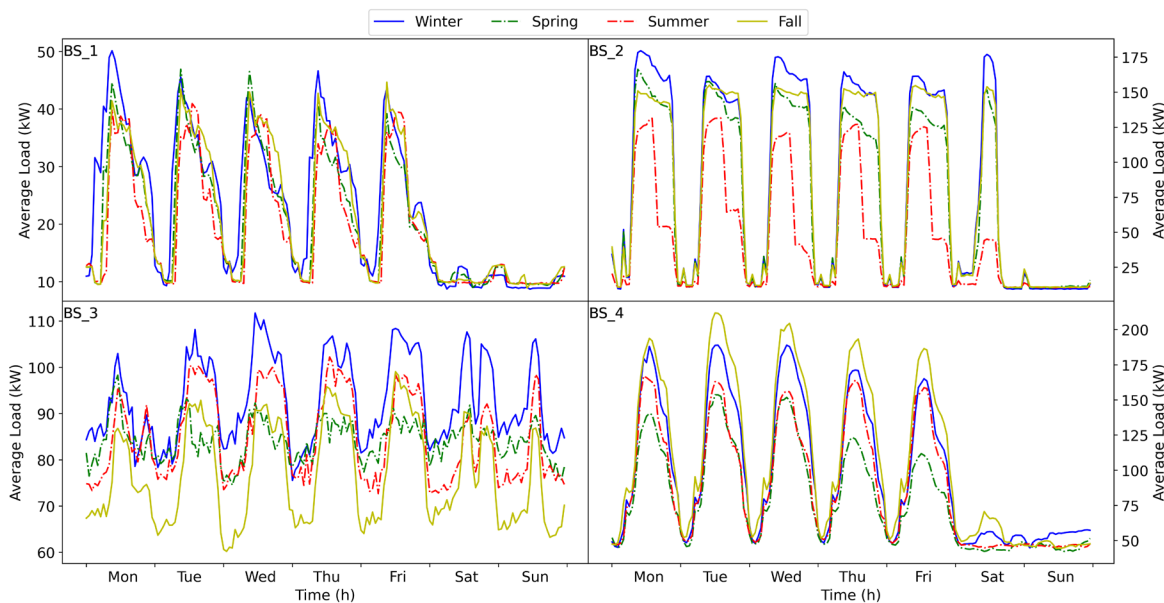


Figure 3. Typical weekly load variation for the four source buildings in the different seasons.

With respect to the target buildings, Figure 4 displays the average load pattern of both buildings. In this case, there is no distinction between seasons since the length of both series is only 60 days. The pattern of load variation is similar in both series, with a regular behavior during weekdays and a nearly flat consumption during weekends. The daily consumption during weekdays starts to rise in the early morning, with the maximum value reached around midday and the minimum at night. However, the base consumption and the range of load variation is higher in BT_1 than in BT_2.

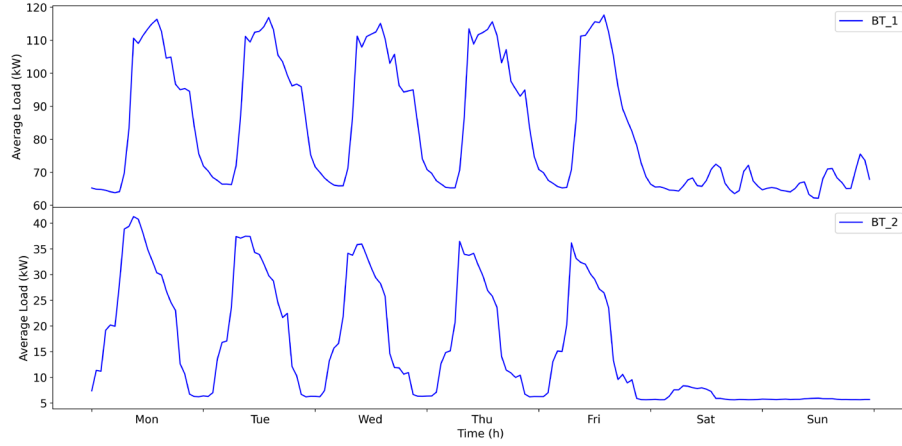


Figure 4. Average load variation for the target buildings.

The main values of electricity consumption patterns can be seen in Table 2. For source buildings, BS_3 and BS_4 are the buildings with the highest electricity consumption and BS_1 the series with the lowest value. Target buildings present a dissimilar average daily consumption.

	BS_1	BS_2	BS_3	BS_4	BT_1	BT_2
Min hourly load (kW)	2.31	5.00	28.00	35.00	22.00	5.09
Max hourly load (kW)	78.15	261.00	141.00	247.00	177.00	48.94
Average daily consumption (kWh)	505.30	1732.30	1932.34	2192.97	2182.37	370.64

Table 2. Consumption values for the 6 buildings.

4. Results and discussion

This section presents and discusses the results derived from the application of the methodology to the building data described above. First, the results of the pre-training process are analyzed and the performance of TFT is compared with other forecasting methods. Then, the effectiveness of the different transfer learning strategies under consideration is shown and discussed.

4.1. Pretrained model

The performance of the TFT model pretrained in the source domain was compared with the multilayer perceptron (MLP) model and the LSTM model, two approaches widely used for the prediction of electricity consumption [9]. The results are shown in Table 3. As it can be seen, TFT outperforms MLP and LSTM with lower values for the different metrics evaluated. Comparing TFT with MLP, it reduces RMSE, MAE and SMAPE by 25.1%, 29.4%, and

29.7%, respectively. When compared with LSTM, the errors are reduced by 11.4%, 14.0%, and 14.8% respectively. This improvement may be explained by the different components of the TFT architecture, which allows it to capture dependencies at different time scales, select only relevant features, and address each input type efficiently. This better performance of TFT was also found in the prediction of photovoltaic energy production [51] or wind speed [52].

	MLP	LSTM	TFT
RMSE (kW)	23.22	19.63	17.39
MAE (kW)	16.64	13.60	11.70
SMAPE	0.28	0.23	0.19

Table 3. Forecasting errors for the different pre-trained models.

The performance of TFT for each building was also analyzed and the results are shown in Table 4. Taking into account the metrics that are scale invariant, CV_RMSE and SMAPE, the better performance is always achieved for BS_3. On the other hand, the lowest accuracy depends on the metric selected, varying between BS_1 and BS_2. According to [18], a result with a CV_RMSE value below 0.3 can be considered a reasonable forecast. In this case, all buildings except BS_1 meet this threshold, although BS_1 is really close.

	BS_1	BS_2	BS_3	BS_4
RMSE (kW)	6.67	20.62	13.77	25.69
CV_RMSE	0.31	0.27	0.16	0.24
SMAPE	0.17	0.27	0.13	0.17
QWLoss $\times 10^{-3}$	4.47	4.82	2.98	3.86

Table 4. Forecasting errors for the three models considered, and consumption characteristics of the test set.

The differences between buildings may be due to the diversity of range in load demand and the regularity of the electricity consumption; since a better forecasting accuracy is reached when both the variations in load through the day and the presence of irregular patterns are lower. Moreover, the presence of very low values of load demand may increase these normalized metrics, since a small absolute error may represent a high percentage of the load value [18,21]. This may be the reason behind the higher error of BS_1 and BS_2 (Table 2). The characteristics of each series can be observed in Figure 5, which shows an example of the pretrained TFT model's day-ahead load forecasting predictions. It also represents the corresponding prediction intervals, which indicate the level of uncertainty associated with each prediction. At each timestep, the model provides seven values: the actual forecast and three underpredicted and three overpredicted forecasts at different quantile values, as trained by the weighted quantile loss function.

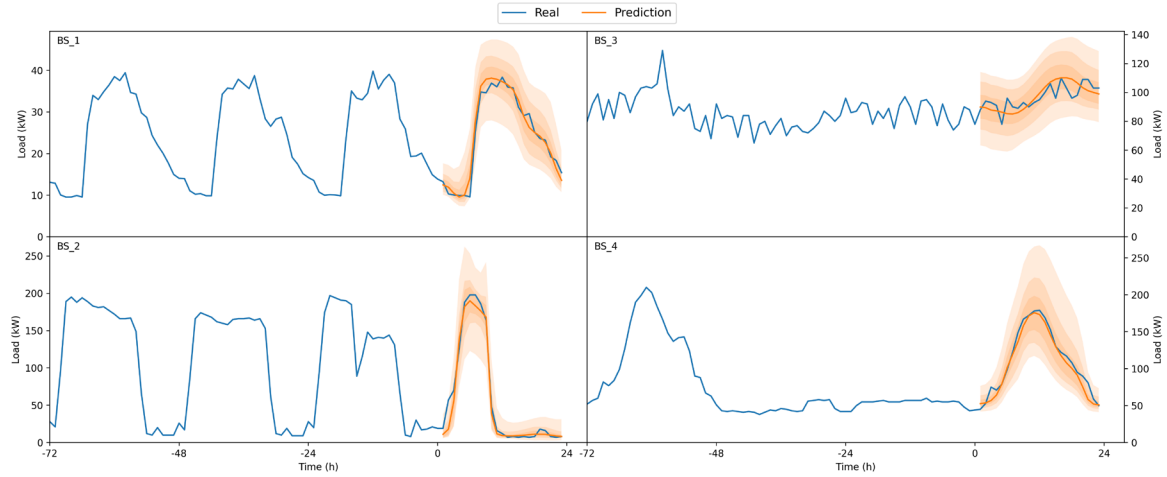


Figure 5. Example of observed and predicted load demand for the four source buildings.

The pretrained TFT model is described in the following table and figures. Table 5 shows the hyperparameters of the pretrained TFT model optimized with NNI,, and also the encoder length of the model. The encoder length refers to the number of past inputs required for the model to make a prediction. In this case, since the time step is hourly and the encoder length is 168, the past input needed by the model is 168 hours, which is equivalent to a week of data. Figure 6 represents the attention weight patterns for one-step-ahead forecasts and Figure 7 the importance of past-observed and future-known time varying inputs (encoder and decoder variables, respectively).

Hyperparameter	Value	Hyperparameter	Value
LSTM layers	2	Learning rate	0.0005
Hidden size	30	Encoder length	168
Hidden continuous size	15	Dropout	0.2
Attention head size	2	Batch size	256

Table 5. Parameters of the pretrained TFT model.

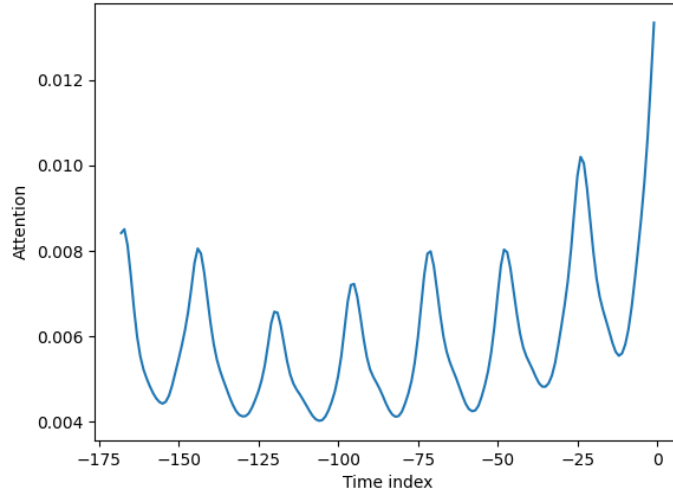


Figure 6. Attention weight patterns for one-step-ahead forecast of the pretrained TFT.

Figure 6 shows that the attention displays different peaks at 24 hour-intervals, which is understandable since the electricity consumption also presents a daily pattern (Figure 3). Therefore, the most important past steps for the TFT model are the preceding hours and the values at the same hour of previous days. Moreover, the value of the daily attention peaks decreases with time, increasing again for the same day of the previous week.

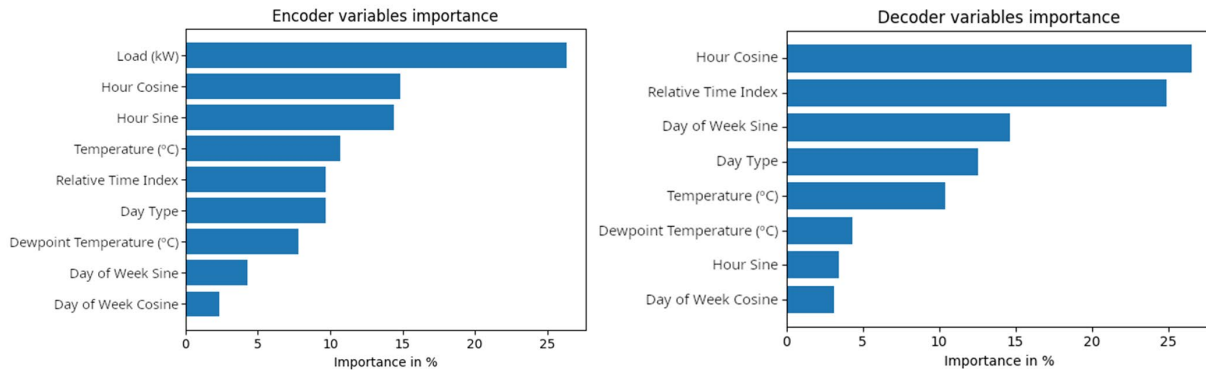


Figure 7. Encoder and decoder variables importance for the pretrained TFT.

The encoder variables are the variables for which past values are known at prediction time, and consist of the variables selected through Backward elimination and the relative time index. Backward elimination is a feature selection technique that helps identify the most relevant features for a task. The process involves fitting the model with all available features, removing the least significant variable, and refitting the model. If the results are better than before, the variable is permanently removed, otherwise, it is reintroduced into the model. The process continues until all variables have been checked and ensures that all kept variables are relevant features for the current task. In this case,

the most important encoder variable is the target variable (over 25%), followed by the features related with the hour of the day. Together, these three variables add up to nearly 60% of the importance. On the other hand, the sine and cosine transformation of the day of the week are the features with the lowest significance on the encoder side.

With respect to the decoder variables, the cosine transformation of the hour and the relative time index represent nearly 50% of the importance, while the cosine transformation of the day of the week is again the feature with the lowest significance. Besides, the weather variables (temperature and dew point) also help to improve the performance of the model, with a significance of around 20% and 15% on the encoder and decoder side, respectively.

These results highlight the importance of the calendar variables on the prediction of building electricity consumption, due to the daily, weekly, and seasonal patterns present on the target variable (Figure 3).

4.2. Transfer learning

The effectiveness of the different TL strategies was evaluated on two target buildings, considering the six different models summarized in Table 6. These TL strategies were trained using 10 days of historical data, divided into training and validation sets. Each experiment was replicated several times to avoid the influence of randomness in the comparisons. The average evaluation results obtained are shown in Figure 8. Moreover, some variables related to the time and resources needed for each strategy are shown in Table 7.

	Strategy	Description
Feature extraction	FT1	Freezing all layers except the last one, fine tuning the last layer in the target domain
	FT2	Freezing the layers before the static enrichment, fine tuning the remaining layers
Weight initialization	WT1	Fine tuning all parameters without freezing any layers
No transfer	NT1	Training the target series without TL
	NT2	Training all series together without TL
Zero shot	ZS1	Using the pretrained model to predict the new series without any new training

Table 6. Summary of the different strategies considered.

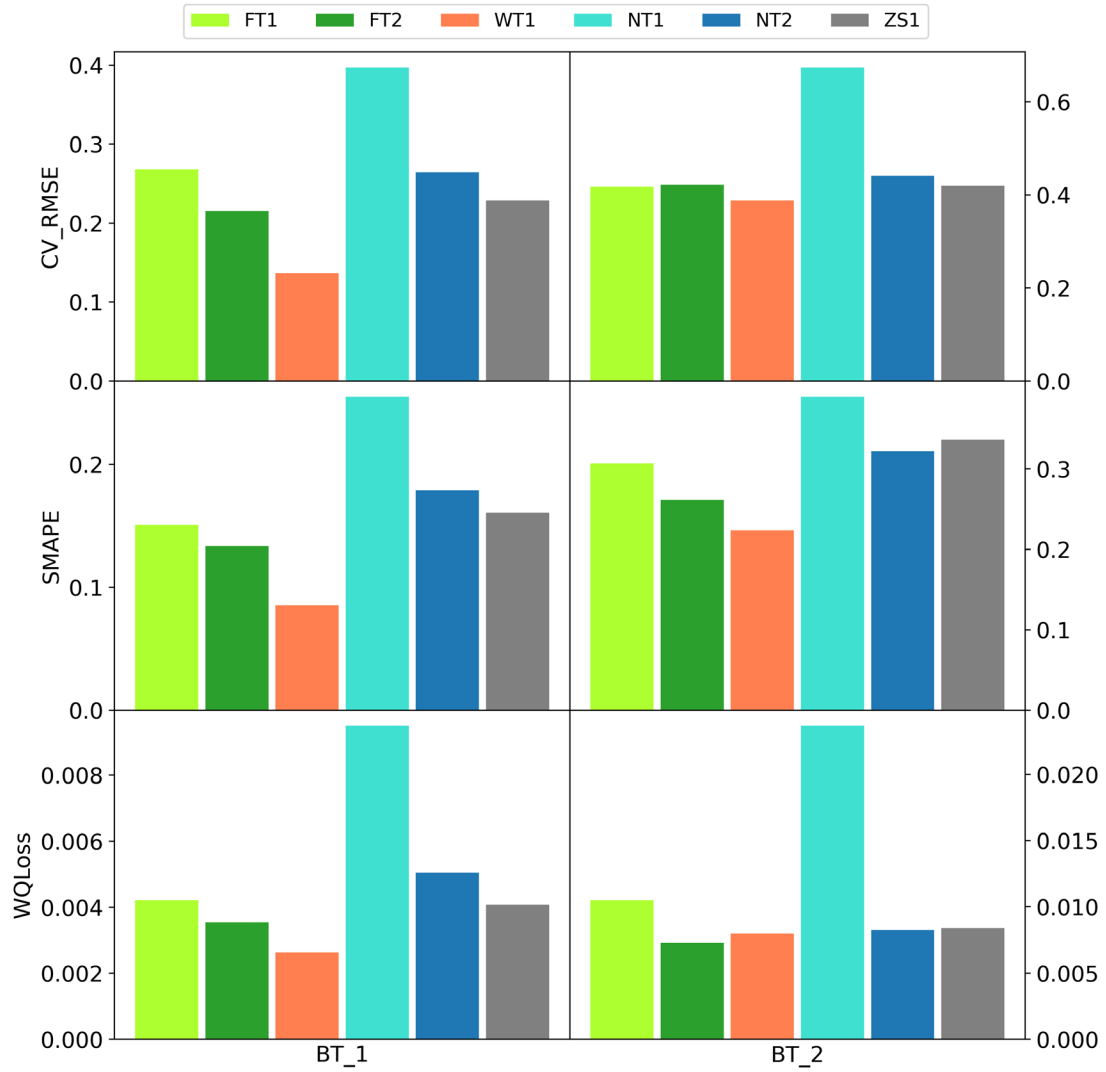


Figure 8. Comparison of the evaluation results for the different strategies considered.

Strategy	Number of trainable parameters	Training time (s)
FT1	217	60
FT2	15300	85
WT1	92800	155
NT1	92800	140
NT2	92800	7000
ZS1	0	0

Table 7. Variables related to the time and resources consumed.

The results indicate that the knowledge transferred through the pretrained model helps to improve the prediction of electricity consumption of buildings with limited availability of historical data. Thus, all TL strategies considered improved the results obtained without TL. WT1 provides the best model, while NT1 is the worst option between the strategies considered. Comparing NT1 and WT1, CV_RMSE, SMAPE, and WQLoss were reduced for building 1 by 66,5%, 66,2%, and 72,3% respectively; while the reduction was 42.6% for SMAPE, 42.4% for CV_RMSE, and 66.2% for WQLoss in building 2. This is reasonable since NT1 model parameters were adjusted with only 10 days of historical data and without any other knowledge of the time series.

FT1 is the TL strategy with the worst performance, followed by FT2, while WT1 almost always obtained the best results. The only exception is when WQLoss is considered for BT_2, where the best implementation strategy is FT2. The possibility of fine tuning all the pretrained weights makes WT1 a better option, since the performance of both FT1 and FT2 depends on the similarity between the frozen weights of the pretrained model and the best parameters for those weights. However, the implementation of FT1 and FT2 requires less computation time and resources, since the number of weights to adjust is reduced (Table 7). The difference in forecasting accuracy between FT1, FT2, and WT1 is lower for BT_2 than for BT_1, and even the FT2 model achieved better accuracy than WT1 considering WQLoss metric for BT_2. This difference may indicate that the pretrained model is more suitable to forecast electricity consumption in BT_2, and therefore, FT2 could be an adequate TL strategy for BT_2, but not for BT_1. Moreover, the improvement in forecasting accuracy between ZS1 and the best TL strategy is 57% for SMAPE, 40% for CV_RMSE, and 36% for WQLoss for BT_1; while the reduction is 33%, 8%, and 13% for SMAPE, CV_RMSE, and WQLoss, respectively for BT_2. This also suggests that the pretrained model is more adjusted to predict BT_2 than BT_1. Therefore, a measure of similarity between the source and target buildings could assist to select the best strategy to perform TL. This conclusion was also found by [41,49].

It is also worth mentioning that, in general, the NT2 strategy does not improve the results obtained with TL, being the implementation that requires more computation time and resources (Table 7). It implies training again all the series together, learning a general knowledge from the larger series and, therefore, improving NT1. However, this improvement is not enough to compensate the higher number of resources needed, and this strategy would not be considered for the prediction of load demand of buildings with limited availability of historical data.

To further evaluate TL strategies, WP1 was implemented with different data availabilities for model training, from 10 days to 60 days. These trainings with different lengths were evaluated using the same test data. The results are shown in Figure 9.

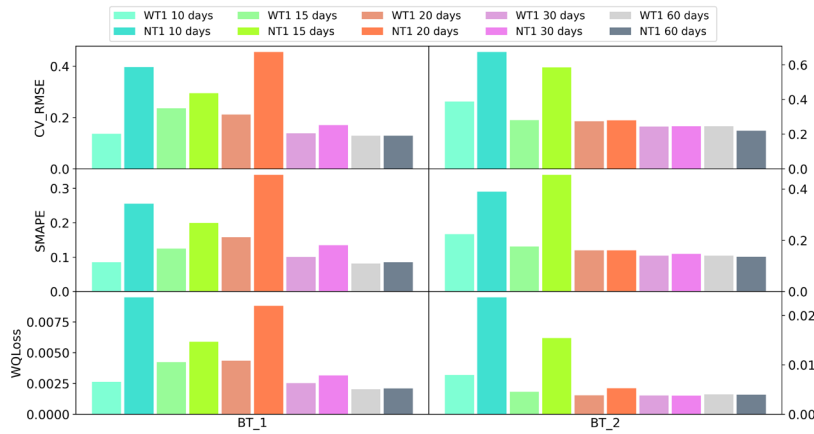


Figure 9. Different data availabilities for model training.

The results indicate that the importance of TL diminishes with the increase in data availability. In general, the errors of both WT1 and NT1 models and the difference in performance between them decreases with the increment in training data. Thus, the results of both models are equivalent for 60 days, and even for 30 days for BT_2. The presence of a very regular pattern on the electricity consumption for these target buildings, especially in BT_2, could be the reason why the difference between models disappears with only one or two months of available data. Moreover, the lower error of WT1 in BT_1 for 10 days than for 20 or 30 days may be due to a higher similarity between training and test patterns for 10 days than for more training days, where the model is more general. It is worth pointing out that WT1 is capable of reproducing the behavior of both target buildings with only 10 training days, which was the minimum tested. (Figure 10).

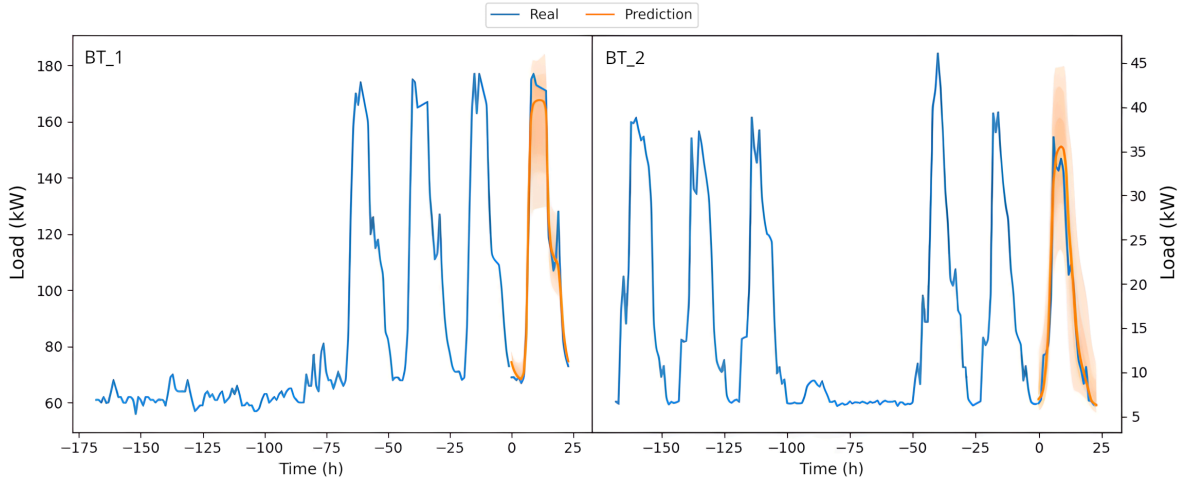


Figure 10. Example of observed and predicted load demand for the target buildings on WT1 10 days.

4.3. Managerial insights

Electricity demand forecasting has become an essential strategy for the energy management in local energy communities, since it facilitates the planning and allocating of available resources within the community. The combination of TFT with the TL methodology provides an accurate model for the prediction of energy demand in buildings with limited historical data. Thus, energy managers can bring new users to the dynamic of the community, optimizing energy distribution and reducing costs. The proposed methodology is applied following the steps shown in Figure 11.

Firstly, the energy manager should analyze the data of the existing buildings. Based on this analysis, a set of buildings that exhibit similarities with the new building, as well as different characteristics of the community network, should be selected. It is important to ensure that the number of samples selected for each building are similar.

After this, the TFT needs to be trained using these selected source buildings. This will result in a pre-trained TFT model that is ready to be used for transfer learning.

Once your TFT model is pre-trained, one of the TL strategies proposed in this document should be selected and applied to the pre-trained model during retraining, preferably the WT1 strategy, which achieved the best results.

The retraining process should be executed for every target building, including those that lack of historical data, either because they are new buildings or because they were already trained but not long enough.

For a new building a minimum of 10 days of historical data is needed to retrain the model, since the model inherently needs 7 days for its inputs and a few samples to capture the building's behavior during the training process, these would be the remaining 3 days.

Once this new building is trained, predictions over the new TL-TFT model can be generated and used for energy management control and planning. However, it was observed that, generally, the more data a TL model is trained on, the more accurate the model will be, so these models should be periodically updated as more historical data is made available.

Assuming that the energy manager already possesses a model to predict every building with enough data, if updating the TL model of the new building with additional input data does not show an improvement in the accuracy of the obtained predictions, this building should be ready to be included in the existing model with all the existing buildings. Nonetheless, since training a dataset with every building with enough data is very resource intensive, adding a new target building and training a new model to predict these buildings, including the new one, should only be done when there is a moderate confidence that the TL approach doesn't offer any improvements. Namely, there should be several trainings of the target building using TL that don't show any improvement over the previous ones before the target building is added.

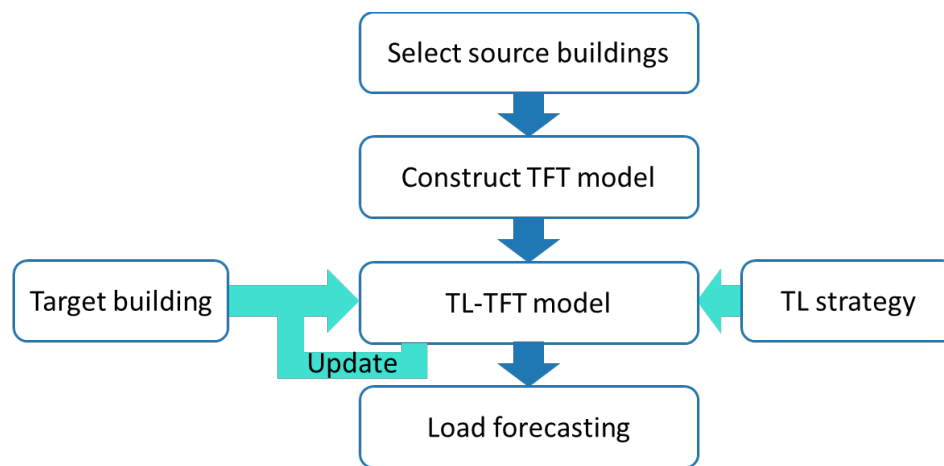


Figure 10. Steps of the proposed methodology.

5. Conclusions

A short-term load forecasting methodology was developed expressly to improve the management of local energy communities, combining the TL approach with the TFT model. This combination facilitates the incorporation of new

users to the community when their availability of historical data is limited, since the knowledge transferred helps to improve the prediction of data-poor buildings. This methodology was validated in a real case study. The TFT model was tested on a local energy community of 4 buildings for which enough data is available to use DL techniques, while the TL strategy was evaluated by adding two new buildings to the community with a limited amount of historical data. TFT provides accurate forecasting results, reducing the error compared to other forecasting methods widely used in this field, such as LSTM and MLP models. In comparison to LSTM, the RMSE is reduced by 11.4%, and the difference is accentuated in the case of MLP, with a reduction of 25.1%. This is probably due to the ability of the TFT architecture to capture dependencies at different time scales, select only the most relevant features, and address each different input type efficiently.

On the other hand, 6 models with different training strategies are analyzed to validate the TL performances. The conclusion reached is that the knowledge transferred enhances load forecasting in the new buildings evaluated. Weight initialization is the strategy that provides the best results and consists of fine-tuning the whole set of the parameters of the pre-training model without freezing any layer. The comparison between this model and the model trained without TL gives a high reduction in all evaluation metrics for the two buildings. However, other strategies of TL based on feature extraction may also be useful. These strategies are based on freezing some layers of the architecture and, therefore, their performance depends on the adjustment between the parameters of the pretrained model and the best parameters for the target building. Hence, a measure of similarity between the source and target buildings could assist to select the best strategy to perform TL. This conclusion was also found by [41,49].

The tests performed show that the TL strategy is useful when the availability of historical data is less than 60 days. From this point on, the model trained without TL achieved a similar performance. These buildings from the tertiary sector present quite a regular pattern of electricity consumption, which allows a good forecasting accuracy with only 2 months of historical data. For shorter times, TL seems like a favorable and useful choice. The main limitation of the method is the necessity of enough historical data of series with similar behavior to achieve a useful pretrained model. Thus, the accuracy of the load demand forecasting may be compromised if the target building consumption pattern differs significantly from those of the source buildings. In this case, the TL strategy may worsen the performance of the model, which is known as negative transfer learning. Moreover, this methodology is highly dependent on the

quality and quantity of the historical data. If the data is incomplete or inaccurate, the forecasting models may not provide accurate results.

However, the TL approach was found to improve the accuracy of forecasts when some similarity exists between source and target buildings, and can contribute to facilitate the management of local energy communities, for which reliable and accurate predictions are essential for efficient resource planning and allocation. Thus, it is possible to include buildings with scarce historical consumption data in the community and in the collaborative management.

Acknowledgments

This research has been supported by CERVERA Research Program of CDTI, the Industrial and Technological Development Centre of Spain, through the Research Projects HySGrid+, grant number CER-20191019, and CEL.IA, grant number CER-20211022.

References

- [1] European Commission, Directorate-General for Energy, Clean energy for all Europeans, Publications Office, 2019. <https://doi.org/doi/10.2833/9937>.
- [2] European Commission, Directive (EU) 2019/944 of the European Parliament and of the Council of 5 June 2019 on common rules for the internal market for electricity and amending Directive 2012/27/EU (Text with EEA relevance.), 2019. <https://eur-lex.europa.eu/eli/dir/2019/944/oj> (accessed June 27, 2022).
- [3] IRENA, Innovation landscape for a renewable-powered future: Solutions to integrate variable renewables, Abu Dhabi, 2019.
- [4] I. Antonopoulos, V. Robu, B. Couraud, D. Kirli, S. Norbu, A. Kiprakis, D. Flynn, S. Elizondo-Gonzalez, S. Wattam, Artificial intelligence and machine learning approaches to energy demand-side response: A systematic review, *Renewable and Sustainable Energy Reviews*. 130 (2020) 109899. <https://doi.org/10.1016/j.rser.2020.109899>.
- [5] T. Ahmad, H. Zhang, B. Yan, A review on renewable energy and electricity requirement forecasting models for smart grid and buildings, *Sustain Cities Soc.* 55 (2020) 102052. <https://doi.org/10.1016/j.scs.2020.102052>.
- [6] R. Lu, S.H. Hong, Incentive-based demand response for smart grid with reinforcement learning and deep neural network, *Appl Energy*. 236 (2019) 937–949. <https://doi.org/10.1016/J.APENERGY.2018.12.061>.
- [7] J. Coignard, M. Janvier, V. Debusschere, G. Moreau, S. Chollet, R. Caire, Evaluating forecasting methods in the context of local energy communities, *International Journal of Electrical Power and Energy Systems*. 131 (2021) 106956. <https://doi.org/10.1016/j.ijepes.2021.106956>.

- [8] H. Dagdougui, F. Bagheri, H. Le, L. Dessaint, Neural network model for short-term and very-short-term load forecasting in district buildings, *Energy Build.* 203 (2019) 109408. <https://doi.org/10.1016/J.ENBUILD.2019.109408>.
- [9] L. Zhang, J. Wen, Y. Li, J. Chen, Y. Ye, Y. Fu, W. Livingood, A review of machine learning in building load prediction, *Appl Energy.* 285 (2021). <https://doi.org/10.1016/J.APENERGY.2021.116452>.
- [10] M. Bourdeau, X. qiang Zhai, E. Nefzaoui, X. Guo, P. Chatellier, Modeling and forecasting building energy consumption: A review of data-driven techniques, *Sustain Cities Soc.* 48 (2019) 101533. <https://doi.org/10.1016/J.SCS.2019.101533>.
- [11] F. Kaytez, M.C. Taplamacioglu, E. Cam, F. Hardalac, Forecasting electricity consumption: A comparison of regression analysis, neural networks and least squares support vector machines, *International Journal of Electrical Power and Energy Systems.* 67 (2015) 431–438. <https://doi.org/10.1016/J.IJEPES.2014.12.036>.
- [12] K.P. Amber, M.W. Aslam, S.K. Hussain, Electricity consumption forecasting models for administration buildings of the UK higher education sector, *Energy Build.* 90 (2015) 127–136. <https://doi.org/10.1016/J.ENBUILD.2015.01.008>.
- [13] P. Chujai, N. Kerdprasop, K. Kerdprasop, Time series analysis of household electric consumption with ARIMA and ARMA models, in: *Proceedings of the International Multiconference of Engineers and Computer Scientists, IAENG Hong Kong, 2013*: pp. 295–300.
- [14] Y. Wang, J. Wang, G. Zhao, Y. Dong, Application of residual modification approach in seasonal ARIMA for electricity demand forecasting: A case study of China, *Energy Policy.* 48 (2012) 284–294. <https://doi.org/10.1016/J.ENPOL.2012.05.026>.
- [15] J.W. Taylor, Triple seasonal methods for short-term electricity demand forecasting, *Eur J Oper Res.* 204 (2010) 139–152. <https://doi.org/10.1016/J.EJOR.2009.10.003>.
- [16] J.W. Taylor, Short-term load forecasting with exponentially weighted methods, *IEEE Transactions on Power Systems.* 27 (2012) 458–464. <https://doi.org/10.1109/TPWRS.2011.2161780>.

- [17] A. Gellert, U. Fiore, A. Florea, R. Chis, F. Palmieri, Forecasting Electricity Consumption and Production in Smart Homes through Statistical Methods, *Sustain Cities Soc.* 76 (2022). <https://doi.org/10.1016/J.SCS.2021.103426>.
- [18] S. Walker, W. Khan, K. Katic, W. Maassen, W. Zeiler, Accuracy of different machine learning algorithms and added-value of predicting aggregated-level energy performance of commercial buildings, *Energy Build.* 209 (2020) 109705. <https://doi.org/10.1016/J.ENBUILD.2019.109705>.
- [19] A. Rahman, V. Srikumar, A.D. Smith, Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks, *Appl Energy.* 212 (2018) 372–385. <https://doi.org/10.1016/J.APENERGY.2017.12.051>.
- [20] M. Cai, M. Pipattanasomporn, S. Rahman, Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques, *Appl Energy.* 236 (2019) 1078–1088. <https://doi.org/10.1016/J.APENERGY.2018.12.042>.
- [21] G. Chitalia, M. Pipattanasomporn, V. Garg, S. Rahman, Robust short-term electrical load forecasting framework for commercial buildings using deep recurrent neural networks, *Appl Energy.* 278 (2020) 115410. <https://doi.org/10.1016/J.APENERGY.2020.115410>.
- [22] W. Kong, Z.Y. Dong, Y. Jia, D.J. Hill, Y. Xu, Y. Zhang, Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network, *IEEE Trans Smart Grid.* 10 (2019) 841–851. <https://doi.org/10.1109/TSG.2017.2753802>.
- [23] S. Muzaffar, A. Afshari, Short-Term Load Forecasts Using LSTM Networks, *Energy Procedia.* 158 (2019) 2922–2927. <https://doi.org/10.1016/J.EGYPRO.2019.01.952>.
- [24] T. Ahmad, H. Chen, Nonlinear autoregressive and random forest approaches to forecasting electricity load for utility energy management systems, *Sustain Cities Soc.* 45 (2019) 460–473. <https://doi.org/10.1016/J.SCS.2018.12.013>.
- [25] O. Valenzuela, I. Rojas, F. Rojas, H. Pomares, L.J. Herrera, A. Guillen, L. Marquez, M. Pasadas, Hybridization of intelligent techniques and ARIMA models for time series prediction, *Fuzzy Sets Syst.* 159 (2008) 821–845. <https://doi.org/10.1016/J.FSS.2007.11.003>.

- [26] X. Zhang, J. Wang, K. Zhang, Short-term electric load forecasting based on singular spectrum analysis and support vector machine optimized by Cuckoo search algorithm, *Electric Power Systems Research*. 146 (2017) 270–285. <https://doi.org/10.1016/J.EPSR.2017.01.035>.
- [27] F. Sheng, L. Jia, Short-Term Load Forecasting Based on SARIMAX-LSTM, 2020 5th International Conference on Power and Renewable Energy, ICPRE 2020. (2020) 90–94. <https://doi.org/10.1109/ICPRE51194.2020.9233117>.
- [28] B. Lim, S. Arık, N. Loeff, T. Pfister, Temporal Fusion Transformers for interpretable multi-horizon time series forecasting, *Int J Forecast.* 37 (2021) 1748–1764. <https://doi.org/10.1016/J.IJFORECAST.2021.03.012>.
- [29] D. Hsu, How much information disclosure of building energy performance is necessary?, *Energy Policy*. 64 (2014) 263–272. <https://doi.org/10.1016/J.ENPOL.2013.08.094>.
- [30] Y. Gao, Y. Ruan, C. Fang, S. Yin, Deep learning and transfer learning models of energy consumption forecasting for a building with poor information data, *Energy Build.* 223 (2020) 110156. <https://doi.org/10.1016/J.ENBUILD.2020.110156>.
- [31] H. Hermessi, O. Mourali, E. Zagrouba, Deep feature learning for soft tissue sarcoma classification in MR images via transfer learning, *Expert Syst Appl.* 120 (2019) 116–127. <https://doi.org/10.1016/J.ESWA.2018.11.025>.
- [32] W. Hu, Y. Qian, F.K. Soong, Y. Wang, Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers, *Speech Commun.* 67 (2015) 154–166. <https://doi.org/10.1016/J.SPECOM.2014.12.008>.
- [33] N. Saeed, N. King, Z. Said, M.A. Omar, Automatic defects detection in CFRP thermograms, using convolutional neural networks and transfer learning, *Infrared Phys Technol.* 102 (2019) 103048. <https://doi.org/10.1016/J.INFRARED.2019.103048>.
- [34] D. Wang, T.F. Zheng, Transfer learning for speech and language processing, 2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2015. (2016) 1225–1237. <https://doi.org/10.1109/APSIPA.2015.7415532>.

- [35] M. Ghiassi, S. Lee, A domain transferable lexicon set for Twitter sentiment analysis using a supervised machine learning approach, *Expert Syst Appl.* 106 (2018) 197–216. <https://doi.org/10.1016/J.ESWA.2018.04.006>.
- [36] S.M. Jung, S. Park, S.W. Jung, E. Hwang, Monthly electric load forecasting using transfer learning for smart cities, *Sustainability (Switzerland)*. 12 (2020). <https://doi.org/10.3390/SU12166364>.
- [37] P. Zeng, C. Sheng, M. Jin, A learning framework based on weighted knowledge transfer for holiday load forecasting, *Journal of Modern Power Systems and Clean Energy*. 7 (2019) 329–339. <https://doi.org/10.1007/S40565-018-0435-Z>.
- [38] D. Wu, B. Wang, D. Precup, B. Boulet, Multiple Kernel Learning-Based Transfer Regression for Electric Load Forecasting, *IEEE Trans Smart Grid*. 11 (2020) 1183–1192. <https://doi.org/10.1109/TSG.2019.2933413>.
- [39] X. Xu, Z. Meng, A hybrid transfer learning model for short-term electric load forecasting, *Electrical Engineering* 2020 102:3. 102 (2020) 1371–1381. <https://doi.org/10.1007/S00202-020-00930-X>.
- [40] D. Zhou, S. Ma, J. Hao, D. Han, D. Huang, S. Yan, T. Li, An electricity load forecasting model for Integrated Energy System based on BiGAN and transfer learning, *Energy Reports*. 6 (2020) 3446–3461. <https://doi.org/10.1016/J.EGYR.2020.12.010>.
- [41] C. Peng, Y. Tao, Z. Chen, Y. Zhang, X. Sun, Multi-source transfer learning guided ensemble LSTM for building multi-load forecasting, *Expert Syst Appl.* 202 (2022) 117194. <https://doi.org/10.1016/J.ESWA.2022.117194>.
- [42] A. Vaswani, G. Brain, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention Is All You Need, (2017).
- [43] L. Jia, J. Liu, A. Chong, X. Dai, Deep learning and physics-based modeling for the optimization of ice-based thermal energy systems in cooling plants, *Appl Energy*. 322 (2022) 119443. <https://doi.org/10.1016/J.APENERGY.2022.119443>.

- [44] A. Li, F. Xiao, C. Zhang, C. Fan, Attention-based interpretable neural network for building cooling load prediction, *Appl Energy*. 299 (2021) 117238. <https://doi.org/10.1016/J.APENERGY.2021.117238>.
- [45] I.H. Witten, E. Frank, M.A. Hall, Data Transformations, *Data Mining: Practical Machine Learning Tools and Techniques*. (2011) 305–349. <https://doi.org/10.1016/B978-0-12-374856-0.00007-9>.
- [46] NNI Documentation — Neural Network Intelligence, (n.d.). <https://nni.readthedocs.io/en/stable/index.html> (accessed June 2, 2022).
- [47] S.J. Pan, Q. Yang, A survey on transfer learning, *IEEE Trans Knowl Data Eng.* 22 (2010) 1345–1359. <https://doi.org/10.1109/TKDE.2009.191>.
- [48] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, C. Liu, A survey on deep transfer learning, in: V., Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, I. Maglogiannis (Eds.), *Artificial Neural Networks and Machine Learning -- ICAN 2018*, Springer International Publishing, 2018: pp. 270–279. https://doi.org/10.1007/978-3-030-01424-7_27/COVER/.
- [49] C. Fan, Y. Sun, F. Xiao, J. Ma, D. Lee, J. Wang, Y.C. Tseng, Statistical investigations of transfer learning-based methodology for short-term building energy predictions, *Appl Energy*. 262 (2020) 114499. <https://doi.org/10.1016/J.APENERGY.2020.114499>.
- [50] MeteoGalicía, (n.d.). https://www.meteogalicia.gal/web/inicio.action?request_locale=es (accessed June 30, 2022).
- [51] M.L. Santos, X. García-Santiago, F.E. Camarero, G.B. Gil, P.C. Ortega, Application of Temporal Fusion Transformer for Day-Ahead PV Power Forecasting, *Energies* 2022, Vol. 15, Page 5232. 15 (2022) 5232. <https://doi.org/10.3390/EN15145232>.
- [52] B. Wu, L. Wang, Y.R. Zeng, Interpretable wind speed prediction with multivariate time series and temporal fusion transformers, *Energy*. 252 (2022) 123990. <https://doi.org/10.1016/J.ENERGY.2022.123990>.