A Global Modeling Framework for Load Forecasting in Distribution Networks

Miha Grabner, Yi Wang, *Member, IEEE*, Qingsong Wen, *Member, IEEE*,
Boštjan Blažič, *Member, IEEE*, Vitomir Štruc, *Senior Member, IEEE*

***Abstract*—With the increasing numbers of smart meter installations, scalable and efficient load forecasting techniques are critically needed to ensure sustainable situation awareness within the distribution networks. Distribution networks include a large amount of different loads at various aggregation levels, such as individual consumers, low-voltage feeders, and transformer stations. It is impractical to develop individual (or so-called local) forecasting models for each load separately. Additionally, such local models also (i) (largely) ignore the strong dependencies between different loads that might be present due to their spatial proximity and the characteristics of the distribution network, (ii) require historical data for each load to be able to make forecasts, and (iii) are incapable of adjusting to changes in the load behavior without retraining. To address these issues, we propose a global modeling framework for load forecasting in distribution networks that, unlike its local competitors, relies on a single global model to generate forecasts for a large number of loads. The global nature of the framework, significantly reduces the computational burden typically required when training multiple local forecasting models, efficiently exploits the cross-series information shared among different loads, and facilitates forecasts even when historical data for a load is missing or the behavior of a load evolves over time. To further improve on the performance of the proposed framework, an unsupervised localization mechanism and optimal ensemble construction strategy are also proposed to localize/personalize the global forecasting model to different load characteristics. Our experimental results show that the proposed framework outperforms naive benchmarks by more than 25% (in terms of Mean Absolute Error) on real-world dataset while exhibiting highly desirable characteristics when compared to the local models that are predominantly used in the literature. All source code and data are made publicly available to enable reproducibility: https://github.com/mihagrabner/GlobalModelingFramework.**

***Index Terms*—Load forecasting, smart meter, global model, distribution networks, deep learning.**

I. Introduction

System-level load forecasting plays a key role in the energy industry, as accurate forecasts are critical to the planning and operation of both power systems and business entities [1]. With the integration of new smart-grid technologies, such as demand-response and distributed energy resources, load forecasting is also becoming increasingly important at various levels of the distribution networks. Here, accurate forecasts are often paramount for specific applications that require load predictions for grid management, storage optimization, peer-to-peer trading, demand-response programs and related tasks. [2], [3]. To support these developments, an increasing number of smart meters are being deployed at different network levels (e.g., individual-consumers, low-voltage (LV) feeders, and transformer stations), which necessitates research into scalable forecasting techniques that can be applied to a large number of measuring points with diverse characteristics, while also ensuring accurate predictions.

Conventional forecasting approaches that work well at the system level and include both deterministic and probabilistic load forecasting techniques, e.g., [4] and [5], are generally not well suited for load forecasting in distribution networks, where large volumes of electricity demand time series need to be efficiently modeled at multiple network levels and multiple locations. System-level demand typically consists of the aggregated demand from thousands or millions of consumers. According to the law of large numbers, such demand is much less volatile than LV demand, and is therefore easier to predict [6]. This assertion was, for instance, also validated in [7], where an empirical scaling law describing the accuracy of load forecasting at different levels of aggregation was introduced. It was shown that for different forecasting methods, a larger number of aggregated consumers consistently leads to better forecasting performance. As suggested by a recent survey on load forecasting in LV networks [3], the biggest difference between system-level and distribution network forecasting is due to: (i) the load volatility at lower voltage levels, where the data uncertainty significantly impacts the forecasting performance [7]; and (ii) the large number of forecasts needed in the distribution networks. These differences clearly warrant a different approach to load forecasting when attempting to cater to the characteristics of the distribution grids.

Traditional load-forecasting techniques commonly learn one prediction model for each given time series and are therefore considered to be *local*. These models work well when forecasts need to be made for a small number of time series but are less suitable for forecasting larger groups of time series, as is the case with distribution networks. Additionally, local forecasting techniques depend on the availability of historical data for each considered load and stationary behavior of the modeled time series. Last but not least, local methods often suffer from

This research was supported in parts by the ARRS Research Program P2-0250 (B) “Metrology and Biometric Systems”, in part by the ARRS Research Project DRIFT (nr. L2-4436), and in part by the Alibaba Innovative Research programme. *(Corresponding author: Yi Wang.)*

M. Grabner, B. Blažič, and V. Štruc are with the Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia (e-mails: miha@ubivivo.com, bostjan.blazic@fe.uni-lj.si, vitomir.struc@fe.uni-lj.si).

Y. Wang is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong SAR, China, and is also with The University of Hong Kong Shenzhen Institute of Research and Innovation, Shenzhen 518057, China (e-mail: yiwang@eee.hku.hk).

Q. Wen is with the DAMO Academy, Alibaba Group (U.S.) Inc., Bellevue, WA 98004, USA (e-mail: qingsong.wen@alibaba-inc.com).

the small sample size problem and tend to overfit, especially when they are based on heavily parameterized (e.g., deep-learning based) prediction models. To address the outlined shortcomings, so-called *global modeling techniques* started to appear recently in the time-series forecasting literature [8]. Unlike (the majority of) local models, these global techniques consider all time series within the same regression task and fit a single model to all time series in the given input set. In other words, global models are trained on all available time series simultaneously and are therefore able to exploit cross-series information during the learning stage. In addition to better (scalability and performance) characteristics, global methods have also been reported to generalize well to unseen (diverse) time series [9] due to the training data heterogeneity [10].

While global modeling techniques have been successfully applied to various problem domains with state-of-the-art results [11]–[14], very limited emphasis was given so far to electricity demand forecasting. In this paper, we address this gap and introduce a global modeling framework for forecasting electricity demand over a large set of diverse and heterogeneous load aggregates in distribution networks. Different from existing solutions from the literature, our framework relies on a single global model to make predictions for all loads. We implement the framework using an extended version of the N-BEATS deep-learning architecture from [11] that was recently shown to yield highly competitive performance for a wide variety of time-series prediction tasks. Furthermore, we propose an unsupervised localization mechanism to adapt the global forecasting models to the different load characteristics. We demonstrate the merits of the proposed framework (and localization mechanism) in comprehensive experiments on publicly available data released by the Commission for Energy Regulation (CER) in Ireland and show that the global modeling framework not only leads to superior forecasting accuracy compared to competing state-of-the-art solutions, but also exhibits a range of desirable characteristics not available with existing local models. In summary, our key contributions in this paper are:

1) We present a *global modeling framework* for efficient forecasting of a large number of loads in distribution networks at various levels of aggregation and implement it using a state-of-the-art (deep-learning) prediction model;
2) We introduce an unsupervised localization mechanism and ensemble-construction strategy to further improve the performance of the global model;
3) We conduct comprehensive experiments on real-world data to demonstrate the superiority of the proposed framework over the state-of-the-art in terms of overall forecasting performance, scalability and ability to handle missing and non-stationary time-series data.

II. Related Work

A considerable number of load-forecasting techniques has been proposed in the literature over the years, ranging from simple deterministic approaches to more elaborate probabilistic models that in addition to point estimates also offer insights into the uncertainties associated with the generated forecasts [15]. As emphasized in [3], most of this work has been centered on system-level forecasting, while forecasting at different levels of the distribution network is still underexplored.

Early solutions to load forecasting based on linear regression, multilayer perceptrons, support vector machines or boosting still provide competitive results for many application scenarios [3]. However, recent state-of-the-art models largely built on advances in deep learning due to the capabilities of modern neural networks of learning powerful prediction models directly from the input data. Following this trend, Chen *et al.* [4], for example, introduced a load forecasting approach based on deep residual networks and reported encouraging results. Kong *et al.* [2] proposed a model utilizing long-term short-term memory (LSTM) networks and showed that the model is capable of handling considerable load volatility when making predictions. Wang *et al.* [16] presented a similar LSTM-based framework, but focused on a probabilistic forecasting task. Tan *et al.* [17] investigated the problem of ultra-short-term power demand and proposed an ensemble approach, again designed around an LSTM network.

While the methods reviewed above train separate forecasting models for each considered load, a few attempts have also been reported in the literature to predict electricity demand for multiple loads at the same time, similarly in spirit to the global modeling framework, presented in this work. Shi *et al.* [18], for example, proposed a pooling-based deep Recurrent Neural Network (RNN) model for household forecasting, where consumers were split into different groups randomly, and the electrical load of each group was forecasted separately. Voß *et al.* [19] compared two forecasting strategies, i.e., one local model for all consumers and one global model for each consumer, for individual consumer load forecasting and reported that using a single model for all consumers yielded superior performance. Wang *et al.* [20] proposed a transformer-based model for forecasting over different types of loads simultaneously and explored the impact of attention mechanisms for this task. Han *et al.* [21] introduced a short-term forecasting model for individual residential loads based on deep learning and K-means clustering. The model first utilized K-means to extract similarities of a large number of loads and then employed deep learning for forecasting over the pooled data. Similarly, Yang *et al.* [22] presented a distribution-aware temporal pooling framework that uses data clustering to identify related time series for modeling. The framework dynamically assigns time series to a forecasting model to account for potential changes in the data distribution.

Related to our work is also the approach from [23], where forecasting in distribution networks is treated as a hierarchical forecasting task that aims to ensure that predictions at the lowest level aggregates are coherent with the predictions at the higher levels.

III. Problem Statement

Load forecasting in the distribution network consists of forecasting large sets of electricity demand time series at multiple network levels and multiple sites. Given the characteristics and requirements of future smart grids, such forecasting should be done for thousands of measuring points, rendering existing (computationally expensive) forecasting models, which commonly model each given time series with a distinct regression problem, impractical. This fact provides a very strong motivation for research into load forecasting models that: (i) scale better with the number of time series considered, (ii) generalize well over load time series with diverse statistical properties, and (iii) produce reliable load forecasts for large groups of time series data. In this section, we formally discuss load forecasting models and elaborate on how *global time series modeling* can address the shortcomings discussed above.

A. Problem Formulation

Let $\mathcal{Y} = \{\mathbf{y}_{i,1:T_i}\}_{i=1}^{N}$ represent a set of N time series, where $\mathbf{y}_{i,1:T_i} = [y_{i,1}, y_{i,2}, ...y_{i,T_i}]^T \in \mathbb{R}^{T_i}$ stands for an univariate time series, $y_{i,t} \in \mathbb{R}$ is the value of i-th time series at timestamp t, and T_i denotes the length i-th of the series. Furthermore, let $H \in \mathbb{N}^+$ and $K \in \mathbb{N}^+$ denote the forecasting horizon and the number of lags considered, respectively. In it's simplest form, the goal of short-term load forecasting (STLF) is to predict the vector of future values $\mathbf{y}_{i,T_i+1:H} = [y_{i,T_i+1}, y_{i,T_i+2}, ..., y_{i,T_i+H}]^T \in \mathbb{R}^H$ given past observations:

$$\mathbf{y}_{i,T_i-K+1:T_i} \mapsto \widehat{\mathbf{y}}_{i,T_i+1:H}, \tag{1}$$

where $\widehat{\mathbf{y}}_{i,T_i+1:H}$ stands for a point forecast of the i-th time series. In our problem setting, the time-series data consists of active power measurements, taken at various levels of aggregation in the distribution network, i.e., individual consumers, transformer stations and feeders.

B. Local and Global Modeling

From a modeling perspective, time series forecasting methods can, in general, be partitioned into methods that utilize either *local* or *global* modeling. An overwhelming majority of univariate time series forecasting methods developed in the last few decades use local modeling [8], where each time series is considered independently from all others. With this approach, each of the time series in a given set is assumed to come from a different data generating process and is, therefore, modeled individually, as a separate regression problem. As a result, a distinct forecasting model is estimated for each series. Conversely, global modeling considers all time series within the same regression task and fits a single univariate forecasting function to all of the time series in the set. This approach makes a strong assumption that all time series in the given set come from *the same data generating process*. Consequently, global (time series) modeling techniques are able to overcome some of the main limitations of local modeling approaches, i.e., (i) they prevent over-fitting because of larger sample size is used during training, (ii) they scale gracefully w.r.t. to the number of time series modeled, and (iii) they exploit cross-series information by sharing parameters across time series resulting in better performing models with highly competitive generalization capabilities [8].

Let $\mathbf{X}$ denote a feature matrix, where $\mathbf{X} \in \mathbb{R}^{m \times p}$, and let $\mathbf{Y}$ denote the corresponding target matrix, where $\mathbf{Y} \in \mathbb{R}^{m \times H}$. Here, m stand for the number of samples of a given time series and p for the number of features (e.g., lag values, categorical features etc.) in $\mathbf{X}$. The samples in $\mathbf{X}$ and $\mathbf{Y}$ for a given time series $\mathbf{y}_{1:T}$ are commonly created using a rolling window approach. Thus, for the (predictor) feature matrix $\mathbf{X}$, m p-dimensional feature representations are computed from the sampled lag-values $\mathbf{y}_{t_0-K+1:t_0}$, where t_0 is a sampling time instance. Similarly, the corresponding (response) target matrix $\mathbf{Y}$ is created from the corresponding future values $\mathbf{y}_{t_0+1:H}$ given the forecasting horizon H, where $H \leq T$. Additionally, assume that such feature matrices and corresponding targets are available for a set of N distinct time series, i.e, $\{\mathbf{X}_i\}_{i=1}^{N}$ and $\{\mathbf{Y}_i\}_{i=1}^{N}$. In a *local forecasting* approach, N models are trained in total and one function $f_i(\cdot)$ with model parameters θ_i is estimated for *each time series* in the set, as follows:

$$f_i(\mathbf{X}_i; \theta_i) = \mathbf{Y}_i. \tag{2}$$

Conversely, in a *global forecasting* approach, samples of all time series are first stacked together, so that:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{X}_3 \\ ... \\ \mathbf{X}_N \end{bmatrix}, \mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \mathbf{Y}_3 \\ ... \\ \mathbf{Y}_N \end{bmatrix}, \tag{3}$$

and one global forecasting function $f(\cdot)$ with model parameters θ is then estimated for *all time series* in the set:

$$f(\mathbf{X}; \theta) = \mathbf{Y}. \tag{4}$$

Such global models represent powerful tools for time-series forecasting that can be further improved through various localization strategies [8]. Global models, to the best of our knowledge, have not yet been widely considered for the problem of load forecasting despite their immense potential for this task. Our main contribution in this work, therefore, lies in the introduction of the *global modeling framework* for load forecasting in distribution networks and a new STLF model designed around this framework.

IV. Methodology

A. Proposed Framework

The proposed global framework for forecasting large groups of electricity demand time series is shown in Fig. 1 and consists of four distinct steps. In *the first step*, a pool of time series is used to fit a (single) global model to all available time series data. This process models the global data characteristics and is strongly related to existing transfer learning strategies where knowledge obtained when solving one problem is utilized in a different but related problem domain. In *the second step*, a clustering procedure is applied to identify data samples (clusters) that share specific data characteristics not necessarily captured by the learned global model. These data clusters serve as additional sources of information that supplement the information already considered by the global model. The identified clusters are then utilized in the *third step* for fitting

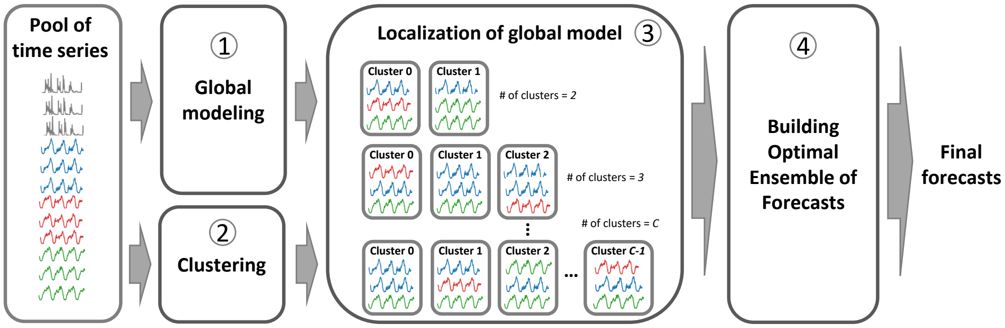


Fig. 1: High–level overview of the proposed *global* load forecasting framework.

multiple localized models on subsets of time series in the input pool. This step is needed to infuse the initial global model with information on cluster-specific data characteristics and further improve its generalization capabilities. Additionally, it also avoids potential locally optimal solutions that could arise if the forecasting models were trained on the data clusters directly. In the *fourth step*, the localized forecasts for each time series are combined into the final prediction by building an optimal forecasting ensemble. We note at this point that a global model applicable to different load aggregates is already generated after the first step of the framework, whereas the remaining three steps correspond to the (optional) localization mechanism used to further improve the model characteristics. Details on all outlined steps are given in the following sections.

B. Deep Learning-based Global Modeling

Given a set of time-series data, the proposed framework first learns a global forecasting model $f(\cdot)$ with model parameters θ, as detailed in Eq. (4). Because the entire input set of time series data, i.e., $\{\mathbf{X}_i\}_{i=1}^N$ and $\{\mathbf{Y}_i\}_{i=1}^N$, can be utilized for training, global models can typically be more heavily parameterized than their local counterparts. As a result, such models are more difficult to overfit while offering superior performance for a wider range of input data due to the larger model capacity. While arbitrary forecasting models could be used as the basis for the proposed framework, we select the recent N-BEATS model [11] for this task due to its excellent performance for various time-series prediction problems. However, note that the proposed framework is *model agnostic* and can be implemented with arbitrary backbones.

1) Model Description: Following existing literature [16], [3], we use load lags and categorical features as the input to the forecasting model. The use of load lags allows us to exploit historical data of each time series for the forecasting task, whereas categorical features encoding the month, the day of the week, and the hour in a day, are employed to model seasonality. Load lags are created using a rolling window approach (with K lags) for each time series separately, and are then stacked together into the combined lag feature matrix $\mathbf{X}_{lags}$ for all time series in the input set. Similarly, categorical features for the month, the day of the week, and the hour of the day are created by first extracting relevant information from each time series at timestamp t, and then stacking the values together for all time series in the set. Following established literature [2], [16], [24], one–hot encoding is utilized to generate the categorical feature matrix $\mathbf{X}_{exog}$. Thus, the global model in the proposed framework aims to implement the following mapping:

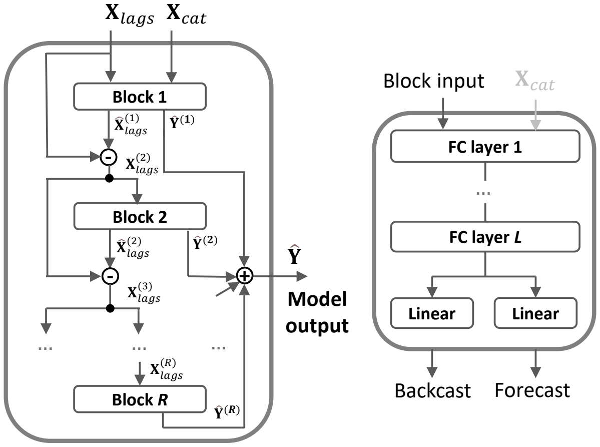


Fig. 2: Architectural details of the extended N-BEATS model: architecture (left), N-BEATS block (right).

$$f : \mathbf{X} \mapsto \mathbf{Y}, \tag{5}$$

where $\mathbf{X} = [\mathbf{X}_{lags}, \mathbf{X}_{exog}]$, and $\mathbf{Y}$ is a target matrix with a horizon of H. To learn the model we use a standard Mean Absolute Error (MAE) learning objective with L1 regularization to control the model complexity, i.e.:

$$\mathcal{L} = ||\mathbf{Y} - f(\mathbf{X}, \theta)||_{L1} + \lambda||\theta||_{L1}, \tag{6}$$

where λ is a regularization factor set to 0.0001 and $\widehat{\mathbf{Y}} = f(\mathbf{X}, \theta)$ are the model predictions.

2) Backbone Model: We use the recent N-BEATS architecture as the basis for the global model $f(\cdot)$ due to its excellent performance for various time-series forecasting tasks [11]. The architecture consists of a series of base architectural blocks linked together through residual connections, as shown in Fig. 2. Each of the R blocks in the model is comprised of L fully-connected (FC) layers with ReLU activations and two linear layers that generate two distinct outputs, i.e.: (1) the (partial) forecast $\mathbf{Y}^{(r)}$, and (2) an estimate of the block's original (lag) input $\widehat{\mathbf{X}}_{lag}$, also called the backcast. The first block in the model takes the lag values $\mathbf{X}_{lag}$ as well as the categorical features $\mathbf{X}_{cat}$ as input, whereas the remaining blocks only accept the (recursive) residuals of the lags, i.e., $\mathbf{X}^{(r)}_{lags} = \mathbf{X}^{(r-1)}_{lags} - \widehat{\mathbf{X}}^{(r-1)}_{lags}$, as the basis for their predictions. Here, r denotes the block index, where $r \in \{1, 2, \dots R\}$. To generate the final prediction for a given input feature matrix $\mathbf{X}$, the N-BEATS model aggregates the forecasting outputs of all R blocks of the model, i.e. [11]:

$$\widehat{\mathbf{Y}} = \sum_{r=1}^{R} \widehat{\mathbf{Y}}^{(r)}. \tag{7}$$

In our extended N-BEATS design, the prediction of the first N-BEATS block $\widehat{\mathbf{Y}}^{(1)}$ makes full use of the historical time-series information but also of the categorical features extracted from the given time-series set. The remaining blocks refine the initial forecast with higher-order predictions based on residual lags values only. It is worth noting that the extended model inherits all characteristics of the original N-BEATS architecture, such as transparent gradient flow during training (due to the residual design) and interpretability due to the aggregation of the partial forecasts $\widehat{\mathbf{Y}}^{(r)}$ in (7).

C. Consumer Clustering

The global forecasting model described in the previous section is readily applicable to all time series from the input set $\mathcal{Y}$. However, given its global nature, the forecasting performance may still be further improved for time-series data with specific characteristics. To accommodate such time series, we propose an (optional) unsupervised localization mechanism that relies on time series clustering. The main idea behind this mechanism is to identify time-series data with common (but distinct) characteristics and then to adapt/localize the learned global model with respect to the identified clusters.

To accommodate the clustering procedure, we first reparameterize the training part of the time series data in $\mathcal{Y}$ and compute descriptive features that encode high-level time-series characteristics for each series in the set. Following the work from [25], we compute the mean, variance, first order of auto-correlation, trend, linearity, and a number of additional features for each time series and use the extracted feature as the basis for clustering[1]. We utilize a hierarchical clustering procedure based on K-means and cluster splitting (akin to the Linde-Buzo-Gray (LBG) algorithm [26]), which results in the following cluster hierarchy:

$$\begin{aligned}
&\mathcal{Y} = \{\pi_0^{(l)} \cup \pi_1^{(l)}\}; l = 1,\\
&\mathcal{Y} = \{\pi_0^{(l)} \cup \pi_1^{(l)} \cup \pi_2^{(l)}\}; l = 2,\\
&\quad \vdots\\
&\mathcal{Y} = \{\pi_0^{(l)} \cup \pi_1^{(l)} \dots \cup \pi_{C-1}^{(l)}\}; l = C-1,
\end{aligned} \tag{8}$$

where $\pi_i^{(l)}$ represents the i-th subset of the time series data at the l-th level of the cluster hierarchy and $\bigcap \pi_i^{(l)} = \emptyset$. Note that a hierarchical clustering procedure is selected for time series partitioning because it offers a convenient and, most importantly, reproducible way of generating time series subsets as opposed to standard approaches that exhibit a certain level of uncertainty due to the initialization procedure.

Algorithm 1: Model localization for cluster hierarchy

```
Input: Cluster hierarchy: $\{\pi_i^{(l)}\}$, for $l = 1, 2, \dots C-1$
       and global model parameters $\theta$
Output: Localized models $\{f_{\pi_i}^{(l)}\}$
for $l = 1$ to $C-1$ do
    for $i = 0$ to $l$ do
        Get subsets $\mathbf{X}_{\pi_i}^{(l)}$ and $\mathbf{Y}_{\pi_i}^{(l)}$ from $\{\mathbf{Y}_k\}_{k=1}^N$
        Initialize the global model $f(\cdot)$ using $\theta$
    Optimize $f(\cdot)$ through Eq. (6) using $\mathbf{X}_{\pi_i}^{(l)}$, $\mathbf{Y}_{\pi_i}^{(l)}$
    Store resulting model $f_{\pi_i}^{(l)}$ with parameters $\theta_{\pi_i}^{(l)}$
```

D. Global Model Localization

The data clusters from Eq. (8) are utilized to localize the global forecasting model by adapting it based on the identified subsets of the overall time-series data. This type of model localization strategy exhibits several desirable characteristics: (i) it is generally applicable and fully unsupervised, i.e., it does not rely on any prior knowledge or data labels, (ii) it still allows to control the forecasting complexity by sharing parameters across groups of times series, and (iii) it enables improved forecasting performance by fine-tuning the initial global model to the specifics of the clustered times series. It is important to note at this point that *localized global models* are formally still global models as they are trained on a set of times series and not a single time series at the time.

Let $\pi_i^{(l)}$ represent the i-th subset of the complete time series data identified by the clustering procedure described above at the l-th level of the hierarchy. Furthermore, let the corresponding feature matrix be denoted as $\mathbf{X}_{\pi_i}^{(l)}$ and the corresponding target matrix as $\mathbf{Y}_{\pi_i}^{(l)}$. The localization strategy used in our framework can then be defined as a model adaptation procedure that estimates a new set of model parameters for the time series in $\pi_i^{(l)}$, so that:

$$f_{\pi}^{(l)}(\mathbf{X}_{\pi_i}^{(l)}, \theta_{\pi_i}^{(l)}) = \mathbf{Y}_{\pi_i}^{(l)}. \tag{9}$$

The resulting model $f_{\pi_i}^{(l)}$ is then a localized version of the global model f and is expected to provide better forecasting performance on the subset $\pi_i^{(l)}$. It needs to be noted that the localized model is learned by initializing the backbone architecture (N-BEATS in this work) with the global model parameters θ and then fine-tuning the model on the data from $\pi_i^{(l)}$. While the localized models could theoretically also be learned from scratch, such a strategy could easily face similar issues as existing local models and be prone to overfitting, poor generalization and local maxima.

The presented localization mechanism is applied to the complete cluster hierarchy from Eq. (8) and generates a hierarchy of localized models, as described in Algorithm 1.

[1] The reader is referred to [25] for details on the complete feature set.

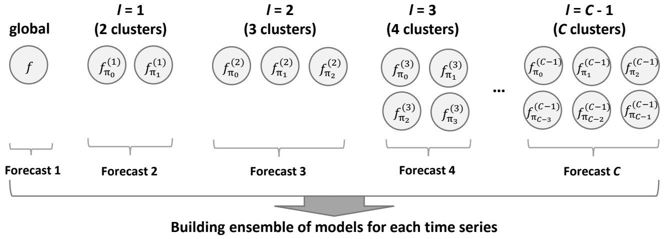


Fig. 3: Illustration of the partial forecasts generated at each level of the calculated cluster hierarchy.

E. Building an Ensemble of Forecasts

Let $\mathbf{x} = [\mathbf{x}_{lag}, \mathbf{x}_{cat}]$ represent a feature vector corresponding to a specific time series of the i-th cluster $\pi_i^{(l)}$ from the l-th level in the cluster hierarchy. The forecast for the feature vector at the l-th level $\widehat{\mathbf{y}}^{(l)}$ is then computed through the following gated superposition, i.e.:

$$\widehat{\mathbf{y}}^{(l)} = \sum_{i=0}^{l} \delta_i(\mathbf{x}) f_{\pi_i}^{(l)}(\mathbf{x}, \theta_{\pi_i}^{(l)}), \tag{10}$$

where the gate $\delta_i(\cdot)$ in the above equation is a Dirac function of the following form:

$$\delta_i(\mathbf{x}) = \begin{cases} 1; & \text{if } \mathbf{x} \in \pi_i^{(l)} \\ 0; & \text{otherwise} \end{cases} \tag{11}$$

To combine the forecasts from the global model and the $C-1$ forecasts from the cluster hierarchy (illustrated in Fig. 3), we utilize a simple bottom-up selection procedure. Following established literature [27], an ensemble of models is created by first generating all C forecasts for all considered time series and evaluating their performance on a (hold out) validation set. Next, a ranked list is generated by sorting the models according to their performance. Finally, a sequential model selection procedure is employed to find the optimal ensemble. This sequential procedure first combines the first and second best-performing models by averaging their forecasts. If the performance on the validation set increases, the third-performing model is added, and the evaluation procedure is repeated. Models are then sequentially added until the performance stops increasing or the entire set of models is exhausted. To ensure optimal performance across the entire time series set, model ensembles are built for each time series in the set separately. The entire procedure is summarized within Algorithm 2.

V. Experiments and Results

In this section, we report experimental results on a real-world dataset that demonstrate the performance and characteristics of the proposed forecasting framework[2]. Specifically, we show experiments and corresponding results that: (i) compare the global modeling framework to its local counterpart, (ii) illustrate the main characteristics of the global forecasting framework, (iii) explore the impact of the proposed localization mechanism, and (iv) benchmark the proposed solution against state-of-the-art techniques from the literature.

Algorithm 2: Ensemble construction procedure

```
Input: Set of $C$ global (localized) models $\{f_{\pi_i}^{(l)}\}_{l=0}^{C-1}$
Output: Set of $N$ optimal ensembles
for $k = 1$ to $N$ do
    Select validation data for $k$-th time series in $\mathbf{Y}_k$
    Generate forecasts for all $C$ models in $\{f_{\pi_i}^{(l)}\}_{l=0}^{C-1}$
      based on Eq. (10)
    Evaluate forecasting performance for all models,
      e.g., using Eq. (14), and generate ranked list
    for $i = 1$ to $C$ do
        if $i = 1$ then
            Select first model from ranked list and
              calculate forecasting error $\Delta$
        else
            Add $i$-th model from list and average
              forecasts from all selected models
            Compute new forecasting error $\Delta_{new}$
            if $\Delta_{new} \geq \Delta$ then
                Stop ensemble construction
    Return optimal ensemble (selection of models)
      for the $k$-th time series
```

A. Dataset Description

We use the dataset published by Commission for Energy Regulation (CER) in Ireland [28] for the experiments. The dataset contains load profiles of over 6000 residential consumers and small & medium enterprises for approximately one and a half years (from July 1, 2009 to December 31, 2010) with a half-hourly resolution. As we are also interested in the forecasting performance on consumer aggregates at the higher network levels, e.g., transformer stations or feeder loads, three additional aggregates, representing *small*, *medium* and *large* transformer stations are created. To generate realistic load aggregates from the initial pool of individual load time series, a total of 1000 time series corresponding to the following groups are created for the experiments, i.e.:

- Individual consumers (single): 250 time series of individual consumers;
- Small transformer stations (sTS): 250 time series each having 50 consumers in an aggregate;
- Medium transformer stations (mTS): 250 time series each having 100 consumers in an aggregate;
- Large transformer stations (lTS) 250 time series each having 200 consumers in an aggregate.

Fig. 4 shows an example of a weekly profile for 4 randomly chosen time series, each taken from one of the considered groups. It can be seen that the variability in the daily profiles decreases with the level of aggregation. The load of the individual consumer (grey line) is highly volatile, whereas the load of the large transformer station, where there are 200 consumers in an aggregate, is much less volatile and is, therefore, expected to be easier to forecast.

In the experiments, we perform STLF with a horizon of $H = 48$ (24 hours) and create dataset samples using a rolling

[2] Python code and data is available at https://github.com/mihagrabner/GlobalModelingFramework

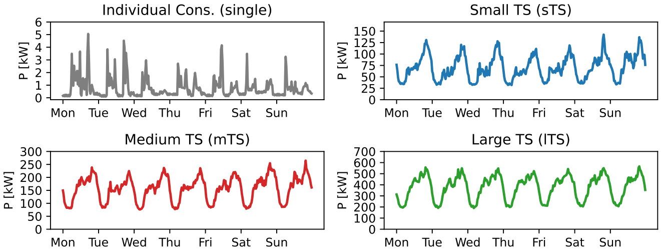


Fig. 4: Examples of time series data used in the experiments.

window approach. We use one year of data for training, the following 12 weeks for validation, and the final 12 weeks for testing. Hyper-parameters are tuned using the validation set and a time-based dataset split is adopted, where the training, validation, and testing sets are consecutive in time.

B. Implementation Details

We learn the parameters of the global model by minimizing the objective from Eq. (6) using the Adam optimizer [29]. To avoid overfitting, we used L1 weight regularization in conjunction with an early stopping criterion by monitoring performance on the validation set. The L1 regularization factor λ is set to 0.0001. When training the initial global model, the learning rate is initially set to $lr = 0.001$ and reduced 3 times by a factor of 10 every time the validation loss plateaus. When training the localized models, a similar procedure that minimizes the learning objective from Eq. (6) is used. The localized models are initialized with the parameters of the global model and then fine-tuned based on the clustered time-series data. We start with a lower learning rate of $lr = 0.0001$ and reduce it by a factor of 10 every 20 epochs. Training is stopped after the validation loss stops decreasing. We consider 30 minute forecasts which result in a dimensionality of the categorical features of 67 (12 for each month, 7 for each day of the week, and 48 for each hour in a day). We also do not share weights across the blocks of the N-BEATS model as advocated by its authors, as this was found to result in better forecasting performance. The model itself is implemented with a total of 3 blocks and 3 FC hidden layers in each block and each FC layer having 512 units.

C. Performance Metrics

Forecasts are evaluated using the Mean Absolute Scaled Error (MASE)[3], which is a standard measure for comparing forecast accuracy across multiple time series [30]. MASE is a scale-independent performance indicator, so it can be used to compare forecasts across different data sets and time series characteristics [8], [11]. Scores below 1 indicate that the generated forecasts outperform a naive (seasonal) model.

TABLE I: Performance evaluation of local and global models. Note that the MASE score (↓) of a naive model is 1.

Modeling framework	Single	sTS	mTS	lTS	All
Local	0.7983	0.7870	0.7553	0.7182	0.7647
Global (ours)	0.8011	0.7546	0.7115	0.6784	0.7364
Improvement [in %]	-0.28	3.24	4.38	3.98	2.83

Let $\mathbf{y}_{i,T_i+1:H} = [y_{i,T_i+1}, y_{i,T_i+2}, ..., y_{i,T_i+H}]^T$ again represent the i-th reference time series of length T_i and with a forecasting horizon of H. Furthermore, let $\widehat{\mathbf{y}}_{i,T_i+1:H} = [\widehat{y}_{i,T_i+1}, \widehat{y}_{i,T_i+2}, ..., \widehat{y}_{i,T_i+H}]^T$ be the corresponding model prediction. MASE is calculated by dividing the Mean Absolute Error (MAE) of a model forecast on the test set with the in-sample MAE of a naive seasonal model calculated over both the training and test data. In our case, the naive seasonal model (further referred as naive benchmark) takes values from a previous week to predict the load in the current week at the same time instance (observations measured one week or S periods in the past). Formally, this can be written as:

$$MAE_{test} = \frac{1}{H} \sum_{h=1}^{H} |y_{T+h} - \widehat{y}_{T+h}| \tag{12}$$

$$MAE_{naive} = \frac{1}{H} \sum_{h=1}^{H} |y_{T+h} - y_{T+h-S}| \tag{13}$$

$$MASE = \frac{MAE_{test}}{MAE_{naive}} \tag{14}$$

Note that we omit the time series index i in the above equations for brevity. In the experiments, MASE scores are computed for every time series in the given set and the average across all time series is then reported as the final performance indicator for the evaluated forecasting models.

D. Global vs. Local Modeling

We first compare the forecasting performance of the proposed global and the established local modeling frameworks to demonstrate the benefits of modeling time series globally and illustrate the main characteristics of the global forecasting models. In the *local* setting, one (N-BEATS) model is trained for each time series, which results in 1000 models trained in total. In the *global* setting, on the other hand, a single (global) model is trained for all 1000 time series. We note that only the first step of the overall framework is considered at this stage (without any model localization or ensembles). To balance model complexity against the amount of available training data and to ensure reasonable generalization, we design the local models with 32 units per layer in each stack of the N-BEATS backbone and use 512 units per layer for the global model. We note that the simpler configuration for the local models was determined through a hyper-parameter optimization process performed during the model development stage. Here, different numbers of units per layer were considered, i.e., 512, 256, 128, 64, 32 and 16. Finally, 32 units per layer were chosen for the experiments, as this configuration yielded the best overall performance, while being less prone to overfitting than the more heavily parameterized versions of the models. With this configuration, each local model has $78,848$ parameters, resulting in a total of $78,848,000$ parameters that have to be estimated for the local modeling framework for the 1000 time

[3] Additional results using the Normalized Mean Absolute Error (NMAE) and Mean Absolute Percentage Error (MAPE) are provided in A.

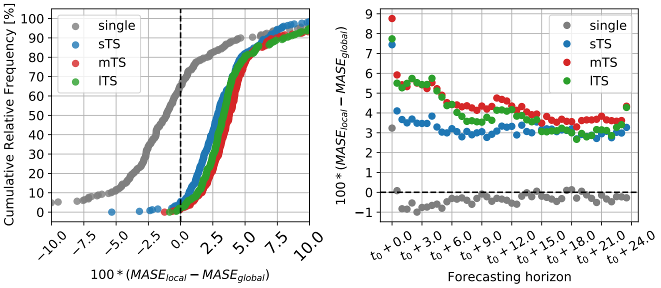


Fig. 5: Global vs. local modeling: overall performance differences for different load aggregates (left), performance differences as a function of the forecasting horizon (right).

series in the input set. The global model, on the other hand, has a total of $2,684,544$ parameters that are shared between all 1000 time series. Thus, the global model needs approximately $30\times$ fewer parameters than the local models in total to forecast the entire set of time series.

1) Overall Performance: The forecasting performances of the global and local models across the different types of consumer aggregates are reported in Table I. As can be seen, the local models as well as the (single) global model significantly outperform the naive baseline and result in MASE scores well below 1. On average, the local models improve on the naive forecasts by 25.5% and the global model by 26.3%. Furthermore, the global modeling framework ensures 2.83% better forecasts on average than the local models in terms of MASE and clearly outperforms its local counterpart for all load aggregates except the individual consumers. For the individual consumers, global modeling performs comparable to the local models, with a minute performance difference of 0.28% in favor of the local solution.

On the left side of Fig. 5 we analyze the performance difference (in percentage) between the global model and the local models for each time series and aggregate type separately using the Empirical Cumulative Relative Frequency (ECDF). Here, each dot represents the performance difference for one time series, where each difference is calculated as $(MASE_{local} - MASE_{global}) \times 100\%$. The global framework performs better than the local one on 35%, 95%, 98% and 98% of the time series for the single, small, medium, and large TSs, respectively. The improvement is significantly better for the small, medium, and large TSs, whereas in the case of single consumers, the local approach performs better. We ascribe this result to the fact that the global model needs to account for a large group of time series and is, therefore, superior on time series that are less volatile. Nonetheless, as we show later, the global model can further improve on the reported performance by utilizing the proposed localization mechanism.

2) Different Horizons: Next, we investigate the behavior of the global and local modeling frameworks w.r.t. the length of the forecasting horizon H. The results on the right of Fig. 5 show that for the first forecasted time step t_0 (for one half-hour ahead), the global approach outperforms the local approach in terms of MASE by 3.2%, 7.4%, 8.7%, and 7.7% for the single, small, medium and large TSs, respectively. The improvement decreases with the increase in the forecasting horizon. With the largest horizon, the improvement of the global approach over the local one saturates between 3 and 4% for the small, medium, and large TSs, whereas for the single consumers, the MASE difference stays between 0 and -0.5%.

TABLE II: Impact of training dataset size on performance

Training data	**Single**	**sTS**	**mTS**	**lTS**	**All**
Whole dataset	0.8011	0.7546	0.7115	0.6784	0.7364
Subsampled by 12×	0.8048	0.7574	0.7146	0.6829	0.7399

3) Dataset Size: Because the global forecasting model is learned on data of all time series in the given input set, an extremely large amount of data is in general available for training. When learning the local models, a standard approach in the load-forecasting literature is to use at least one year of data for training and a fixed window of data for testing. With global modeling, the amount of training data increases linearly with the number of time series in the input set. In our case, this means that the global model has $1000\times$ more samples available for training than each of the local models. As a result, it is expected that such global models can be learned well even if the training data is subsampled. This has important implications for real-world applications, where a portion of data might be missing.

To explore the impact of dataset size (and resolution), we analyze the performance of the global model with smaller training sample sizes per time series. We keep the test data fixed to 3 months, but sample the training data from the entire (one year) data history during the model learning stage. Specifically, we use a subsampling procedure that reduces the amount of training data by a factor of $12\times$.

The results in Table II show that even when using $12\times$ less data points, the performance decreases by only 0.35%, whereas the training time decreases by approximately 4 times, i.e., training the model takes a little less than 2 days instead of 8 on the utilized hardware. Additionally, we observe that even when trained on the subsampled data, the global approach still outperforms the local approach that uses the whole history of data. This represents a highly desirable characteristic that can help to mitigate some of the data quality issues seen with the current smart meter infrastructure, as the model does not need the whole history of data of each smart meter if the training is done within a global modeling framework. Based on this insight, we use the presented subsampling in all following experiments as it significantly decreases the training time, while having a minimal impact on the overall forecasting performance.

4) Scalability and Training Complexity: To investigate how the global modeling framework scales with the number of time series considered, we analyze the training time required to learn the set of local models compared to a single global model in Fig. 6. Here, the red line shows the training time needed by the local models (which is linear in the number of considered time series N) and the blue line the training time for the global model. For the analysis, the global model is trained on 20, 100, 500 and 1000 time series (blue dots). We can see that even with 20 considered time series, the global model already has

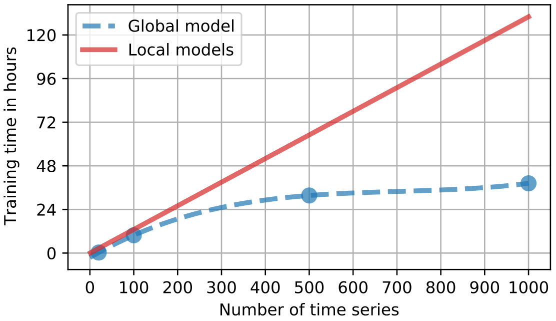


Fig. 6: Scalability analysis.

TABLE III: Performance on a new dataset using MASE.

Forecasting setting	Single	sTS	mTS	lTS	All
Without historical data	0.7997	0.7519	0.7366	0.7149	0.7508
With historical data (retrained)	0.7948	0.7489	0.7254	0.7025	0.7429

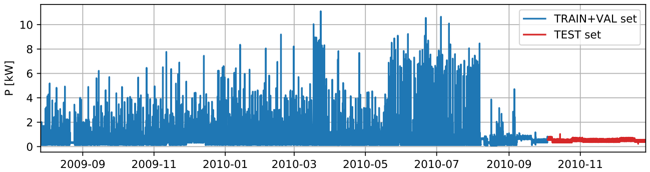


Fig. 7: Example of a change in load behavior of a consumer.

an edge over the local models in terms of training time. With the largest set of time series considered, training of the global model is $3.3\times$ faster than training 1000 separate local models.

E. Characteristics of the Global Modeling Framework

In the next experimental series, we demonstrate some of the key characteristics of the global approach to load forecasting. We again use a single global model (without localization) to make predictions over different load aggregates.

1) Forecasting without Historical Data: Conventional (local) load-forecasting models require historical data of a given time series to be able to learn a prediction model. Here we show that even for time series without historical observations, global models are capable of generating accurate forecasts, which is a unique characteristic of the proposed framework. In practical scenarios, this characteristic allows making forecasts for new consumers or transformer stations as soon as a single observation window (one week of data) becomes available. To demonstrate this powerful capability of the global model, we generate a new set of 1000 time series that does not overlap with the original data but corresponds to valid individual consumers and consumer aggregates. We use a similar data-generation procedure, as described in Section V-A. Next, we take the global model trained on the original CER data and forecast the load over the newly generated (unseen) time series. The results of this experiment are presented in the first row of Table III, where it can be seen that the global model outperforms the naive baseline by approx. 25% on average and, hence, exhibits impressive generalization capabilities. To put these results into perspective, we also fine-tune (retrain) the global model on the newly generated dataset using historical data of the newly considered time series. We observe that after retraining the performance increases by an additional 0.8%, suggesting that having historical observations for each load is important, but also that competitive predictions can be produced for unobserved time series. This observation has important implications of the industry, as it allows forecasting time series with no historical data (e.g. new consumers), which is not possible with current local modeling techniques.

2) Change in Load Behavior: Another desirable characteristic of the proposed global model is its ability to handle change in load behavior. Because global models can efficiently leverage cross-series information by training on multiple (diverse) series, such models can accurately forecast time-series even if the behaviour of the series changes. This is very welcome in reality as consumers might change electrical appliances throughout time (e.g. using a heat pump for heating instead of natural gas or going on vacations for a longer time period) or transformer stations and feeders get reconfigured. In Fig. 7 we show an example of a consumer that for some reason significantly changed behavior after October 2010. Only a small amount of this new behaviour was captured in the data available for training the forecasting models (marked blue; labeled TRAIN+VAL set). In Fig. 8 we show the actual load, and the predictions of the global and local models for a selected (test) week in December 2010. Note how the global model accurately predicts the new behaviour, whereas the local model fails to properly account for the unexpected change.

F. Model Localization Mechanism

In the next series of experiments, we consider the entire proposed framework, including the model localization mechanism and final ensembles. The presented experiments: (i) demonstrate the impact of the localization step, (ii) ablate parts of the framework to evaluate the effect of various components on performance, and (iii) study the clustering procedure and compare it to selected alternatives.

1) Model Localization: First, we explore the effect of the proposed model localization strategy on the overall forecasting performance. To this end, we start with the global model evaluated in the previous subsections and localize it based on subsets of the time series data identified through our clustering procedure. Finally, we build an ensemble of forecasts using the procedure described in Subsection IV-E. For the experiments, we use up to 20 clusters in the generated cluster hierarchy, i.e., $C = 20$, $l = 19$, which results in 209 localized models. As we show later, using 20 clusters ensures a sufficient level of granularity for the studied forecasting problem.

The results of the experiments are presented in Fig. 9. Here, the left part of the figure shows the percentage improvement observed due to the localization procedure for each aggregate type using ECDF plots. It can be seen that for almost all the series, localization improves the forecasting performance. The most significant improvement is for individual consumers (grey dots), where the infusion of additional information through the localization procedure contributes towards much more convincing forecasting results. To get better insight into the impact of the proposed localization strategy, we show on

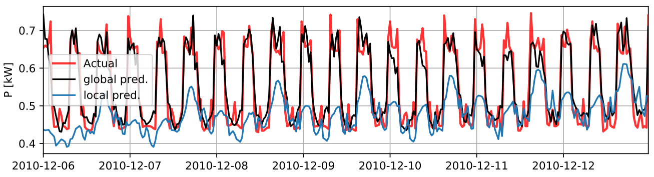


Fig. 8: Forecasting example after change in load behaviour.

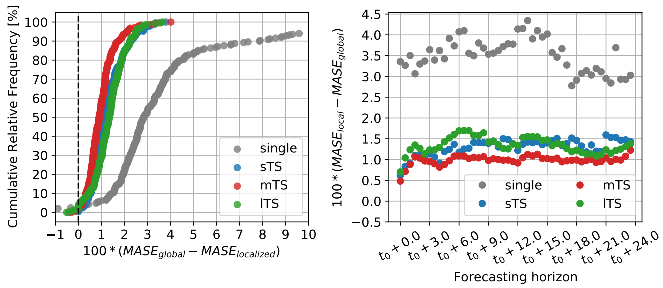


Fig. 9: Impact of localization: overall performance differences for different load aggregates (left), performance differences as a function of the forecasting horizon (right).

the right part of Fig. 9 the improvements caused by localization as a function of the length of the forecasting horizon H. It can be seen that localization significantly improves forecasting performance on all horizons and for all aggregate types.

While the localization procedure is beneficial for the forecasting performance, it needs to be noted that the localization process introduces additional computational overhead into the training and inference processes. The most time-consuming part of the localization framework is the fine-tuning of the 209 localized models. On our experimental hardware (i.e., a computing server with 8 CPU Intel Xeon cores with an all-core turbo frequency of 3.5GHz), training the initial global model takes a little less than 2 days (as shown in Fig. 6), whereas fine-tuning the 209 localized models takes an additional 2 days. Thus, training the whole localization framework (end-to-end time) takes a little less than 4 days. In comparison with the local modeling, training the localized models in the optimal ensemble is still approximately $1.5\times$ faster. It is also worth mentioning that the clustering procedure is done offline and only once during the training stage. The total computational time needed for clustering is negligible (around 1 minute on our hardware) compared to the time needed for training or fine-tuning the deep-learning models.

2) Design Alternatives and Ablations: To demonstrate the efficacy (and contribution) of the design choices made with the proposed forecasting approach, we consider various other alternatives and benchmark them against the proposed solution. In general, the performance of the localization strategies relying on data clustering depends on: (i) the representation of each time series supplied to the clustering algorithm, (ii) the characteristics of the selected clustering algorithm, and (iii) the way the final forecasts are generated/combined from the identified clusters. We, therefore, evaluate different options for each of these factors, i.e.:

- *Representation.* We use feature-based clustering for all experiments, where a set of features is first extracted from the training part of the time series data $\mathcal{Y}$ and a clustering procedure is applied on top of the extracted features. We explore features based on time-series characteristics as described in Subsection IV-C (denoted as TF) but also features based on mean daily profiles, where each time series is represented with a vector of size 3×48, consisting of a mean daily profile of a working day, Saturdays and Sundays, that are stacked together (denoted as DF) [31].
- *Clustering Algorithm.* We evaluate two clustering algorithms: K-Means and DBSCAN [32]. Preliminary experiments show that DBSCAN cannot identify any useful clusters when representing time series with mean daily profiles; therefore, we apply DBSCAN only on features based on time-series characteristics. As a result, three clustering approaches are considered for the comparative analysis: (i) K-Means using features based on mean daily profiles, (ii) K-Means using time-series characteristics (as proposed for our framework), and (iii) DBSCAN using time-series characteristics.
- *Forecast Generation.* When generating forecasts from the localized model, we investigate three competing strategies, where we: (i) identify the optimal number of clusters for all time series in the input set (ALL hereafter) and make predictions in accordance with Eq. (10) using the same fixed number of clusters for all given time series, (ii) identify the optimal number of clusters for each time series in the input set separately and make predictions based on Eq. (10) using a different input set partitioning for each time series (BEST hereafter), and (iii) create an ensemble from the cluster hierarchy, as proposed for our framework (denoted as ENS).

The results of the experiments with all model variants are presented in Table IV. To further demonstrate the value of the proposed localization procedure, we additionally include results for the naive localized global model denoted as *Singleton*, where each time series is considered as a separate cluster and the global model is localized only with the training data of the time series, on which the forecasts are performed. This configuration results in N localized models, where N is again the number of time series in the input set. For comparison purposes, we also report *Baseline* results obtained with the global model trained without localization on the subsampled training data (see Table II).

We can see that fine-tuning the global model for each time series separately (Singleton) significantly degrades the performance (MASE increases by 4.46%) and leads to even worse results than the local models in Table I. Additionally, we observe that in all three cases, localizing the global model using groups of similar time series performs better than using the localization procedure on each time series separately (Singleton) which can easily lead to over-fitting. Overall, we observe that: (i) model localization improves performance over the baseline global model, (ii) features based on time-series characteristics produce better results than fea-

TABLE IV: Performance comparison of different model localization strategies in terms of MASE. Results are reported for two different clustering algorithms, two time series representations, three forecast generation/combination strategies and two additional baselines. Results for the configuration proposed in this paper are shaded grey. The best results are shown in bold.

Clustering Algorithm	Representation	Forecast Generation	Single	sTS	mTS	lTS	All
Baseline	n/a	n/a	0.8048	0.7574	0.7146	0.6829	0.7399
Singleton (naive)	n/a	n/a	0.7828	0.7994	0.7844	0.7713	0.7845
K-Means	DF	ALL	0.7939	0.7595	0.7211	0.6888	0.7408
K-Means	DF	BEST	0.7844	0.7518	0.7121	0.6802	0.7321
K-Means	DF	ENS	0.7823	0.7460	0.7058	0.6712	0.7263
DBSCAN	TF	ALL	0.8048	0.7574	0.7146	0.6829	0.7399
DBSCAN	TF	BEST	0.7795	0.7575	0.7146	0.6829	0.7336
DBSCAN	TF	ENS	0.7757	0.7550	0.7154	0.6844	0.7326
K-Means	TF	ALL	0.7795	0.7574	0.7231	0.6912	0.7378
K-Means	TF	BEST	0.7753	0.7540	0.7160	0.6845	0.7325
K-Means	TF	ENS	**0.7698**	**0.7444**	**0.7047**	**0.6694**	**0.7221**
Improvement [%]			3.50	1.30	0.99	1.35	1.78

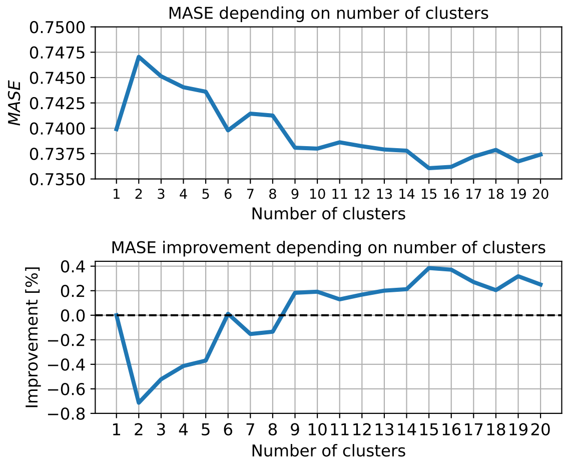


Fig. 10: Forecasting performance as a function of the number of clusters C used by the localization mechanism.

tures based on daily profiles, and (iii) the ensemble approach consistently yields the lowest overall MASE scores among all considered forecast-generation strategies. Considering all tested model variants, it can be concluded that the proposed localization approach (K-Means+TF+ENS – shaded gray) and the corresponding design choices all contribute to the increased forecasting performance of the proposed framework and lead to the best overall results. The bottom line in Table IV shows the improvement of the proposed approach over the baseline. As can be seen, the biggest improvements are observed when forecasting the load of individual consumers, with a performance gain of 3.5% over the baseline.

3) Understanding the Clustering: The localization strategy used in the proposed modeling framework relies on time series clustering. To better understand the impact of the clustering procedure, we conduct a detailed analysis and generate forecasts with different numbers of clusters considered in the prediction step. Note that because historical observations for each time series are typically available, such an analysis can be considered as a hyper-parameter optimization procedure that is performed once (off-line) as part of the training. The results in the top part of Fig. 10 show the value of MASE as a function of the number of clusters for this approach, whereas the results in the bottom part show the improvement over the (baseline) global model without localization. It can be seen that the localization mechanism utilizing between 2 and 8 clusters actually worsens performance. Conversely, performance starts increasing when the number of clusters is at least 9. The best results are observed with 15 clusters, where the performance is improved by 0.4%. These results suggest that it is critical to identify a suitable data granularity for the clustering procedure, as performance gains can only be expected with a sufficient number of clusters utilized for the localization step. In addition to performance, an important consideration is the number of parameters of the localized ensemble. Because the global model used in our experiments has a total of $2,684,544$ parameters, the entire ensemble has $M \cdot 2,684,544$ parameters, where M denotes the number of localized models used and depends on the selected number of clusters C. Thus, if space complexity is an issue, a trade-off between performance and parameter count can be sought when selecting the optimal number of clusters.

Next, we analyze how well the selection of a particular number of clusters performs across the time series in the input set. For this experiment, we first determine the optimal number of clusters for each time series separately using the K-Means+TF+BEST strategy and then observe for what fraction of time series a certain number of clusters results in the best overall performance. The results of this analysis are reported in Fig. 11 - separately for each aggregate type. It can be seen that for the individual consumers, for example, the model localization procedure with 19 clusters is the optimal choice for 14% of the time series in this group. In the case of sTSs, mTSs and lTSs, we can see that a single cluster (i.e., the global model without localization) provides the best forecasting performance in most cases (for 24.8%, 56.4%, 75.6% of all cases for the sTSs, mTSs and lTSs, respectively). These results suggest that model localization with a larger number of time-series subsets is more important for more volatile time series, but also that an adaptive procedure is needed to ensure good performance across time series with diverse characteristics, such as those

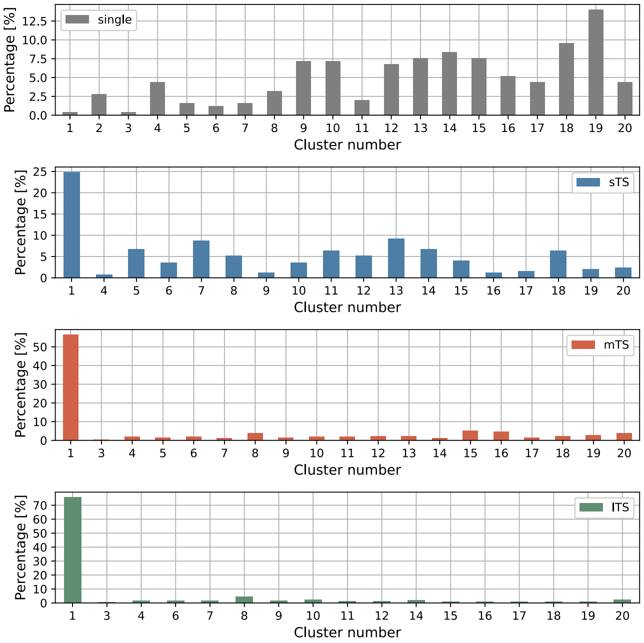


Fig. 11: Percentage of time series, for which a certain number of clusters leads to optimal performance with the K-Means+TF+BEST model localization strategy. Results are presented for each aggregate type separately.

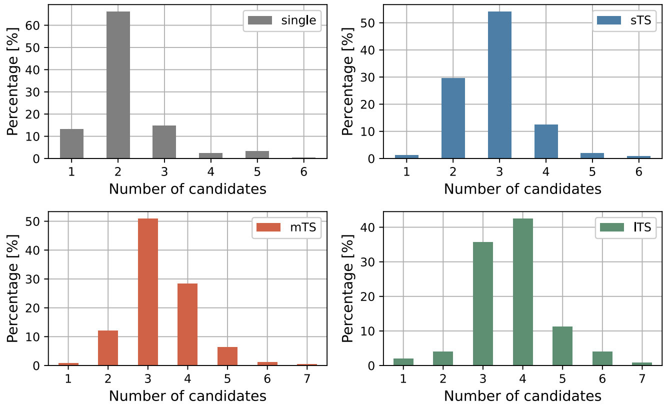


Fig. 12: Percentage of time series, for which a combination of a certain number of models leads to optimal performance with the proposed K-Means+TF+ENS model localization strategy. Results are presented for each aggregate type separately.

caused by different aggregate types. Such a procedure is, for example, provided by the proposed ensemble approach.

Last but not least, we perform a detailed analysis of the proposed ensemble approach, i.e., K-Means+TF+ENS. Our approach averages the partial forecasts from Eq. (10) and[4], therefore, combines the predictions from multiple localized models that capture the characteristics of the input time series data at different levels of granularity. The results in Fig. 12 again show the fraction of time series, for which the combination of a certain number of models results in optimal performance. It can be seen that for 66% of time series, corresponding to individual consumers, the combination of models from two levels in the cluster hierarchy in the ensemble yields the best forecasts. For sTSs and mTSs the combination of models from three levels is optimal for 54% and 50% of the time series, respectively. In the case of lTSs, the combination of 4 models is optimal for 42.4% of the time series. These results suggest that combining models using an adaptive strategy that is able to account for specific data characteristic is important for the final forecasting performance and is, therefore, utilized in the proposed approach.

G. Comparison with the State-of-the-Art

In the last experimental series, we compare the proposed global modeling framework, i.e., with and without localization, with the following competing state-of-the-art models:

- **Local-MLR**, which is a popular benchmark, originally proposed in [33]. The model is based on Multiple-Linear Regression (MLR) and instead of temperature as in [33], uses the last 48 lag values in an auto-regressive setting. One model is trained for each forecasting step, resulting in 48 models for each time series, and 48×1000 models in total due to the local nature of this approach.
- **Local-LSTM** from [2], which is an LSTM recurrent neural network-based framework that utilizes a local modeling approach. A total of 1000 local models are, therefore, trained for the evaluation.
- **N-BEATS** proposed in [34], but without the exogenous features added in this paper. Instead of ensembling multiple models with different lags and weight initializations, we implement this model with L1 regularization to control model complexity.
- **K-means-LSTM** proposed in [21], which is a model that employs K-means clustering to determine clusters of similar loads and then applies an LSTM network for the forecasting task. Since the forecasting model is applied over pooled data, this can be seen as an intermediate framework between local and global models.
- **DeepAR** proposed in [12], which is an encoder-decoder framework based on Long Short-Term Memory (LSTM) cells. For this model, the decoder part is modified to enable multi-step horizon prediction. Additionally, we apply our proposed localization procedure to the original DeepAR model to demonstrate its benefit beyond N-BEATS.

The comparison of the proposed approach (with and without localization) in terms of average MASE scores on different load aggregates and the competing state-of-the-art algorithms is presented in Table V. The best result for each load is marked in bold, and the second-best is underlined. It can be seen that on average (column All), the proposed global model without localization already outperforms the majority of reference models. With the localization step included (for $C = 20, l = 19$), the performance increases even further. If we analyze the performance for each aggregate type separately, we can see that the original DeepAR model (without

[4] Recall that each partial forecast is generated from the localized models from one level in the cluster hierarchy.

TABLE V: Comparison with state-of-the-art forecasting models (MASE). The MASE (↓) of naive predictions is 1. The best result for each load aggregate is marked in bold, the second-best is underlined.

Modeling Approach	Model	Single	sTS	mTS	lTS	All
Local	Local-MLR [33]	0.9333	0.8933	0.9045	0.9069	0.9095
Local	Local-LSTM [2]	0.8334	0.8400	0.8133	0.8024	0.8223
Local	Local-N-BEATS†	0.7983	0.7870	0.7553	0.7182	0.7647
Global	N-BEATS [34]	0.8306	0.7775	0.7253	0.6920	0.7564
Global‡	K-means-LSTM [21]	0.7705	0.7858	0.7234	0.6828	0.7406
Global	DeepAR (w/o localization) [12]	0.7725	0.7891	0.7241	0.6832	0.7422
Global	DeepAR (w/ localization)	**0.7642**	0.7749	0.7174	0.6763	0.7332
Global	Ours (w/o localization)	0.8048	0.7574	0.7146	0.6829	0.7399
Global	Ours (w/ localization)	0.7698	**0.7444**	**0.7047**	**0.6694**	**0.7221**

† This model represents the local N-BEATS version from Table I.

‡ This pooling-based approach is formally a global procedure applied over a subset of time series data at the time.

localization) outperforms our initial global model (without localization) only on the time series corresponding to the single consumers, whereas after localization, our approach outperforms the original DeepAR model also on this type of time series. We also observe that the inclusion of categorical features is beneficial for performance. The proposed global approach based on the extended N-BEATS model outperforms the original architecture by 1.6%, whereas after localization, this gap is further increased to 3.4%. The results also clearly point to the superiority of the global models (ours, N-BEATS, DeepAR, K-means-LSTM) over the local competitors (Local-MLR, Local-LSTM, and Local-N-BEATS).

Additionally, we also show that our (localization) framework is model agnostic by applying the proposed localization mechanism to DeepAR. If we compare DeepAR with localization and DeepAR without the proposed localization in Table V, we see that this model can also greatly benefit from the localization process. The performance is improved for all aggregate types due to the localization process, but is overall, still below the localized version of our model.

VI. Conclusions

The load forecasting literature is currently dominated by forecasting techniques that rely on local modeling, where each given time series is modeled independently of all others. While such techniques have shown good performance for STLF tasks, the expected scale of future smart grids will soon render them impractical. To address this shortcoming, we presented in this paper a novel framework for STLF based on deep learning that, different from existing solutions, relies on global modeling to capture the characteristics of a large group of time series simultaneously. Based on this framework, a novel approach to load forecasting was introduced that not only utilizes global time-series modeling, but also adopts a powerful model localization strategy. The proposed approach was evaluated in comprehensive experiments and in comparison to state-of-the-art techniques from the literature.

Appendix

In the main part of the paper, we reported all results only in terms of MASE scores to keep the presentation uncluttered. In this Appendix, we now also report the main results in terms of additional performance indicators regularly used in the STLF literature, i.e., the Normalized Mean Average Error (NMAE) & the Mean Absolute Percentage Error (MAPE) [24]. Since both performance scores correspond to error measures, lower scores again indicate better forecasting performance. The results reported in this section look at: (i) the comparison between local and global modeling for STLF, (ii) the impact of model localization on performance, (iii) an ablation study, and (iv) additional analyses related to the clustering procedure that forms the basis for the model localization strategies used in the proposed framework. Implementation detail, needed to better understand details of the trained models are also presented in this Appendix.

A. Local vs. Global Modeling

The first results in Tables VI and VII compare the performance of the proposed global modeling approach to its local counterpart in terms of MAPE and NMAE, respectively, using the same experimental setup as in Section V-D. Thus, the global model is built based on the first step of the proposed framework only, i.e., without the localization mechanism. Note that for the MAPE results, we don't report forecasting performance for the individual consumers, as this error cannot be computed for a single time series. For the NMAE error, performance is reported for all aggregate types.

TABLE VI: Performance evaluation of the local and global modeling frameworks to STLF using MAPE. All† indicates the mean MAPE (↓) across all aggregate types but without the individual consumers.

Modeling framework	Single	sTS	mTS	lTS	All†
Local modeling	n/a	10.2987	8.9845	7.5793	8.9542
Global modeling	n/a	9.7199	8.2061	6.8396	8.2552

TABLE VII: Performance evaluation of the local and global modeling frameworks to STLF using NMAE (↓).

Modeling framework	Single	sTS	mTS	lTS	All
Local modeling	0.5688	0.1069	0.0910	0.0746	0.2103
Global modeling	0.5768	0.1025	0.0857	0.0701	0.2088

Overall, the results show similar behavior as with the MASE scores. Global modeling in general improves performance for

TABLE VIII: Impact of the model localization procedure on forecasting performance. All† indicates the mean MAPE (↓) across all aggregate types but without the individual consumers.

Modeling framework	**Single**	**sTS**	**mTS**	**lTS**	**All†**
Ours (w/o localization)	n/a	9.7633	8.2500	6.8993	8.3042
Ours (w localization)	n/a	9.5996	8.1377	6.7692	8.1688

TABLE IX: Impact of the model localization procedure on forecasting performance using NMAE (↓).

Modeling framework	**Single**	**sTS**	**mTS**	**lTS**	**All**
Ours (w/o localization)	0.5794	0.1029	0.0860	0.0706	0.2097
Ours (w localization)	0.5536	0.1012	0.0848	0.0692	0.2022

all aggregate types with both MAPE and NMAE scores. The only exception here are the results for the individual consumers when forecasting performance is measured in terms of NMAE. Due to the variability of this type of time series, the local modeling has a slight edge over the global modeling. However, as we show in the next section, using a localization procedure on top of the global model results in superior performance on this type of time series as well.

B. Impact of Localization

To further demonstrate the importance of localization in the global modeling framework, Tables VIII and IX summarize the forecasting performance of the global modeling framework with and without localization in terms of MAPE and NMAE scores, respectively. The reported results correspond to the proposed ensemble-based localization procedure.

As can be seen, the localization improves performance across all aggregate types, with the biggest performance gains being observed for the time series that correspond to the individual consumers (Single). These observations are consistent with the observations made in the main part of the paper, where MASE was used as a performance indicator instead of MAPE and NMAE. The results show that the global model, after localization (as proposed in this paper), convincingly outperforms the local modeling framework regardless of the aggregate type.

References

[1] T. Hong, P. Pinson, Y. Wang, R. Weron, D. Yang, and H. Zareipour, "Energy forecasting: A review and outlook," *IEEE Open Access Journal of Power and Energy*, 2020.

[2] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on lstm recurrent neural network," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, 2017.

[3] S. Haben, S. Arora, G. Giasemidis, M. Voss, and D. Vukadinović Greetham, "Review of low voltage load forecasting: Methods, applications, and recommendations," *Applied Energy*, vol. 304, 2021.

[4] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, and J. He, "Short-term load forecasting with deep residual networks," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3943–3952, 2019.

[5] J. Xie, T. Hong, T. Laing, and C. Kang, "On normality assumption in residual simulation for probabilistic load forecasting," *IEEE Trans. Smart Grid*, vol. 8, no. 3, pp. 1046–1053, 2015.

[6] S. Haben, G. Giasemidis, F. Ziel, and S. Arora, "Short term load forecasting and the effect of temperature at the low voltage level," *International Journal of Forecasting*, pp. 1469–1484, 2019.

[7] R. Sevlian and R. Rajagopal, "A scaling law for short term load forecasting on varying levels of aggregation," *International Journal of Electrical Power & Energy Systems*, vol. 98, pp. 350–361, 2018.

[8] P. Montero-Manso and R. J. Hyndman, "Principles and algorithms for forecasting groups of time series: Locality and globality," *International Journal of Forecasting*, 2021.

[9] N. Laptev, J. Yosinski, L. E. Li, and S. Smyl, "Time-series extreme event forecasting with neural networks at Uber," in *International conference on machine learning*, vol. 34, 2017, pp. 1–5.

[10] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "M5 accuracy competition: Results, findings, and conclusions," *International Journal of Forecasting*, 2022.

[11] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting," in *International Conference on Learning Representations*, 2020.

[12] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks," *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[13] S. Smyl, "A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting," *International Journal of Forecasting*, vol. 36, no. 1, pp. 75–85, 2020.

[14] Y. Chen, Y. Kang, Y. Chen, and Z. Wang, "Probabilistic forecasting with temporal convolutional neural network," *Neurocomputing*, 2020.

[15] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 3125–3148, 2018.

[16] Y. Wang, D. Gan, M. Sun, N. Zhang, Z. Lu, and C. Kang, "Probabilistic Individual Load Forecasting Using Pinball Loss Guided LSTM," *Applied Energy*, vol. 235, pp. 10–20, 2019.

[17] M. Tan, S. Yuan, S. Li, Y. Su, H. Li, and F. He, "Ultra-short-term industrial power demand forecasting using lstm based hybrid ensemble learning," *IEEE Trans. power systems*, vol. 35, no. 4, 2019.

[18] H. Shi, M. Xu, and R. Li, "Deep learning for household load forecasting—a novel pooling deep rnn," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5271–5280, 2017.

[19] M. Voß, C. Bender-Saebelkampf, and S. Albayrak, "Residential short-term load forecasting using convolutional neural networks," in *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2018, pp. 1–6.

[20] C. Wang, Y. Wang, Z. Ding, T. Zheng, J. Hu, and K. Zhang, "A transformer-based method of multi-energy load forecasting in integrated energy system," *IEEE Trans. Smart Grid*, 2022.

[21] F. Han, T. Pu, M. Li, and G. Taylor, "Short-term forecasting of individual residential load based on deep learning and k-means clustering," *CSEE Journal of Power and Energy Systems*, vol. 7, no. 2, pp. 261–269, 2020.

[22] E. Yang and C.-H. Youn, "Individual load forecasting for multi-customers with distribution-aware temporal pooling," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, 2021, pp. 1–10.

[23] S. B. Taieb, J. W. Taylor, and R. J. Hyndman, "Hierarchical probabilistic forecasting of electricity demand with smart meter data," *Journal of the American Statistical Association*, vol. 116, no. 533, pp. 27–43, 2021.

[24] P. Wang, B. Liu, and T. Hong, "Electric load forecasting with recency effect: A big data approach," *International Journal of Forecasting*, vol. 32, no. 3, pp. 585–597, 2016.

[25] R. J. Hyndman, E. Wang, and N. Laptev, "Large-scale unusual time series detection," in *IEEE international conference on data mining workshop (ICDMW)*, 2015, pp. 1616–1619.

[26] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Trans. communications*, pp. 84–95, 1980.

[27] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2018.

[28] Commission for Energy Regulation. CER smart metering project - electricity customer behaviour trial, 2009-2010.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International journal of forecasting*, pp. 679–688, 2006.

[31] S. Haben, C. Singleton, and P. Grindrod, "Analysis and clustering of residential customers energy behavioral demand using smart meter data," *IEEE Trans. smart grid*, vol. 7, no. 1, pp. 136–144, 2015.

[32] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[33] T. Hong, P. Wang, and H. L. Willis, "A naïve multiple linear regression benchmark for short term load forecasting," in *2011 IEEE Power and Energy Society General Meeting*, 2011, pp. 1–6.

[34] B. N. Oreshkin, G. Dudek, P. Pełka, and E. Turkina, "N-beats neural network for mid-term electricity load forecasting," *Applied Energy*, vol. 293, p. 116918, 2021.

Miha Grabner received his Master's degree from the Faculty of Electrical Engineering, University of Ljubljana in 2013 and is currently finishing his Ph.D. at the same faculty. He worked as a lead data scientist and researcher at Electric Power Research Institute Milan Vidmar for nine years, where his work was divided between managing projects, consulting, and research in the field of Data Analytics and Machine Learning in Smart Grids.

Miha is currently Head of Energy Forecasting at Ubivivo - a data intelligence company that leverages space technologies and AI for situational awareness and forecasting.

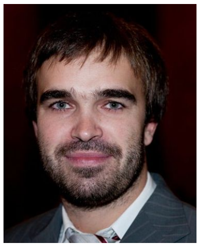
Yi Wang received the B.S. degree from Huazhong University of Science and Technology in June 2014, and the Ph.D. degree from Tsinghua University in January 2019. He was a visiting student with the University of Washington from March 2017 to April 2018. He served as a Postdoctoral Researcher in the Power Systems Laboratory, ETH Zurich from February 2019 to August 2021.

He is currently an Assistant Professor with the Department of Electrical and Electronic Engineering, The University of Hong Kong. His research interests include data analytics in smart grids, energy forecasting, multi-energy systems, Internet-of-things, cyber-physical-social energy systems.

Qingsong Wen (SM'23) is currently a Staff Engineer and Manager at DAMO Academy-Decision Intelligence Lab, Alibaba Group, working in the areas of intelligent time series analysis, data-driven intelligence decisions, machine learning, and signal processing. Before that, he worked at Qualcomm and Marvell in the areas of big data and signal processing, and received his M.S. and Ph.D. degrees in Electrical and Computer Engineering from Georgia Institute of Technology, Atlanta, USA. He has published over 50 top-ranked journal and conference papers, received AAAI/IAAI 2023 Deployed Application Award, and won the First Place in 2022 ICASSP Grand Challenge Competition (AIOps in Networks). He is an Associate Editor for Neurocomputing, Guest Editor for Applied Energy, and regularly served as an SPC/PC member of the major AI and signal processing conferences including AAAI, IJCAI, KDD, ICDM, GLOBECOM, EUSIPCO, etc.

Boštjan Blažič received the B.Sc., M.Sc. and Ph.D. degrees, all in electrical engineering, from the University of Ljubljana, Ljubljana, Slovenia, in 2000, 2003 and 2005, respectively. Currently he is working as a professor at the Faculty of Electrical Engineering, in the Laboratory of Electricity Networks and Devices.

His research work is focused on three main areas: smart grids, mathematical analysis and control of power converters (active filters, FACTS devices…) and power quality (sources and propagation of disturbances, mitigation of power quality problems…). His main competences lie in the field of modelling and simulations of power systems, network components and power converters.



Vitomir Štruc is a full Professor at the University of Ljubljana, Slovenia. His research interests include problems related to biometrics, computer vision, image processing, pattern recognition and machine learning. He (co-)authored more than 150 research papers for leading international peer reviewed journals and conferences in these and related areas. He served in different capacities on the organizing committees of several top-tier vision conferences, including IEEE Face and Gesture, ICB, WACV and IJCB. Vitomir is a Senior Area Editor for the IEEE Transactions on Information Forensics and Security, a Subject Editor for Elsevier's Signal Processing and an Associate Editor for Pattern Recognition, the EURASIP Journal on Image & Video Processing and IET Biometrics. He served as an Area Chair for WACV, ICPR Eusipco and FG and as the Program Chair for ISPA 2019, IWBF 2022, 2023 and IJCB 2020. Currently, he acts as a General Co-Chair for IEEE IJCB 2023 and a Program Co-Chair for IEEE FG 2024. Dr. Struc is a Senior member of the IEEE, a member of IAPR, EURASIP, Slovenia's ambassador for the European Association for Biometrics (EAB) and the former president and current executive committee member of the Slovenian Pattern Recognition Society, the Slovenian branch of IAPR. Vitomir is also the VP Technical Activities for the IEEE Biometrics Council 2022-2024.