Cluster-based Aggregate Load Forecasting with Deep Neural Networks

Andrea Cini
Università della Svizzera italiana
Lugano, Switzerland
andrea.cini@usi.ch

Slobodan Lukovic
Università della Svizzera italiana
Lugano, Switzerland
slobodan.lukovic@usi.ch

Cesare Alippi
Politecnico di Milano
Milan, Italy
Università della Svizzera italiana
Lugano, Switzerland
cesare.alippi@polimi.it

***Abstract*—Highly accurate power demand forecasting represents one of key challenges of Smart Grid applications. In this setting, a large number of Smart Meters produces huge amounts of data that need to be processed to predict the load requested by the grid. Due to the high dimensionality of the problem, this often results in the adoption of simple aggregation strategies for the power that fail in capturing the relational information existing among the different types of user. A possible alternative, known as Cluster-based Aggregate Forecasting, consists in clustering the load profiles and, on top of that, building predictors of the aggregate at the cluster-level. In this work we explore the technique in the context of predictors based on deep recurrent neural networks and address the scalability issues presenting neural architectures adequate to process cluster-level aggregates. The proposed methods are finally evaluated both on a publicly available benchmark and a heterogenous dataset of Smart Meter data from an entire, medium-sized, Swiss town.**

***Index Terms*—short-term load forecasting, smart grid, deep learning, time-series forecasting, time-series clustering**

I. Introduction

The diffusion of smart metering infrastructures and the wide adoption of the Smart Grid paradigm ([1], [2]) are making available a large amount of data in the form of load profiles, i.e., time-series describing the energy consumption of a specific customer as measured by a smart meter (SM). Accurate Short-term Load Forecasting (STLF) is an important ingredient for the development of smart power grids. In fact, an accurate estimate of the energy demand facilitates effective planning of grid operations, maintenance of the grid assets and efficient energy distribution [3]. Furthermore, improved forecasting techniques may significantly contribute to the development of effective Demand Side Management strategies [4]. We focus on the day-head prediction task, i.e., we aim at predicting the load on the grid for each time-step of the next 24 hours.

Despite the granularity of the available data, predicting the load at the level of a single SM is impractical both for privacy, scalability and accuracy constraints. In fact, a typical power load consumption signal is erratic at single household and characterized by high frequency components. The common practice is to simply aggregate the load profiles and consider predictors only of the total load; this preserves privacy as well as reduces fluctuations associated with unpredictable behaviors. While this is effective and can lead to good results [5], the standard approach does not take into account the topological and, more importantly, functional affinities that exist among households, commercial buildings and industrial facilities and that could be exploited to forecast the short-term load on the grid. Improvements can come by considering Cluster-based Aggregate Forecasting (CBAF) [6]. CBAF methods rely on clustering algorithms to partition the customer-base in homogeneous subgroups of users and exploit the similarities among customers to learn a predictor for each cluster aggregate, hence helping the design of the inference engine.

While CBAF methods empirically improve prediction accuracy, they cannot capture inter-cluster dependencies that, in some cases, could be relevant to achieve accurate forecasting. For instance, the energy consumption of industrial costumers is most likely important to predict the short term load requested by residential households. Standard CBAF requires to train a model from scratch for each cluster, limiting the scalability of the approach when considering expensive models such as deep neural networks. In fact, the availability of these large, heterogenous datasets makes deep learning (DL) models a suitable and appealing approach to address the problem of energy demand forecast [7].

In this work we study CBAF in the context of deep learning; in particular, we present a novel CBAF architecture suitable for deep learning models and introduce a neural network implementing it. Taking inspiration from the Multitask Learning literature (MTL) [8], we consider the cluster level aggregates as a single, multivariate, sequence and feed it into a single deep recurrent model with multiple output modules predicting the cluster level energy demand. We also present a practical method to cluster long time-series for CBAF. Finally, we evaluate our approach on a publicly available benchmark dataset, showing the benefits of the proposed approach and test our method in a new challenging dataset from the Swiss town of Arbon, providing empirical evidence that CBAF is a viable technique to improve prediction accuracy in STLF.

The paper is organized as follows. In Section II we for-

This project is carried out within the frame of the Swiss Centre for Competence in Energy Research on the Future Swiss Electrical Infrastructure (SCCER-FURIES) - Digitalisation programme with the financial support of the Swiss Innovation Agency (Innosuisse SCCER program).

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

malize the problem of STLF and cluster-based forecasting and, in Section III, we give a brief overview of the related works. In Section IV we present our approach and discuss advantages and drawbacks of various architectures for CBAF predictors. Finally, in Section V, we perform an extensive empirical evaluation of the proposed methods.

II. Problem description

In this Section, we define at first the general problem of STLF using data from SMs, then we discuss how clustering can be applied in the same settings to reduce the prediction error. Finally we introduce the performance metrics that will be used to asses proposed solutions.

A. Short-term load forecasting using Smart Meters data

We assume to have a dataset $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ of N univariate time-series where each $\boldsymbol{x}_i \in \mathbb{R}^T$ corresponds to a load profile associated with the i-th SM and $x_i[t]$ is a scalar measure of the energy drawn from the grid between discrete time $t-1$ and t, recorded at time t. We assume that all the time-series are synchronized, that is, a time step t corresponds to the same point in time across all the sequences. The objective is to build a predictor of the aggregated energy consumption, i.e., a predictor of the stochastic process generating the sequence $\boldsymbol{s} = (s[0], s[1], \ldots, s[T])$ where:

$$s[t] = \sum_{i=1}^{N} x_i[t] \tag{1}$$

Given a prediction horizon H (i.e., the number of steps ahead to predict) and a prediction window W (i.e., the number of previous steps used to make the prediction), this translates into minimizing a loss function, e.g., the prediction mean squared error (MSE):

$$MSE = \frac{1}{T-H-W} \sum_{t=W}^{T-H} \frac{1}{H} \sum_{h=1}^{H} \left(s[t+h] - \hat{y}_h\left(\boldsymbol{s}_t, \boldsymbol{u}_t; \boldsymbol{\theta}\right)\right)^2 \tag{2}$$

where $\boldsymbol{s}_t = (s[t-W], \ldots, s[t])$, $\boldsymbol{u}_t$ is a vector of exogenous variables, $\boldsymbol{\theta}$ represents the model's parameter vector and $\hat{y}_h$ is the h-step-ahead prediction.

In this paper, we focus on a day-ahead prediction horizon and on nonlinear autoregressive models, with optional exogenous variables, in particular deep recurrent neural networks ([9], [10]).

B. Clustering-based load forecasting

The idea behind cluster-based forecasting is to use relational information and similarities among data points to improve prediction accuracy [11]. In the aggregate load forecasting settings, this information is exploited by considering cluster-level aggregates that are expected to display more regular patterns, have higher auto-correlation and, thus, be more predictable signals than that accounting for the total energy consumption.

In practice, using CBAF, the dataset $\mathcal{D}$ is divided into K clusters C_k, and, by aggregation of the load profiles of each cluster, we obtain a second dataset $\mathcal{D}' = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_K\}$ such that:

$$X_k[t] = \sum_{\boldsymbol{x}_i \in C_k} x_i[t] \tag{3}$$

where $\boldsymbol{x}_i$ is a load profile belonging to cluster C_k.

At this point, a possible forecasting technique is to learn a model for each cluster-level aggregate and use the resulting ensemble of predictors to forecast the total load [6]. This approach, that we refer to as CBAF Ensemble (CBAF-E), is not well suited for deep learning analytics, due to the computational requirements of deep networks, in particular recurrent ones. This is a critical drawback considering that, due to the massive amount of data being generated, these models are expected to be often retrained. Furthermore, the combination of the clustering algorithm and the forecasting model already increases the number of hyperparameters to tune. On the other hand, the achievable accuracy gains make for a compelling argument in favor of the adoption of this technique when computational power is not an issue.

C. Metrics

To asses the performance of the designed predictors we use - as with STLF literature - the Mean Absolute Error (MAE) and the Root Means Square Error (RMSE) defined as (for clarity sake we omit the dependence on the prediction horizon and window):

$$RMSE = \sqrt{\frac{1}{T} \sum_{t=1}^{T} \left(y[t] - \hat{y}[t]\right)^2} \tag{4}$$

$$MAE = \frac{1}{T} \sum_{t=1}^{T} |y[t] - \hat{y}[t]| \tag{5}$$

Due to the MAE and RMSE being dependent on the scale of the observed values, we also report the normalized versions:

$$NRMSE = 100 \frac{RMSE}{y_{max} - y_{min}} \tag{6}$$

$$NMAE = 100 \frac{MAE}{y_{max} - y_{min}} \tag{7}$$

that is, we normalize the errors by the delta between the maximum and minimum observed values.

III. Related works

Load forecasting for power systems is an important research topic with clear and practical applications [12]. A recent survey [7] provides an empirical evaluation of deep learning models for STLF using modern best practices and showing convincing performance on multiple benchmark datasets. A particularly interesting line of research concerns the study of techniques to improve the performance of predictive models exploiting clustering algorithms [11].

Clustering of time-series is a challenging research problem [13], with a wide range of applications from customer segmentation [14] and stock-market analysis [15] to biology [16]. Cluster-based aggregate forecasting has been studied

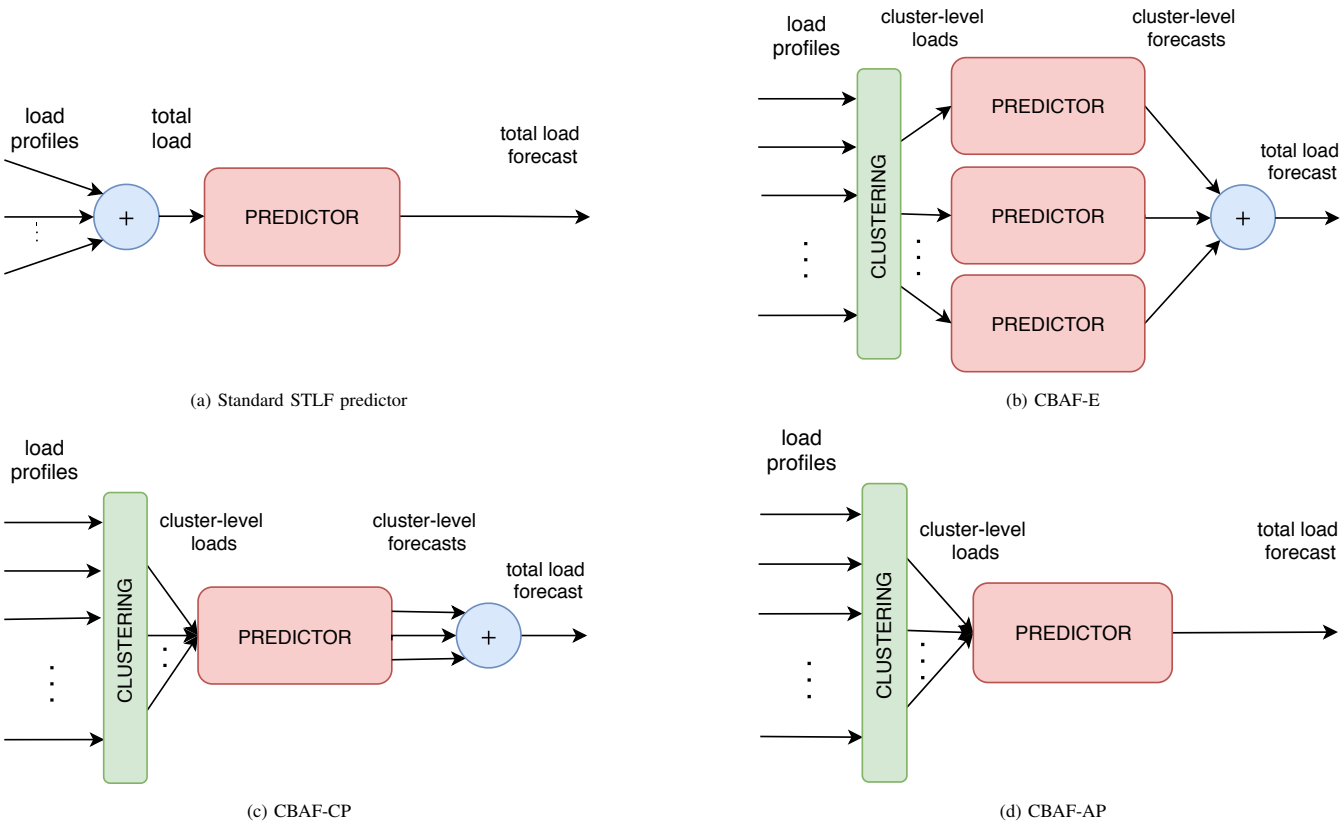

Fig. 1. **Forecasting architectures** (a) Standard forecasting architecture, based on complete aggregation of the load profiles. (b) **CBAF-E**: CBAF based architecture with a predictor for each cluster level aggregate. (c) **CBAF-CP**: CBAF architecture with a single model that jointly learns to predict each cluster level aggregate. (d) **CBAF-AP**: CBAF model that takes as an input the cluster-level aggregates and learns to directly predict the total aggregated load.

in depth in [6], where the authors provide an empirical analysis of different clustering algorithms and regression models on the Commission for Energy Regulation in Ireland (CER) dataset [17], a collection of SM data from residential and industrial customers. However, there, authors focus only on residential customers, using only shallow regression models, and report that the proposed methods do not show convincing performance gains over random cluster assignments. In [18], instead, CBAF is performed on a reduced version of the CER dataset with only industrial customers. Alzate and Sinn in [19], on the other hand, focus on the clustering algorithm and use an ensemble of simple Poisson autoregressive predictors. To the best of our knowledge, no prior work addresses CBAF as a MTL problem. The only example of CBAF with deep models that we are aware of is [20], where Fahiman et al. rely on rather simple networks and use engineered features rather than the raw time-series. Furthermore, they do not address the scalability issues of the approach and, consequently, limit their analysis to a maximum of 8 clusters in a reduced version of the CER dataset.

IV. Cluster-based aggregate load forecasting with deep recurrent neural networks

In this Section we first present our approach for CBAF, describing the general architecture of the proposed cluster-based predictors, then introduce a neural network model that implements those architectures. Finally, we discuss a practical procedure to cluster highly dimensional load profiles.

A. Cluster-based forecasting as a Multitask Learning problem

The standard STLF predictors, shown in Figure 1(a), do not exploit relational information to predict the load on the grid, while the standard CBAF-E approach, shown in Figure 1(b), scales poorly with the number of clusters and model complexity. To solve these problems, we propose CBAF in multi-task learning setting, studying predictors that require to train only a single model to forecast all the cluster-level loads at once. We focus on models that take as an input all the cluster-level aggregates as a single multivariate time-series. This choice is motivated by 1) improving scalability, 2) aiming at exploiting causality among the aggregates or, in other words, the fact that the load profile of one cluster might be useful to forecast, also, the behavior of the others and 3) establishing a framework and baseline for future research on cluster-based forecasting with deep neural networks.

We investigate two main forecasting architectures:

- **CBAF Cluster-level Predictors (CBAF-CP)**, Figure 1(c), learn to predict the short term energy demand of each cluster using a shared representation;

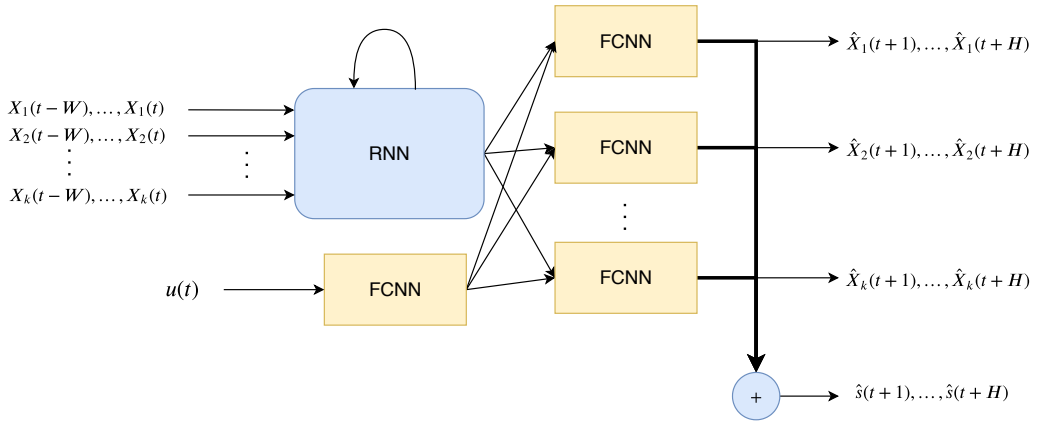

Fig. 2. Deep recurrent neural network with multiple regression heads. Each head shares the same representation of the input data and can be trained to predict the cluster-level aggregate load. The RNN block can be any recurrent multi-layer neural networks, while the FCNNs blocks are stacks of fully connected layers.

- **CBAF Aggregate Predictors (CBAF-AP)**, Figure 1(d), take the cluster-level aggregates as an input, but learn to directly predict the total consumption.

As already mentioned, the CBAF-CP architecture is inspired by the Multitask Learning literature [8], where the objective is to achieve better generalization sharing the representation of the input data across tasks. Following this approach, the features learned from the model to solve a specific task can be exploited as inductive biases to solve others. We include in our study the CBAF-AP architecture to asses if it would be possible to improve the prediction accuracy using clustering for feature extraction only. The empirical results supports the argument for a shared representation across cluster-level predictors.

B. Neural network architecture

We introduce a specific neural network architecture, shown in Figure 2, for CBAF with deep models (it follows that the predictor box of Figure 1 is that of Figure 2). We use a shared recurrent backbone, that can be implemented using any recurrent (multilayer) neural network, such as Long Short-Term Memory networks (LSTMs [9]). This module takes as input a multivariate time-series of length W representing the load profiles aggregated at a cluster level, the last hidden state - here acting as a feature vector - is then fed into K feedforward fully-connected network *heads* (FCNNs). Each head, one per cluster, can include multiple hidden layers and is trained to predict the aggregated energy consumption at each time-step of the prediction horizon H. In particular, we use the Multi input - Multi output (MIMO) prediction strategy [21], which consists in predicting the values to be forecasted for each time-lag all at once. Exogenous variables (i.e., date and/or weather features) can easily be included, as shown in the Figure 2, using a separated fully connected block, whose output can then be concatenated to the features extracted by the recurrent block. For simplicity, we consider only exogenous variables that are cluster independent, but it is straightforward to adapt the model to handle different cases.

Multi-head neural networks, often with a CNN backbone [22], are generally used to solve multiple regression or classification tasks that might benefit from the same representation of the input data. The use of shared layers greatly reduces complexity, both in time and space, and acts as a form of regularization, pushing the network to learn features that are useful across different problems, rather than overfitting to a single one. Furthermore, the MTL formulation of the problem allows us to conduct a more thorough empirical analysis of the advantages of using clustering techniques in the context of deep predictors.

This model represents an implementation fo the CBAF-CP architecture in Figure 1(c) and is jointly trained to predict each cluster-level aggregate. Empirically we found useful to scale the gradient flowing backward from each network head by a weight w_k such that:

$$w_k = \frac{\overline{X}_k}{\sum_{i=1}^{K} \overline{X}_i} \tag{8}$$

where $\overline{X}_k$ is the average load of cluster k. Note that this is different from optimizing the weighted sum of the prediction errors at the cluster level, since only the gradient of the shared parameters is scaled. This method allows the network heads to freely update their weights to minimize the task-specific error. At the same, the scaling avoids disruptive updates of the shared parameters due to bad predictions on smaller (in terms of load) clusters, which are typically characterized by harder to predict dynamics.

For $K = 1$, the model is reduced to a more standard deep STLF predictor that will be used in the following as a baseline. The CBAF-AP predictor is implemented, instead, simply by using a single-headed network with a K-dimensional input

sequence and trained to predict directly the total load. This last approach is conservative since clustering is used only for feature extraction and differences from the baseline predictor are minimal.

C. Clustering algorithm

The algorithm used for clustering needs to be tailored to reduce the prediction error across the cluster-level aggregates. The fundamental problem of clustering load profiles arises from the high dimensionality of the problem, both in terms of the length of the sequences, T, and the number of SMs, N, and the resulting necessity of building compact, but powerful, representations of the data points and their relationships.

In this work we do not analyze in depth the problem, but rather present a practical and efficient way to cluster long time-series based on their correlation, while we leave the extensive study of clustering algorithms for CBAF to future work. We obtain the cluster-level aggregates through the following steps.

1) We reduce the dimensionality of the problem considering, for each time-series, contiguous segments with a length corresponding to data collected over a week. Then, for each month of the year, we take for each time step the average across the aforementioned segments. At this point the signal is transformed in 12 subsequences of length $7 \cdot samples\_per\_day$.
2) For each pair of load profiles we compute a similarity measure as the average Person correlation coefficient between the subsequences obtained at the previous step[1].
3) We use the similarity matrix obtained at the previous step to build a K-Nearest neighbor graph of the dataset.
4) We perform spectral clustering [23] using the graph representation obtained at the previous step.

Building the similarity matrix can be computationally expensive if the number of load profiles is large, but the process can be efficiently parallelized and it needs to be executed only once since the hyperparameters (i.e., the number of neighbors and clusters) can be tuned independently afterwards. In practice, using the K-neighbors graph greatly simplifies the clustering problem.

V. Experiments

A. Experimental setup

In this section we carry out an extensive empirical evaluation of the discussed methods on two relevant real-world datasets. We compare the prediction accuracy of the baseline predictor of the aggregated energy demand against the CBAF-CP and CBAF-AP. For all experiments we use the neural network architecture described in Section IV. In particular, the RNN block is always implemented as a two layer LSTM, while each FCNN block is implemented with a single fully connected hidden-layer with $relu$ activations. We use the clustering algorithm presented in Section IV for the CBAF methods and we also include in the empirical analysis, a comparison against the case in which clusters are generated randomly partitioning the dataset. Each network is trained until convergence using early-stopping with a patience of 25 epochs and the Adam [24] optimizer with learning rate $1\mathrm{e}{-3}$. For each benchmarked method we run a grid-search to optimize the number of hidden units in each recurrent layer and in the network heads, the dropout rate, number of clusters and number of nearest neighbors (when applicable). The sampled values of the hyperparameters are shown as in Table I. The number of units in the fully connected block that extracts features from the exogenous variables is kept at $\frac{1}{4}$ of the neurons per layer in the recurrent block. The input data are Z-scaled. The best configuration for each method is chosen based on prediction accuracy on a validation set, while each reported result represents the average score on a test set across 10 independent runs (of both the clustering algorithm and the neural network training procedure). The objective of these experiments is to asses the validity of the proposed architecture as an alternative to a standard deep STLF predictor.

TABLE I
Grid search configuration. To save computational time, values indicated with a $*$ were not sampled when tuning CBAF methods.

hyperparameter	sampled values
n. units per recurrent layer	$[32^*, 64, 128, 256]$
n. units per head	$[128, 256]$
dropout in heads	$[0.0, 0.2^*]$
n. clusters	$[5, 10, 15]$
nearest neighbors	$[10, 20, 40]$

For both datasets, the load profiles are sampled with a period of 30 minutes and we keep the same prediction horizon and window, both of 24 hours (48 time-steps), across all the experiments. The only exogenous variables that we use are the one-hot encoding of the day of the week and the month of the year, to let the model learn the seasonalities in the data. Better prediction accuracy might be achieved by adding temperature and weather information, but we prefer to keep the model and the analysis simple, as the focus here is on the effectiveness of the clustering approach.

B. CER dataset

The CER dataset [17], is a collection of 6k load profiles of residential and industrial customers from an Irish city, recorded over a time-span of approximately one year and a half between 2009 and 2010. The objective of the experiment was to observe the impact of time-of-use tariffs, introduced to reduce peak load, among different customers. Data were acquired with a sampling rate of 30 minutes and have been preprocessed before being open-sourced. We impute the (few) missing values for each time-series as the average consumption of the specific customer at each time of the day across the same weekday and month. We use as training set all the data except the last 4 months, 1 of which is used for validation and 3 for testing. Figure 3 shows a sample of the aggregated energy consumption, while Figure 4 shows cluster-level aggregates

[1]In practice, we found concatenating adjacent subsequences two by two, before computing the average correlation, to work better if the data are noisy.

in the same week. From a qualitative analysis with only 4 clusters it is already possible to see how the different groups exhibit different behaviors across different periods of the day and different days of the week, in particular at weekends.

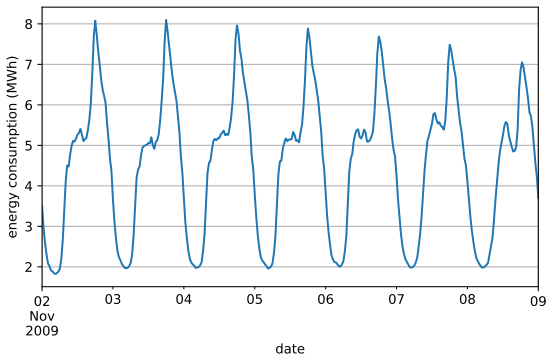

Fig. 3. Sample of CER total energy demand across one week, in November 2009.

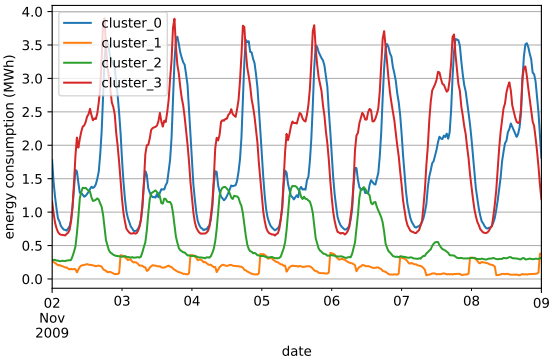

Fig. 4. Sample of CER cluster level aggregates. The number of clusters used here has been selected for the purposes of illustration and is different from the one used in final models. Best viewed in color.

C. Arbon dataset

We evaluate the proposed methods on a dataset of SM data from the town of Arbon in northeastern Switzerland. The anonymized data have been acquired, in the context of the SCCER-FURIES project framework [25], from SMs installed in the city and cover the time-span between September 2017 and September 2019. The dataset consists in around 10k load profiles from the different SMs installed in the grid at different sites. For the purpose of this study we filter out almost 25% of the available load profiles, due to outliers and a high number of missing or null readings. The other 75% of the data are preprocessed, filling missing values using seasonal averages, and resampled from the original 15 minutes rate to a 30 minutes period. We refer to [26] for more details about the data acquisition process. Differently from the CER dataset here the load profiles report the average active power for each time-step. We use 2 months for validation and 4 for testing.

A sample of the total aggregated energy demand and cluster-level aggregates are show in Figure 5 and 6 respectively. The difference between the two datasets here is clear: the Arbon dataset is characterized by higher frequency components and more heterogeneous signals. It is also more apparent how predicting cluster-level aggregates may result in an easier learning task since they appear to be smoother and more autocorrelated. Another interesting aspects is that the clustering approach highlights the presence of load profiles with extremely regular consumption patterns that are probably due to automated processes.

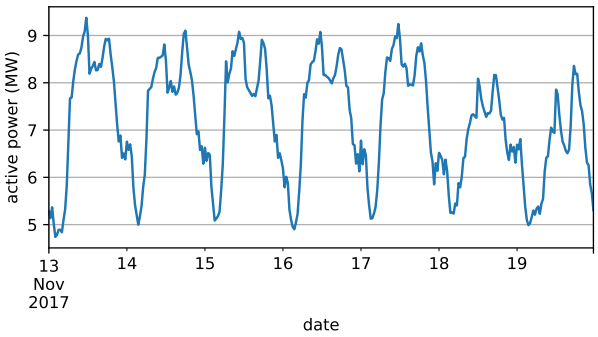

Fig. 5. Sample of the Arbon total energy demand across one week, in November 2017.

Fig. 6. Sample of Arbon cluster level aggregates. The number of clusters used here has been selected for the purposes of illustration and is different from the one used in final models. Best viewed in color.

D. Results

Results of the empirical evaluation are shown in Table II for the CER dataset and in Table III for the Arbon dataset. For both datasets, using CBAF with spectral clustering results in a substantial improvement in prediction accuracy over a naïve aggregation strategy. CBAF, as we could expect, appears to be more beneficial for the more complex and noisier dataset, where the SM data are more heterogeneous. In particular CBAF-CP achieves a **23%** improvement in MAE over the standard STLF predictor on the Arbon dataset. However the improvement in prediction accuracy is remarkable also in the CER - simpler - dataset case, confirming the results of the analyses conducted in previous works.

TABLE II
RESULTS ON THE **CER** DATASET. RANDOM INDICATES CLUSTERING BASED ON RANDOM CLUSTER ASSIGNMENTS, WHILE SPECTRAL THE CLUSTERING PROCEDURE INDICATED IN SECTION IV

		MAE	NMAE	RMSE	NRMSE
NO-CBAF		0.189 ± 0.006	2.551 ± 0.076	0.278 ± 0.011	3.754 ± 0.146
CBAF-AP	RANDOM	0.189 ± 0.008	2.551 ± 0.108	0.272 ± 0.012	3.680 ± 0.169
	SPECTRAL	0.173 ± 0.003	2.338 ± 0.044	0.246 ± 0.006	3.332 ± 0.082
CBAF-CP	RANDOM	0.188 ± 0.007	2.544 ± 0.098	0.274 ± 0.012	3.701 ± 0.160
	SPECTRAL	$\mathbf{0.167 \pm 0.005}$	$\mathbf{2.253 \pm 0.068}$	$\mathbf{0.234 \pm 0.007}$	$\mathbf{3.160 \pm 0.093}$

TABLE III
RESULTS ON THE **ARBON** DATASET. RESULTS ON THE CER DATASET. RANDOM INDICATES CLUSTERING BASED ON RANDOM CLUSTER ASSIGNMENTS, WHILE SPECTRAL THE CLUSTERING PROCEDURE INDICATED IN SECTION IV

		MAE	NMAE	RMSE	NRMSE
NO-CBAF		0.304 ± 0.007	5.127 ± 0.121	0.437 ± 0.012	7.372 ± 0.197
CBAF-AP	RANDOM	0.295 ± 0.014	4.967 ± 0.242	0.413 ± 0.023	6.960 ± 0.381
	SPECTRAL	0.260 ± 0.009	4.378 ± 0.159	0.377 ± 0.016	6.355 ± 0.262
CBAF-CP	RANDOM	0.321 ± 0.036	5.412 ± 0.616	0.450 ± 0.059	7.585 ± 0.981
	SPECTRAL	$\mathbf{0.232 \pm 0.008}$	$\mathbf{3.916 \pm 0.141}$	$\mathbf{0.344 \pm 0.135}$	$\mathbf{5.798 \pm 0.227}$

TABLE IV
RESULTS ON THE ARBON DATASET. ΔT INDICATES THE DIFFERENCE IN COMPUTATIONAL TIME, ΔM THE DIFFERENCE IN NUMBER OF PARAMETERS.

	n_clusters	MAE	ΔT	ΔM
CBAF-CP	15	0.232 ± 0.008	–	–
CBAF-E	7	0.242 ± 0.007	$\sim 2.5\times$	$\sim 1.2\times$
	15	$\mathbf{0.209 \pm 0.003}$	$\sim 5.3\times$	$\sim 2.7\times$

The performance gain achievable solely with the simple CBAF-AP approach, which has almost no impact in time and space complexity, represents yet another interesting result. Furthermore, the superiority of the principled approach (based on spectral clustering) over the random baseline (where load profiles are randomly assigned to clusters) suggests that the clustering algorithm actually plays an important role in the improvement of prediction accuracy.

E. CBAF-E

As already mentioned, training a single model for each cluster level aggregate is expensive in terms of both memory and time. In particular, tuning the models hyperparameters through an extensive grid search is almost unfeasible. For a fair comparison we compare the CBAF-E approach against the MTL architecture tuning down the number of clusters in order to keep the number of parameters similar, while keeping the other hyperparameters unchanged. Finally, we evaluate the level of accuracy achievable when trading-off space and time complexity using an ensemble of predictors for an high number of clusters. Table IV shows the results of these comparisons on the Arbon dataset. In this case, fitting a predictor for each cluster yields better results but, as already discussed, it is limited in terms of scalability.

VI. CONCLUSIONS AND FUTURE WORKS

In this work we presented a MTL approach for electrical load forecasting based on a clustering algorithm that groups load profiles based on their correlation. Furthermore, we studied CBAF on a heterogeneous dataset of raw load profiles. The proposed method scales nicely with the number of clusters and the empirical results strongly advocate for the adoption of CBAF to tackle the STLF problem. In particular, the clear performance gains over the random clustering baselines, differently from previous works, confirm that clustering can effectively be used to improve prediction accuracy. However, matching the accuracy achievable using multiple predictors is still a challenge and requires further effort.

The neural network adopted here is versatile and future work should focus on more specialized neural architectures to better exploit the available relational information. In particular, the backbone of the presented model could easily be replaced to handle a graph representation of the signals. Another interesting approach could be to study an hybrid architecture between the ensemble approach and the MTL models, where, for instance, tasks are grouped and assigned to models using some similarity measure. Future work should also address the non-stationarity of the problem, studying inter-cluster dynamics and proper techniques to handle new SMs data being installed in the grid.

REFERENCES

[1] X. Fang, S. Misra, G. Xue, and D. Yang. Smart grid — the new and improved power grid: A survey. *IEEE Communications Surveys Tutorials*, 14(4):944–980, Fourth 2012.

[2] S. Massoud Amin and B. F. Wollenberg. Toward a smart grid: power delivery for the 21st century. *IEEE Power and Energy Magazine*, 3(5):34–41, Sep. 2005.

[3] Eisa Almeshaiei and Hassan Soltan. A methodology for electric power load forecasting. *Alexandria Engineering Journal*, 50(2):137 – 144, 2011.

[4] L. Song, Y. Xiao, and M. van der Schaar. Demand side management in smart grids using a repeated game framework. *IEEE Journal on Selected Areas in Communications*, 32(7):1412–1424, July 2014.

[5] Seyedeh Fallah, Ravinesh Deo, Mohammad Shojafar, Mauro Conti, and Shahaboddin Shamshirband. Computational intelligence approaches for energy load forecasting in smart energy management grids: State of the art, future challenges, and research directions. *Energies*, 11:596, 03 2018.

[6] T. K. Wijaya, M. Vasirani, S. Humeau, and K. Aberer. Cluster-based aggregate forecasting for residential electricity demand using smart meter data. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 879–887, Oct 2015.

[7] Alberto Gasparin, Slobodan Lukovic, and Cesare Alippi. Deep learning for time series forecasting: The electric load case. *CoRR*, abs/1907.09207, 2019.

[8] Richard Caruana. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 41–48. Morgan Kaufmann, 1993.

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[10] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[11] A. Sfetsos and C. Siriopoulos. Time series forecasting with a hybrid clustering scheme and pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 34(3):399–405, May 2004.

[12] Hesham K. Alfares and Mohammad Nazeeruddin. Electric load forecasting: Literature survey and classification of methods. *Int. J. Systems Science*, 33:23–34, 2002.

[13] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. Time-series clustering – a decade review. *Information Systems*, 53:16 – 38, 2015.

[14] Omid Motlagh, Adam Berry, and Lachlan O'Neil. Clustering of residential electricity customers using load time series. *Applied Energy*, 237:11 – 24, 2019.

[15] Tak-Chung Fu, Fu-Lai Chung, Vincent Ng, and Robert Luk. Pattern discovery from stock time series using self-organizing maps. *Workshop Notes of KDD2001 Workshop on Temporal Data Mining*, 01 2001.

[16] Numanul Subhani, Luis Rueda, Alioune Ngom, and Conrad J. Burden. Multiple gene expression profile alignment for microarray time-series data clustering. *Bioinformatics*, 26(18):2281–2288, 07 2010.

[17] Paula Carroll, Tadhg Murphy, Michale Hanley, Daniel Dempsey, and John Dunne. Household classification using smart meter data. *Journal of official statistics*, 34, 01 2018.

[18] M. Misiti, Y. Misiti, G. Oppenheim, and J. M. Poggi. Optimized clusters for disaggregated electricity load forecasting. *REVSTAT*, 8:105–124, 2010.

[19] C. Alzate and M. Sinn. Improved electricity load forecasting via kernel spectral clustering of smart meters. In *2013 IEEE 13th International Conference on Data Mining*, pages 943–948, Dec 2013.

[20] F. Fahiman, S. M. Erfani, S. Rajasegarar, M. Palaniswami, and C. Leckie. Improving load forecasting based on deep learning and k-shape clustering. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 4134–4141, May 2017.

[21] Gianluca Bontempi. Long term time series prediction with multi-input multi-output local learning. *Proceedings of the 2nd European Symposium on Time Series Prediction (TSP), ESTSP08*, 01 2008.

[22] Ian J. Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay D. Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks. *CoRR*, abs/1312.6082, 2013.

[23] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, page 849–856, Cambridge, MA, USA, 2001. MIT Press.

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[25] SCCER-FURIES – FUTURE SWISS ELECTRICAL INFRASTRUCTURE, (accessed 2020-01-05). https://www.epfl.ch/research/domains/sccer-furies/.

[26] I. Herbst, S. Lukovic, A. Gasparin, N. Schulz, J. Witzig, and S. Kieber. LV grid data analysis demonstrated at DSO Arbon Energie. In *25th International Conference on Electricity Distribution (CIRED)*, 2019.