One model fits all: Individualized household energy demand forecasting with a single deep learning model

Spiros Chadoulos, Iordanis Koutsopoulos, and George C. Polyzos
Mobile Multimedia Laboratory, Department of Informatics
School of Information Sciences and Technology
Athens University of Economics and Business
Greece
{spiroscha,jordan,polyzos}@aueb.gr

ABSTRACT

Energy demand forecasting at the household level is an important issue in smart energy grids to facilitate applications such as residential Demand Response (DR). However, if a separate machine learning model is trained for each house, the erratic nature of some consumers will lead to significant inaccuracies for the respective models, while predictions for new households with scarce data will not be possible to generate. In this work, we propose an approach with a single deep learning model that is trained on multiple households, which can create hourly energy consumption forecasts for individual households. We present a novel architecture that combines a Recurrent Neural Network (RNN) encoder and a Multilayer Perceptron (MLP). Our approach captures both the impact of past consumption time-series and that of energy profiles on future energy demand. Our model incorporates energy profiles to derive different characteristics between consumers, and it features a "double" clustering procedure that is specially designed for a mixture of time-series and non-time-series data. Experiments with real smart meter data show that the proposed neural network architecture achieves high performance in predicting energy consumption both for known and new consumers not present in the training dataset, with a Mean Absolute Percentage Error (MAPE) of 10.1% and 12.5% respectively.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Hardware** → **Smart grid**.

KEYWORDS

Smart grids, Energy consumption forecasting, Deep learning

ACM Reference Format:
Spiros Chadoulos, Iordanis Koutsopoulos, and George C. Polyzos. 2021. One model fits all: Individualized household energy demand forecasting with a single deep learning model . In *The Twelfth ACM International Conference on Future Energy Systems (e-Energy '21), June 28–July 2, 2021, Virtual Event, Italy.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3447555.3466587

e-Energy '21, June 28–July 2, 2021, Virtual Event, Italy

ACM ISBN 978-1-4503-8333-2/21/06…$15.00
https://doi.org/10.1145/3447555.3466587

1 INTRODUCTION

Energy demand forecasting is a crucial component of many smart grid applications in order to ensure grid stability by balancing energy supply and demand. Short-Term Load Forecasting (STLF) refers to energy consumption predictions with a granularity ranging from a few minutes up to several hours or even days, while Long-Term Load Forecasting (LTLF) refers to granularities of over two weeks [13]. STLF is utilized by applications and services such as Demand Response (DR), hour-ahead scheduling, and day-ahead scheduling [13]. DR programs are designed to change the consumption profile of energy consumers in order to avoid demand-supply imbalances, through incentive mechanisms such as dynamic pricing or through direct automated control of appliances [6]. STLF is also crucial for energy markets since market players need to provide accurate supply/demand bids for the day-ahead and real-time markets [8].

Consequently, energy demand needs to be predicted at the individual consumer level since accurate forecasts can improve energy market and DR mechanisms through better targeting consumers for DR events based on their predicted consumption. Other applications of household-level demand forecasting include consumer targeting for energy efficiency campaigns and smart meter fault detection.

A common approach in the literature is to train an individual consumption forecasting model for each house with historical data collected from smart meters [22]. However, *a machine learning model trained on data from a single household cannot generalize well enough and cannot make good enough predictions for new consumers that are not included in the training dataset* [22]. On the other hand, it is not possible to train a new model for each new household if historical consumption data are not available for it, which is the so called cold-start problem.

Furthermore, energy consumption forecasting at the household level is by definition a difficult task due to the involved energy demand uncertainty, which mostly depends on the behaviour of consumers. In other words, despite the fact that the single-model-per-household approach has achieved accurate demand forecasting in the literature, there are cases where in the presence of erratic consumers with completely unpredictable consumption patterns, even the most powerful deep learning models fail to perform accurately. In such cases, a single deep learning model trained on a large number of households can effectively discover common patterns between consumers and provide accurate forecasts even for households with erratic energy demand patterns.

Moreover, an electricity retailer might want to acquire individual predictions for a large number of households, e.g. hundreds or even

thousands of residential customers in order to increase the impact of a DR program. However, the approach of an individual model for each consumer might be computationally demanding since demand variability and uncertainty require the use of resource-consuming deep learning models, which often need individual hyper-parameter tuning to perform accurately on each house [22].

For the aforementioned reasons, in this work *we design and validate an approach with a single deep learning model architecture trained on consumption data from multiple houses.* The goal of the proposed approach is to distinguish the energy demand patterns between different households and conduct predictions for any electricity consumer, by constructing an energy profile for each household. Energy profiles have been used in the literature to train clustering and classification models [3], [14], [15], [12], but they are not used as as a direct input for a deep learning model that predicts energy consumption for individual houses. Specifically, such a model should achieve *"generalization"*, meaning that it will be able to accurately predict the future consumption of houses on which it is trained, while it should also achieve *"representativeness"* in the sense that it will be capable of predicting the future energy demand of new (unseen) houses with a few days' data.

In our architecture, we utilize a "double" clustering procedure where the time-series features of each profile (e.g. daily load profiles) are used for consumer clustering with a version of k-means for time-series data and the rest of the profile features are used with classical k-means. The derived distances from each cluster centroid for each household act as encodings of the energy profiles and are used as an additional input for the deep learning consumption forecasting model. Our architecture features a *novel RNN encoder and MLP neural network architecture that captures the effect that both past consumption and energy profiles have on future energy demand*, thus achieving greater prediction performance than the state-of-the-art approaches in hourly household demand forecasting for individual households with a single model. Furthermore, our neural network architecture along with the derived combination of input features leads to a model capable of making individualized predictions even for unseen houses. To summarize, the contributions of this work are the following:

- We introduce a novel neural network architecture with a combination of an RNN encoder and an MLP for individual household energy consumption forecasting.
- We propose the utilization of energy profiles as additional input features to the deep learning architecture, enabling it to learn the differences between individual consumers' energy consumption patterns.
- We extend the model's ability to distinguish different energy demand patterns among individual households by proposing a "double" clustering approach on the energy profiles.
- We validate the proposed approach with energy consumption data from real households, with the experiments showing that the model achieves both "generalization" (by making accurate hourly forecasts on test data with a *MAPE* of 10.1% for houses included in the training set), and "representativeness" (by conducting accurate predictions for completely new households, with a *MAPE* of 12.47%).

The rest of the paper is organized as follows: in section 2, prior related work is discussed and the novelty of our work is highlighted. In section 3, a brief background presentation is conducted. In section 4, the proposed approach is explained in detail, while in section 5, the experimental setup and results are presented and discussed. Finally, in section 6 we conclude and summarize this work's contributions and highlight future directions.

2 RELATED WORK

A commonly used approach for predicting the future energy consumption of a household or a building is to train a machine learning model with historical smart meter data specifically from this household. Various machine learning algorithms have been studied for this purpose [1], [2], [10], [19], [20], [21], [26], [28], showing promising results regarding the prediction performance for a single house/building, with neural networks generally achieving the highest accuracy. However, as discussed earlier, a model trained to conduct forecasts for a single household/building cannot provide predictions for new houses. If these houses do not have sufficient historical data to train new models for them, the cold-start problem will emerge.

Another approach to tackle the problem of household energy demand forecasting is to train a single machine learning model with historical consumption data from multiple households. The literature is significantly sparser on this approach. In [22], the authors developed a general purpose time-series probabilistic forecasting model named *DeepAR* using Autoregressive Recurrent Networks, which is capable of predicting a future portion of a time-series with a single encoder-decoder neural network. The proposed model assumes a probability distribution for the data and learns its mean and standard deviation for each time slot, while also using a set of time-dependent covariates as an input. In their evaluation, the authors conducted experiments with datasets from many domains, including household energy consumption forecasting, where the model achieved a Root Mean Squared Error (RMSE) of 1.0 on predicting the hourly energy demand for the next 24 hours.

Our approach differs from that in [22] since we propose a simpler deep learning architecture with an RNN encoder and an MLP that is specially designed for the task of deterministic individualized household energy demand forecasting by integrating energy profiles and consumer clustering in the model's inputs. Contrary to [22], our model focuses on predicting the energy consumption for a single time slot without having to assume a probability distribution for the data, instead of forecasting the energy consumption's probability distributions for a time window. This leads to a lighter architecture, which however seems capable of achieving better performance in terms of single prediction point accuracy (since we report lower RMSEs). Furthermore, even though determining the probability distribution forecast for an individual consumer would be a plus, this does not seem to be a current requirement in the energy sector for DR programs.

In [25] the authors use Dynamic Time Warping (DTW) to cluster 24-hour load curves instead of houses, originating from a dataset of approximately 1000 households with a time horizon of 22 days, resulting to 20 clusters. Then, each load curve is encoded with the nearest cluster centroid, in terms of DTW distance, and Markov

models are trained to conduct next day load curve forecasting. This approach achieved lower DTW-error compared to prior works, while the prediction was extended to appliance-level consumption.

In [23], a pooling-based deep RNN is proposed for household demand forecasting. The authors use the term "pooling" to introduce a set of input features that consists of a batch of randomly selected load profiles from other "neighbouring" households. The proposed pooling strategy achieves an improvement of 6.96% (compared to not using the model) in terms of prediction performance while avoiding overfitting. In [18] a transfer learning approach is introduced to tackle energy demand forecasting for multiple houses. More specifically, k-means is used to cluster apartments based on their daily load profiles, while a separate "base" RNN is trained on each cluster centroid's profile. Each trained RNN is utilized as a base model to train an individual neural network for each apartment through Transfer Learning, which clearly goes beyond the setting of one model trained on multiple houses. Experiments with 15-min consumption data from two buildings consisting of 96 and 91 apartments respectively show that the proposed approach achieves greater performance in terms of computational time and forecasting error compared to training a separate RNN model for each apartment, with a *MAPE* of 34.3% and 41.07% for the two buildings respectively.

Our work differs from [25], [23], and [18] since we use consumer energy profiles along with distances from cluster centroids as additional inputs for the demand forecasting model to learn how to distinguish consumption patterns and characteristics between different consumers. Furthermore, we propose a novel neural network architecture for energy demand prediction that captures the effect of both past consumption with an RNN encoder component and energy profiles with an MLP component. This leads to a MAPE of 10.1% for already seen houses and 12.47% for new/unseen households, meaning that the proposed approach achieves both "generalization" and "representativeness".

3 BACKGROUND

3.1 Deep learning

A neural network consists of neurons organized in layers, with each neuron having a feature vector $\boldsymbol{x}$ as an input from the previous layer and outputting a value calculated as:

$$h_n = f(\boldsymbol{w_n}^T \boldsymbol{x} + b_n), \tag{1}$$

where h_n refers to the n-th neuron of this specific layer, $\boldsymbol{w_n}$ is the weight vector, b_n is the bias term, and $f(\cdot)$ is the activation function. The model learns the weight and bias parameters of each neuron using backpropagation [27] on the training data, which are iterated several times, while each full iteration is defined as an epoch. During each epoch, the training set is split into mini-batches and the error gradient $\nabla E(\boldsymbol{w})$ is computed for each mini-batch according to a loss function $E(\cdot)$, the network's output, and the real labels. The error gradient is distributed to all neurons utilizing the chain rule and the weights are updated with an optimizer such as Stochastic Gradient Descent (SGD), which uses the following rule: $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \nabla E(\boldsymbol{w})$, where η refers to the learning rate selected.

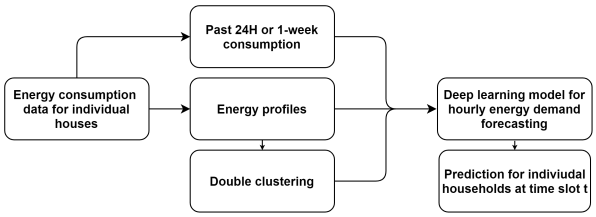

Figure 1: Proposed approach flow.

A neural network variant which also keeps an internal memory that captures the temporal characteristics of input feature sequences, is called a Recurrent Neural Network (RNN). RNNs are utilized in various applications that use time-series data, such as energy consumption measurements. The most popular RNN cell versions are Long Short-Term Memory neural networks (LSTMs) and Gated Recurrent Units (GRUs). The two variants have shown similar performance in many problem settings, and GRUs are more efficient in terms of computations since they use fewer parameters [7].

3.2 Clustering

A clustering algorithm groups a set of objects based on their features, in order to have similar objects on the same cluster and dissimilar ones to different clusters, based on a predefined distance measure. The most popular clustering algorithm is k-means with Euclidean distance (2), which iteratively places each data point into the closest cluster, while the number of clusters k is predefined. The Euclidean distance for two feature vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ is defined as:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \tag{2}$$

A variant of k-means that is utilized to cluster time-series data based on their curve shape, uses Dynamic Time Warping (DTW) distance [4], [25] as a similarity measure, instead of Euclidean distance. A warping path on a $n \times n$ matrix needs to be defined as a sequence $p = (p_1, \ldots, p_L)$ where $p_l = (i_l, j_l)$ and $l \in [1 : L]$, with $p_1 = (1, 1)$, $p_L = (n, n)$, $i_1 \leq i_2 \leq \ldots \leq i_L$, and $j_1 \leq j_2 \leq \ldots \leq j_L$. Furthermore, the cost of a warping path p for two feature vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined as:

$$c_p(\boldsymbol{x}, \boldsymbol{y}) = \sum_{l=1}^{L} (x_{i_l} - y_{j_l})^2, \tag{3}$$

and the DTW distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined as:

$$DTW(\boldsymbol{x}, \boldsymbol{y}) = c_{p^*}(\boldsymbol{x}, \boldsymbol{y}), \tag{4}$$

where $p^* = \arg\min c_p(\boldsymbol{x}, \boldsymbol{y})$ is the warping path with the lowest possible cost, which is found using dynamic programming [4], [25].

4 PROPOSED APPROACH

The approach that is proposed in this work is depicted in Fig. 1. It takes advantage of household energy profiles to learn the differences between the consumption patterns and characteristics of individual consumers. Then, a "double" clustering procedure is conducted to group households with similar energy profiles, leading

Table 1: Energy profile features used in our approach.

IDs	Profile feature	Description	Type
1-24	24h load profile	Hourly average normalized consumption (24 features)	Time-series
25	mean	Mean consumption	Non-time-series
26	var	Consumption variance	Non-time-series
27	max	Maximum consumption	Non-time-series
28	min	Minimum consumption	Non-time-series
29	min_over_mean	min / mean	Non-time-series
30	mean_over_max	mean / max	Non-time-series
31	PR_1 (Relative average consumption 1)	$\bar{P}_1$ / mean	Non-time-series
32	PR_2 (Relative average consumption 2)	$\bar{P}_2$ / mean	Non-time-series
33	PR_3 (Relative average consumption 3)	$\bar{P}_3$ / mean	Non-time-series
34	PR_4 (Relative average consumption 4)	$\bar{P}_4$ / mean	Non-time-series
35	weekend_weekday_difference_score	$\frac{1}{4}\sum_{j=1}^{4}\frac{\lvert PWD_j - PWE_j\rvert}{\bar{P}_j}$	Non-time-series
36	mean_relative_std	$\frac{1}{4}\sum_{j=1}^{4}\frac{\sigma_j}{\bar{P}_j}$	Non-time-series
37	seasonal_score	$\frac{1}{4}\sum_{j=1}^{4}\frac{\lvert PW_j - PS_j\rvert}{\bar{P}_j}$	Non-time-series

to an encoding for each energy profile based on its distance from each cluster's centroid. Both the energy profiles and the clustering distances are used as additional input features for the neural network that predicts the hourly energy demand for individual households. Moreover, a novel RNN-encoder-based deep neural network architecture is proposed to further increase the predictive performance of the model, in terms of both "generalization" and "representativeness".

4.1 Household energy profiles

An energy profile is essentially a vector of features consisting of multiple characteristics and statistics calculated from the available consumption data of a household, which can be categorized as either time-series or non-time-series features. The time-series features of an energy profile can include ordered statistics for specific time periods, such as average hourly energy consumption (24 features). The rest of the profile features are described as non-time-series features and can include any other statistic derived from the consumer's consumption data. In this work, the energy profiles' non-time-series features include a set of statistics proposed by [3], [14], and [15]. Specifically, these features consist of: consumption figures (i.e. aggregates of consumption during different periods of the day), consumption ratios (features calculated as the ratio of two consumption figures), and statistical features (e.g. mean, variance, etc.).

The energy profiles we use also include a set of features proposed by [12], namely relative average power in each period of the day, mean relative standard deviation, seasonal score (to capture the seasonality observed in the data), and weekend vs weekday score. The periods in which each day is divided are: overnight (period 1, 22:00-6:00), breakfast (period 2, 6:00-9:00), daytime (period 3, 09:00-15:00), and evening (period 4, 15:00-22:00). Similarly to the notation of [12], for each consumer, we define $\bar{P}_j$ as the mean power consumption and σ_j as the standard deviation for each time period j ($j = 1, 2, 3, 4$). Furthermore, the mean power consumption during summer and winter for each time period j is defined as PS_j and PW_j respectively, while the mean power consumption during weekends and weekdays for each time period j is defined as PWE_j and PWD_j respectively. The complete energy profile we consider for each household in this work is presented in Table 1.

4.2 Household "double" clustering

Clustering of residential energy consumers is a common approach in the literature to assign consumers with similar consumption characteristics into groups. Most approaches use classic clustering algorithms such as k-means with Euclidean distance, which is not always the most appropriate approach when the clustering inputs include time-series features.

Specifically, classical k-means with Euclidean distance is not invariant to minor time shifts since it measures point-to-point distance, meaning that two 24-hour load curves with similar shape and consumption levels will have a large Euclidean distance if one of them is shifted by just one hour [25]. Hence, we utilize a variant of k-means that uses Dynamic Time Warping (DTW) as a distance measure to cluster households based only on the time-series features of the energy profiles, while the rest of the profile features are utilized as an input for k-means with Euclidean distance. Thus, the clustering procedure of the households is split into two separate clustering flows based on their energy profiles, with each house having two cluster memberships for the two clustering algorithms respectively. All the distances from the cluster centroids can serve as an additional input for other machine learning models, while in this work we utilize them as additional input features for the demand forecasting neural network. In section 5 we present experiments conducted with real data that show both the improvement with the "double" clustering approach in terms of cluster quality, as well as the predictive performance improvement of the demand forecasting model when the cluster centroid distances are included in the input feature vector.

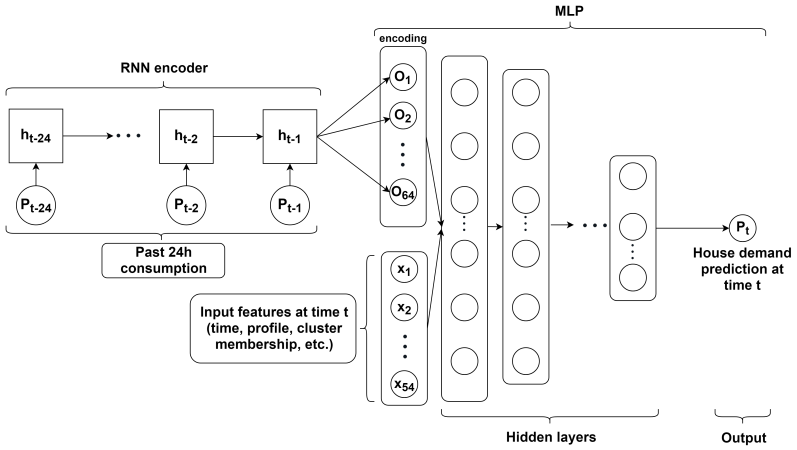

Figure 2: The proposed RNN-encoder-and-MLP architecture.

4.3 A single consumption forecasting model for any household

The goal is to train a model that generalizes well on test data from observed houses (i.e. houses included in the training data), while also achieving "representativeness" by having a high prediction performance even for completely new/unseen houses (i.e. houses for which the training set did not include any data). As depicted in Fig. 1, using the constructed energy profiles described earlier, as well as the distances from all cluster centroids calculated for each household, an input feature vector is assembled to train a machine learning model that outputs the household's demand P_t for time slot t (in this work we use 1-hour time slots to tackle STLF). The input feature vector also includes time-related features, i.e. Hour (0-23), Weekday (0-6), DayOfYear (1-365), and Month (1-12), while it can also incorporate past consumption measurements (e.g. for the past 24 hours: $P_{t-24}, \ldots, P_{t-1}$) and other available metadata for each household depending on the dataset (e.g. house size in m^2, solar panel integration, etc.).

The rationale is that features such as energy profiles, clustering distances, and past consumption help the model to learn the differences between individual households in terms of demand patterns and characteristics, thus distinguishing among certain house attributes through the hidden layers, and providing accurate forecasts. In other words, such a model is capable of providing hourly consumption forecasts for any house, with just a few days' data to construct the energy profile. In section 5, different combinations of the aforementioned input features are tested with different models and their impact on the results is assessed.

4.4 The proposed RNN-encoder-and-MLP deep neural network architecture

As mentioned in the previous section, past consumption (e.g. past 24 hour or even past week) can increase the model's performance when it predicts the household demand P_t for time slot t. However, if the past consumption measurements are directly used as a part of the input feature vector for a neural network (e.g. an MLP [11]), the time dimension of the past consumption sequence will be ignored. In particular, the model will treat the past consumption features as independent and not as a time-series sequence of measurements. On the other hand, if an RNN is trained on just the time-series consumption measurements, the rest of the features discussed earlier, such as energy profiles, will not be included.

We propose a novel model architecture depicted in Fig. 2, that consists of an RNN encoder with the past consumption features as input, and an MLP with the rest of the features. The input to the MLP also includes the encoding vector that is the result of the RNN encoder. The rationale is that this combines the advantages both of an RNN trained on energy demand time-series data and an MLP trained on features such as energy profiles and cluster distances for each household. In other words, when predicting a household's energy consumption for time slot t, the model also incorporates an encoding that represents all the information it assumes is important for the past 24 hours. Specifically, the input feature vector for a specific household at time slot t with a 24-hour look-back window is:

$$\boldsymbol{I} = (P_{t-24}, \ldots, P_{t-1}, x_1, \ldots, x_{54}), \tag{5}$$

where $(P_{t-24}, \ldots, P_{t-1})$ is the past 24h energy consumption for the specific household, and $(x_1, \ldots, x_{54})$ includes the 37 energy profile features presented in Table 1 along with the 17 following features:

- Hour (0-23), Weekday (0-6), Month (1-12), DayOfYear (1-365);
- Distances from non-time-series cluster centroids (k^* distances for k^* clusters, with 5 clusters in our case);
- Distances from time-series cluster centroids (k distances for k clusters, with 6 clusters in our case);
- pv (Boolean feature for solar panel existence);
- total_square_footage (house area in m^2).

Various combinations of these features can lead to different results, as we show in section 5. The past consumption part of $\boldsymbol{I}$, i.e. $(P_{t-24}, \ldots, P_{t-1})$, is used as an input for an RNN encoder with Gated Recurrent Unit (GRU) cells [7]. We use a GRU RNN as an encoder since it is computationally more efficient than LSTMs while preserving similar performance. Furthermore, the utilization of more complex RNN architectures, such as stacked RNNs, did not

show any performance gains in our experiments. An example of the last GRU cell $\boldsymbol{h_{t-1}}$ as depicted in Fig. 2, is presented below:

$$\boldsymbol{z_{t-1}} = \sigma(\boldsymbol{U_z} P_{t-1} + \boldsymbol{W_z h_{t-2}} + \boldsymbol{b_z}), \tag{6}$$

$$\boldsymbol{r_{t-1}} = \sigma(\boldsymbol{U_r} P_{t-1} + \boldsymbol{W_r h_{t-2}} + \boldsymbol{b_r}), \tag{7}$$

$$\tilde{\boldsymbol{h}}_{\boldsymbol{t-1}} = \tanh(\boldsymbol{U_h} P_{t-1} + \boldsymbol{W_h}(\boldsymbol{r_{t-1}} \odot \boldsymbol{h_{t-2}}) + \boldsymbol{b_h}), \tag{8}$$

$$\boldsymbol{h_{t-1}} = \boldsymbol{z_{t-1}} \odot \tilde{\boldsymbol{h}}_{\boldsymbol{t-1}} + (1 - \boldsymbol{z_{t-1}}) \odot \boldsymbol{h_{t-2}}, \tag{9}$$

where $\boldsymbol{z_{t-1}}$ is the update gate vector, $\boldsymbol{r_{t-1}}$ is the reset gate vector, $\tilde{\boldsymbol{h}}_{\boldsymbol{t-1}}$ is the candidate activation vector, $\boldsymbol{h_{t-1}}$ is the output vector, and $\sigma(\cdot)$ refers to the sigmoid activation function $\sigma(x) = \frac{1}{1+e^{-x}}$, while $\boldsymbol{U_z}, \boldsymbol{U_r}, \boldsymbol{U_h}, \boldsymbol{W_z}, \boldsymbol{W_r}, \boldsymbol{W_h}, \boldsymbol{b_z}, \boldsymbol{b_r}$, and $\boldsymbol{b_h}$ are the weight and bias parameters of the neural network cell [7].

The RNN encoder output $\boldsymbol{h_{t-1}}$ is an encoding of the past consumption that the neural network learns during training. The size of the encoding vector is a hyper-parameter derived from the parameters of the RNN encoder. In this work, we use an encoding with a size equal to 64, after hyper-parameter tuning, which is the number of neurons each GRU cell contains, i.e. $\boldsymbol{h_{t-1}} = (O_1, \ldots, O_{64})$ as depicted in Fig. 2. The encoding along with the rest of the features of $\boldsymbol{I}$, $\boldsymbol{x} = (x_1, \ldots, x_{54})$ in our case, are used as an input feature vector for a Multilayer perceptron (MLP) with 4 hidden layers. Apart from the input layer $\boldsymbol{I_{MLP}} = (\boldsymbol{h_{t-1}}, \boldsymbol{x})$, the MLP consists of a number of hidden layers, and an output neuron which is the energy demand prediction P_t for time slot t, regarding the specific household. Each hidden layer includes multiple neurons (the number of hidden layers and neurons are hyper-parameters), and each neuron uses the previous layer outputs as an input:

$$\boldsymbol{H_1} = ELU(\boldsymbol{w_1}^T \boldsymbol{I_{MLP}} + \boldsymbol{b_1}), \tag{10}$$

$$\boldsymbol{H_n} = ELU(\boldsymbol{w_n}^T \boldsymbol{H_{n-1}} + \boldsymbol{b_n}), \ n = 2, \ldots, 4, \tag{11}$$

$$P_t = \sigma(\boldsymbol{w_5}^T \boldsymbol{H_4} + \boldsymbol{b_5}), \tag{12}$$

where ELU refers to the Exponential Linear Unit activation function, which is defined as:

$$ELU(x) = \begin{cases} x, & x \geq 0 \\ \alpha(e^x - 1), & x < 0. \end{cases} \tag{13}$$

In our case, it is $\alpha = 1$ and we use ELU instead of $ReLU$ since it does not face the dying $ReLU$ problem and leads the cost to zero faster while producing more accurate results. Furthermore, the MLP layers have 500, 100, 50, 10, and 1 neurons respectively after hyper-parameter tuning.

5 DATA EXPERIMENTS

In this section, a detailed experimentation with a real dataset is carried out. The experimental setup and data preprocessing are presented together with evaluation metrics, and the results and main takeaway messages from the experimentation results are discussed.

5.1 Dataset

We use the Pecan Street Dataport [24] dataset, which consists of smart meter energy consumption data from approximately 1000 real U.S. households, along with various metadata and appliance-level measurements for each household. For the clustering part, a subset of 653 houses is utilized with hourly energy consumption measurements from 2012 to 2019, while for the deep learning training phase, a subset of 310 households with data from 2018 to 2019 is used. Besides hourly energy consumption measurements, we also utilize the house area (m^2) and a Boolean feature about solar panel existence.

5.2 Data preprocessing

The majority of energy consumption measurement data have low values, while there are significantly less measurements with high energy values. Neural networks try to perform well on the entire training dataset on average, thus energy peaks are underestimated due to the positive skewness of the demand data. Hence, as part of the data preprocessing procedure, a Box Cox transformation [5] is applied to the energy consumption data before training, in order to transform the data into a normal distribution. The Box Cox transformation is defined as follows:

$$Y_i = \begin{cases} \frac{Y_i^\lambda - 1}{\lambda}, & \text{if } \lambda \neq 0 \\ \log Y_i, & \text{if } \lambda = 0 \end{cases} \tag{14}$$

where Y_i refer to the target variables, which in our case are the energy consumption measurement data, and λ is a parameter selected in order to approximate a normal distribution curve. Furthermore, all input and output features of the deep learning model are normalized into $[0, 1]$ using a MinMaxScaler from the Scikit-Learn Python library[1]. Both transformations are inversed after a prediction takes place, so that the system outputs the appropriate energy consumption value.

5.3 Experimental setup

The dataset consisting of 310 households with measurements from 2018 to 2019 is randomly split into training and test sets with a 80-20 ratio. This means that the deep learning models presented are trained on all of the 310 houses, but only with 80% of the measurements. In addition, a set of 100 different unseen houses to the model is used to test the its predictive performance on new houses. The loss function used for all the neural networks for n observations is Mean Squared Error (MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2, \tag{15}$$

where Y_i refer to real/target observations and $\hat{Y}_i$ are the model predictions. The optimizer used for training is Adam [16], along with early stopping based on a validation set consisting of 10% of the training set. The experiments were conducted using an NVIDIA GTX 1060 6GB GPU.

5.4 Evaluation metrics

In order to individually evaluate the clustering performance, we use the Hopkins statistic and the Davies–Bouldin index. The Hopkins statistic measures a dataset's cluster tendency, i.e. the probability that the data points were generated by a uniform distribution, and is performed before the clustering procedure. This evaluation metric includes a null hypothesis H_0 and an alternate hypothesis H_a, where H_0 implies that the data points are generated by a uniform

[1] https://scikit-learn.org

distribution, and H_a assumes that they are generated by a random distribution which might indicate the presence of meaningful clusters. If $\mathcal{D}$ is the examined dataset, m points $(p_1, \ldots, p_m)$ are sampled from $\mathcal{D}$, and m artificial points $(q_1, \ldots, q_m)$ are generated from a random uniform distribution. The Hopkins statistic [17] is defined as follows:

$$H = \frac{\sum_{i=1}^{m} u_i}{\sum_{i=1}^{m} u_i + \sum_{i=1}^{m} w_i}, \tag{16}$$

where u_i is the distance between each artificial point and the nearest point from $\mathcal{D}$, and w_i is the distance between each point from $(p_1, \ldots, p_m)$ and its nearest neighbour from $\mathcal{D}$. A value of H close to 1 indicates that the examined dataset has a high clustering tendency, while a value close to 0 indicates that the data points are uniformly distributed.

The Davies–Bouldin index (DB index) [9] measures the average similarity between the resulted clusters, by comparing the distances between clusters and their size. Values closer to 0 indicate a better cluster partition, where 0 is the lowest possible value. The DB index is defined as follows:

$$DB = \frac{1}{n_c} \sum_{i=1}^{n_c} D_i, \tag{17}$$

where

$$D_i = \max_{j=\{1,\ldots,n_c\}, j\neq i} R_{ij}, i = \{1, \ldots, n_c\}, \tag{18}$$

$$R_{ij} = \frac{s_i + s_j}{d(v_i, v_j)}, \tag{19}$$

$$s_i = \frac{1}{\|c_i\|} \sum_{x \in c_i} d(x, v_i), \tag{20}$$

where n_c is the number of clusters, $d(\cdot, \cdot)$ is the Euclidean distance, c_i refers to cluster i, and v_i refers to the centroid of cluster i.

The metrics used for the energy consumption prediction model evaluation are the Mean Absolute Percentage Error (MAPE), the R-squared (R^2) metric, and the MSE. The original MAPE definition is:

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{Y_i - \hat{Y}_i}{Y_i} \right|. \tag{21}$$

However, (21) is not defined when $Y_i = 0$, which is possible in the case of energy consumption measurements. Thus we utilize a slight variation of MAPE defined as:

$$MAPE = \begin{cases} \frac{100}{n} \sum_{i=1}^{n} \left| \frac{Y_i - \hat{Y}_i}{Y_i} \right|, \text{if } Y_i \neq 0 \\ \frac{100}{n} \sum_{i=1}^{n} \left| \frac{\hat{Y}_i}{\frac{1}{n}\sum_{i=1}^{n} Y_i} \right|, \text{if } Y_i = 0 \end{cases} \tag{22}$$

where $Y_i = 0$ occurs very few times in our data after the preprocessing phase (i.e. 3 out of 399,465 test measurements).

The R-squared metric measures the amount of variability (of the response data around its mean) that the trained model explains and is defined as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{n} (Y_i - \frac{1}{n}\sum_{i=1}^{n} Y_i)^2}. \tag{23}$$

In other words, R-squared measures the closeness of the predicted regression values to the real measurements.

Table 2: Clustering evaluation.

	Full profile	Time-series	Non-time-series
H	84.5%	86.8%	83.5%
DB	1.81	1.65	1.07
Clusters	5	6	5

5.5 Experiments and discussion

First, an evaluation of the clustering phase is conducted using the Hopkins statistic (H) and the Davies–Bouldin index (DB), while the optimal number of clusters was determined using the elbow method. As presented in Table 2, the case of using the full energy profile to train k-means with Euclidean distance is compared to the case of splitting it through the "double" clustering procedure described in section 4.2. The results show a slight improvement in terms of Hopkins statistic and a significant improvement in terms of DB index when using the "double" clustering approach, through splitting the profile into time-series and non-time-series features and applying the appropriate variant of k-means. Hence, it is evident that the proposed clustering approach results to a better clustering of households compared to a direct application of k-means on the energy profiles dataset.

In Table 3, experiment results for different hourly demand prediction models are depicted. In the first column, we present the average performance of a set of separate MLP models trained for each house that has over 8,000 hours (333 days) of data, namely 159 households. The MLP models have the same architecture with the MLP part of the model described in section 4.4 using the past 24 hour consumption time-series as an additional input directly. An average $R^2 = 69.2\%$ and $MAPE = 30.77\%$ show that the inherent uncertainty and variation of residential energy consumption significantly affect the prediction performance for the models of some households. For instance, the highest and lowest R^2 achieved by a model trained on a single household were 97.3% and 29.5% respectively. Hence, the approach of training a separate model per house does not always lead to accurate hourly energy consumption forecasts. This experiment was also conducted using the same set of 310 households used for the single-model-for-all-houses approach, but had a lower predictive performance achieving on average $R^2 = 67.5\%$ and $MAPE = 33\%$. This happened due to the fact that several houses had a few hours worth of data which are insufficient to train a machine learning model. In other words, the cold-start problem emerged for some of the houses, which is one of the main motivations of the proposed approach.

In the four middle columns of Table 3, we present four variants of the single-model-for-all-houses approach, that achieved high prediction performance on the test set (20% of the energy consumption measurements for 310 households). As depicted in the second column, a single RNN with a look-back window of 168 hours (1 week) trained on multiple houses, achieved a significant performance increase compared to the one-model-per-house approach. The RNN managed to distinguish different houses using the 1-week look-back when making each prediction, without having the energy profiles and cluster distances.

In the third column, we present the results of an MLP with all the input features described in section 4.3, i.e. time features,

Table 3: Experiment results for different variants of the proposed approach.

	One MLP per house (with past data as input)		RNN (with 1-week window)	MLP (with past data as input)	RNN encoder & MLP	RNN encoder & MLP (with 1-week window)	RNN encoder & MLP (with 1-week window)
Number of models	159 models for 159 houses*	310 models for 310 houses	1 model for 310 houses	1 model for 310 houses	1 model for 310 houses	1 model for 310 houses	1 model for 100 unseen houses
R^2	69.2%	67.5%	84.6%	85.2%	85.1%	85.5%	74.8%
MAPE	30.77%	33%	10.9%	10.6%	10.4%	10.1%	12.47%
MSE	0.011	0.013	0.0067	0.0064	0.0064	0.0063	0.0093

* Houses with adequate data, i.e. over 8,000 hours.

energy profiles, and distances from cluster centroids, along with the past consumption values for the past 24 hours/time slots. A slight performance increase is observed compared to the RNN with 1-week window, which makes this MLP architecture a better candidate, especially in cases where data for the entire past week are not available.

In the last three columns of Table 3, results for two variants of the novel RNN encoder and MLP architecture described in section 4.4 are presented. The architecture depicted in Fig. 2 with a 24-hour look-back RNN encoder achieves an R^2 of 85.1% and a *MAPE* of 10.4%, which are approximately the same with the MLP with past data input and slightly better than the RNN with 1-week window. Thus, in the second to last column of Table 3 we present an architecture that combines the best characteristics of all the previous models, namely an RNN encoder and MLP architecture with a 1-week look-back window for the RNN encoder. We are able to extend the RNN encoder look-back from 24 to 168 hours by entirely removing the energy profile from the input features due to computational resource limitations. However, this does not affect the model's performance since we keep the distances from cluster centroids that act as an encoding of the energy profiles. The aforementioned model achieved the best predictive performance in our experiments with $R^2 = 85.5\%$ and $MAPE = 10.1\%$. The results of Table 3 show that the proposed single model trained on multiple houses approach achieves "generalization" by outperforming the one model per house approach. The novel RNN encoder and MLP architecture showed a slight predictive performance increase, but the appropriate variant should be selected according to the case examined. For example, the 1-week look-back variants are not ideal in cases of frequent smart meter missing values and computational limitations.

Furthermore, we tested the trained RNN encoder and MLP model (with 1-week look-back) on a completely new set of houses (last column of Table 3) to determine if our approach achieves "representativeness". That is, the model provides hourly energy demand forecasts for new/unseen households, just by constructing its energy profile with the available data. In order to showcase this, we used a set of 100 houses having most of measurements for 2017. The model has never seen these houses before, i.e. they were not included in the training set. The model achieved: $R^2 = 74.8\%$, $MAPE = 12.47\%$, and $MSE = 0.0093$ on average for those houses. These results show that the proposed model has a lower but accurate prediction performance on unseen houses, compared to houses on which it has been trained. Hence, it achieves "representativeness" by providing reliable forecasts for any new household, and it addresses the cold-start problem.

6 CONCLUSION

In this study we tackle the problem of energy demand forecasting for individual households with a single deep learning model, which discovers different patterns among electricity consumers and provides accurate predictions even for completely new houses. We propose an RNN encoder + MLP architecture that utilizes both past consumption data and the computed energy profiles to make a prediction for a specific household. Experiment results with real data show that the proposed approach with a single deep learning model achieves accurate prediction for multiple households on test data, with a *MAPE* of 10.1% for households included in the training dataset and a *MAPE* of 12.5% for new houses, that were not included in the training phase. Hence, our approach is able to produce reliable forecasts even for completely new, previously unseen households, with very few data.

As a part of our future work, we would like to compare our approach with *DeepAR* [22] to (i) identify the impact on performance of those parts of our architecture that are different from those in *DeepAR*, (ii) make performance comparisons of the two approaches on the same datasets. Future research directions also include the utilization of higher-dimension energy profiles with more features, adding weather forecasts in the model's input feature vectors, and testing the approach with other smart meter datasets to demonstrate its capabilities with households from different countries. It would be interesting to observe the results of the proposed approach with experiments on consumers with diverse characteristics, e.g. from different regions, in order to broaden the model's applicability. Furthermore, other neural network components, such as bidirectional RNNs could be explored.

ACKNOWLEDGEMENT

This work was supported by project InterConnect (Interoperable Solutions Connecting Smart Homes, Buildings and Grids), which has received funding from European Union's Horizon 2020 Research and Innovation Programme under Grant 857237.

REFERENCES

[1] Muhammad Waseem Ahmad, Monjur Mourshed, and Yacine Rezgui. 2017. Trees vs Neurons: Comparison between random forest and ANN for high-resolution prediction of building energy consumption. *Energy and Buildings* 147 (2017), 77–89.

[2] Abdulaziz Almalaq and Jun Jason Zhang. 2018. Evolutionary deep learning-based energy consumption prediction for buildings. *IEEE Access* 7 (2018), 1520–1531.

[3] Christian Beckel, Leyna Sadamori, and Silvia Santini. 2013. Automatic socio-economic classification of households using electricity consumption data. In *Proceedings of the fourth international conference on Future energy systems*. ACM, California, USA, 75–86.

[4] Donald J Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series. In *KDD workshop*, Vol. 10. Seattle, WA, USA, 359–370.

[5] George EP Box and David R Cox. 1964. An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological)* 26, 2 (1964), 211–243.

[6] Spiros Chadoulos, Iordanis Koutsopoulos, and George C Polyzos. 2020. Mobile Apps Meet the Smart Energy Grid: A Survey on Consumer Engagement and Machine Learning Applications. *IEEE Access* 8 (2020), 219632–219655.

[7] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

[8] Peter Cramton. 2017. Electricity market design. *Oxford Review of Economic Policy* 33, 4 (2017), 589–612.

[9] David L Davies and Donald W Bouldin. 1979. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence* PAMI-1, 2 (1979), 224–227.

[10] Cheng Fan, Fu Xiao, and Shengwei Wang. 2014. Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques. *Applied Energy* 127 (2014), 1–10.

[11] Matt W Gardner and SR Dorling. 1998. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment* 32, 14-15 (1998), 2627–2636.

[12] Stephen Haben, Colin Singleton, and Peter Grindrod. 2015. Analysis and clustering of residential customers energy behavioral demand using smart meter data. *IEEE Transactions on Smart Grid* 7, 1 (2015), 136–144.

[13] Tao Hong and Shu Fan. 2016. Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting* 32, 3 (2016), 914–938.

[14] Konstantin Hopf, Mariya Sodenkamp, Ilya Kozlovkiy, and Thorsten Staake. 2016. Feature extraction and filtering for household classification based on smart electricity meter data. *Computer Science-Research and Development* 31, 3 (2016), 141–148.

[15] Konstantin Hopf, Mariya Sodenkamp, and Thorsten Staake. 2018. Enhancing energy efficiency in the residential sector with smart meter data analytics. *Electronic Markets* 28, 4 (2018), 453–473.

[16] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG]

[17] Richard G Lawson and Peter C Jurs. 1990. New index for clustering tendency and its application to chemical problems. *Journal of chemical information and computer sciences* 30, 1 (1990), 36–41.

[18] Tuong Le, Minh Thanh Vo, Tung Kieu, Eenjun Hwang, Seungmin Rho, and Sung Wook Baik. 2020. Multiple electric energy consumption forecasting using a cluster-based strategy for transfer learning in smart building. *Sensors* 20, 9 (2020), 2668.

[19] Chengdong Li, Zixiang Ding, Dongbin Zhao, Jianqiang Yi, and Guiqing Zhang. 2017. Building energy consumption prediction: An extreme deep learning approach. *Energies* 10, 10 (2017), 1525.

[20] Elena Mocanu, Phuong H Nguyen, Madeleine Gibescu, and Wil L Kling. 2016. Deep learning for estimating building energy consumption. *Sustainable Energy, Grids and Networks* 6 (2016), 91–99.

[21] K Muralitharan, Rathinasamy Sakthivel, and R Vishnuvarthan. 2018. Neural network based optimization approach for energy demand prediction in smart grid. *Neurocomputing* 273 (2018), 199–208.

[22] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting* 36, 3 (2020), 1181–1191.

[23] Heng Shi, Minghao Xu, and Ran Li. 2017. Deep learning for household load forecasting—A novel pooling deep RNN. *IEEE Transactions on Smart Grid* 9, 5 (2017), 5271–5280.

[24] Pecan Street. 2021. *Pecan Street Dataport*. pecanstreet.org. https://www.pecanstreet.org/dataport/

[25] Thanchanok Teeraratkul, Daniel O'Neill, and Sanjay Lall. 2017. Shape-based approach to household electric load curve clustering and prediction. *IEEE Transactions on Smart Grid* 9, 5 (2017), 5196–5206.

[26] Zeyu Wang, Yueren Wang, Ruochen Zeng, Ravi S Srinivasan, and Sherry Ahrentzen. 2018. Random Forest based hourly building energy prediction. *Energy and Buildings* 171 (2018), 11–25.

[27] Paul J Werbos. 1990. Backpropagation through time: what it does and how to do it. *Proc. IEEE* 78, 10 (1990), 1550–1560.

[28] Junjing Yang, Chao Ning, Chirag Deb, Fan Zhang, David Cheong, Siew Eang Lee, Chandra Sekhar, and Kwok Wai Tham. 2017. k-Shape clustering algorithm for building energy usage patterns analysis and forecasting model accuracy improvement. *Energy and Buildings* 146 (2017), 27–37.