energies

Article

Implementing Very-Short-Term Forecasting of Residential Load Demand Using a Deep Neural Network Architecture

Reynaldo Gonzalez, Sara Ahmed and Miltiadis Alamaniotis *

Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, TX 78249, USA

* Correspondence: miltos.alamaniotis@utsa.edu

Abstract: The need for and interest in very-short-term load forecasting (VSTLF) is increasing and important for goals such as energy pricing markets. There is greater challenge in predicting load consumption for residential-load-type data, which is highly variable in nature and does not form visible patterns present in aggregated nodal-type load data. Previous works have used methods such as LSTM and CNN for VSTLF; however, the use of DNN has yet to be investigated. Furthermore, DNNs have been effectively used in STLF but have not been applied to very-short-term time frames. In this work, a deep network architecture is proposed and applied to very-short-term forecasting of residential load patterns that exhibit high variability and abrupt changes. The method extends previous work by including delayed load demand as an input, as well as working for 1 min data resolution. The deep model is trained on the load demand data of selected days—one, two, and a week—prior to the targeted day. Test results on real-world residential load patterns encompassing a set of 32 days (a sample from different seasons and special days) exhibit the efficiency of the deep network in providing high-accuracy residential forecasts, as measured with three different error metrics, namely MSE, RMSE, and MAPE. On average, MSE and RMSE are lower than 0.51 kW and 0.69 kW, and MAPE lower than 0.51%.

Keywords: deep neural network; residential load; very-short-term forecasting; small data; individual household; 1 min data; parameter selection analysis

Citation: Gonzalez, R.; Ahmed, S.; Alamaniotis, M. Implementing Very-Short-Term Forecasting of Residential Load Demand Using a Deep Neural Network Architecture. *Energies* **2023**, *16*, 3636. https://doi.org/10.3390/en16093636

Academic Editors: Barbara Cannas and Sara Carcangiu

Received: 9 March 2023
Revised: 18 April 2023
Accepted: 20 April 2023
Published: 23 April 2023

1. Introduction

Growing penetration of information technologies into the operation of electricity markets has allowed a transformation of the role of the stakeholders [1]. Notably, the role of residential customers has evolved from a fully passive to an active one, where their interactions with the rest of the market participants shape the overall load demand and the electricity clearing prices. These interactions are performed via residents' smart meters equipped with intelligent decision-making algorithms [2].

One of the main tasks involved in most of the decisions pertaining to electricity market activities is load forecasting. Forecasting is an integral part of the overall consumer purchase decision strategy [3]. In practice, forecasting of future load demand allows optimal decision-making concerning the cost, as well as the timing, of electricity utilization. Driven by the length of the time horizon, load forecasting may be characterized as very short term, short term, medium term, and long term [4]. Notably, each type serves different purposes and contributes to different sets of decisions.

Regarding a resident's participation in the electricity market, very-short- and short-term horizons are needed for purchasing decisions. Furthermore, the high penetration of information technologies in electricity markets requires residential consumers to decrease the granularity of their forecasts [5]. Technological advances allow residential customers to further refine their electricity demand in terms of minutes, in contrast to hourly demand, which is the prevailing scale in traditional power systems [6].

Given that human residents are not able to monitor the electricity market 24/7 and to subsequently forecast their load demand, implementation of accurate very-short-term forecasting for residential loads becomes a challenge. Reported solutions exhibit a trend in utilizing data-driven methodologies based on tools from the realms of statistics and artificial intelligence (AI). In [7], a linear regression model is discussed for both very-short- and short-term forecasting and tested on datasets taken from smart meter data at the NIT Patna campus. Results show that the method performs well as a fast method that can be applied to multiple time horizons considering various load types. The inclusion of larger building type loads, however, creates visible patterns and removes the high variability present in residential-only loads. A deep autoformer method is presented in [8], which aims to improve the high memory cost, computational complexity, and long training time. Results show that the autoformer method is able to predict more irregular data patterns and frequent fluctuations; however, some downsides include the need for pre-processing and added system complexity of the autoformer framework. In [9], a two-level load forecasting is implemented, consisting of an ANN in the first level for time-series prediction and a feed-forward ANN in the second level for improving accuracy by using more training data. Results show improvement when the scheme is used for prosumer operation cost optimization. One potential downside is the requirement of a whole month's worth of data and the resulting training time for such a large dataset. A deep network that includes a long-short-term memory network (LSTM) is presented in [10]. Forecasts are performed for 1 h ahead, considering a residential load dataset. The LSTM is combined with a sequence-to-sequence architecture, which improved on the basic LSTM structure. Results showed that standard LSTM fails at 1 min resolution, while the proposed architecture was able to perform well on both 1 min and hourly data. One potential downside is the requirement for a large dataset when using LSTM networks, which require more input data. The authors left as future work exploring other deep architectures. The combination of an LSTM and a convolutional neural network (CNN) for 10 min forecasts of residential appliances demand is presented in [11]. This work presents another improvement upon the standard LSTM and results show improvement in comparison to support vector machines, gradient boosting machine, and random forest. Appliance-level data may, however, not always be available.

In [12], the extreme gradient boosting (XGBoost) method is compared to several other methods for the load consumption of a warehouse. Results show the XGBoost method outperformed the others in terms of RMSE, MAE, and MAPE. The data used, however, is of hourly resolution and for a warehouse, which differs from residential profiles. In [13], individual household data is utilized while addressing the gap on methods that work for multiple forecasting horizons. Bayesian networks are used on Irish smart meter data, and results showed that the proposed method showed improvement through NRMSE. The time resolution used was half-hourly datapoints. An attention-based temporal–spatial convolutional network (ACN) is proposed in [14], which aims to improve upon temporal convolutional networks (TCN). Results show improvement in MAPE and RMSE compared to benchmark TCN methods using hourly datasets from China, which are available by request. In [15], the Markov-chain mixture model (MCM) is used to forecast one step ahead on 30 min residential data from Australia. Results show that MCM performs similarly to quantile regression. The work in [16] aims at high volatility due to new energy sources and proposes phase space reconstruction (PSR) for load forecast at the substation level. Results show that the PSR method works better than the benchmarks when the forecasting horizon is greater than half an hour.

Other recent works propose new methods; however, they are only applied to the short-term forecast horizon. In [17], a graph neural-network-based framework is proposed, which captures the hidden spatial dependencies of different houses. The work also provides analysis of computational timing, and results show improvement compared to baseline methods. While the source of the data is given as houses from Los Angeles, New York, and Texas and is available (OpenEI), the data is aggregated by selecting random houses from each location, making the task of replicating the dataset a difficult one. Ensemble methods are presented in [18,19], which show improvement compared to baseline methods; however, the methods are applied to 30 min and hourly resolution datasets. A bi-directional LSTM network is presented in [20], in which an analysis comparing six training data configurations with four prediction methods is performed, resulting in the bi-directional LSTM yielding the most accurate results. A further point to consider in this work is the requirement for more data for bi-LSTM networks. Recurrence plots are used in [21] to create a 2-D input into a CNN. The proposed method is able to show accuracy improvement over the 1-D counterpart. One of the future areas of work noted is exploring the use of deep learning.

Graph spectral clustering is used in [22], where aggregate power is decomposed into appliance-level consumption, and each appliance is forecasted separately. The separate forecasts are combined to get the total consumption forecast, and the results show superior performance in comparison to other methods. A gated recurrent unit (GRU) neural network is presented in [23], which is used for the load forecasting of a residential community. Results show similar performance of the GRU compared to LSTM; however, the GRU shows shorter simulation time. LSTM combined with recurrent neural network is utilized in [24], where focus is placed on forecasting at the individual household level. Results show that the LSTM-RNN framework does the best forecasting compared to several benchmarks. A future direction of research noted is the methodology for parameter tuning. Deep neural networks are used in [25] for STLF on appliance-level consumption, and results show improvement compared to SVM. Different data resolutions of 1 min, 15 min, and 30 min are used, but final data configurations are not presented. A feed-forward ANN with Nesterov learning is featured in [26], which is compared with six other algorithms. Results show that the method outperforms the benchmarks; however, the timing horizon is hourly, leaving VSTLF to still be explored. Because many of the works investigating DNN and other neural network structures show promising use but have yet to be implemented for a VSTLF horizon, we infer that DNN for VSTLF is a desirable topic to explore. In this paper, a new deep neural network (DNN) architecture is developed and tested for very-short-term forecasting of residential loads. The current work builds on our preliminary work and the promising results that were obtained, as presented in [27]. In the current work specifically, we propose a deep feedforward neural network architecture and extensively test it for forecasting residential loads for a very-short-term ahead of time horizon on various days from different seasons and on a set of special days.

One of our main contributions is the application of the method to 1 min data. Of the aforementioned works, very few focus on 1 min resolution. Table 1 shows a summary of the literature with methods and data granularity used. The data used in [7] are aggregated into 10 min intervals, and no reference is given for online availability. The data in [8] are in 15 min intervals, and forecasting is done 1 day ahead (15 min) and 3 days ahead (hourly). The data in [9] are hourly data taken from commercial and residential profiles, resulting in recognizable load patterns. In [10], individual household data (UCI) is utilized at 1 min resolution. The same data is utilized in [20] and [21]; however, the data is aggregated into hourly and 15 min intervals, respectively. The data in [12–15] are all either hourly or 30 min. Data used in [16] are of 5 min resolution; however, they are at the substation level. The work in [11] utilizes publicly available data from a Belgium house with data in 10 min intervals. Regarding [17–20,23,24], all consider data that are in (or have been aggregated to) hourly or 30 min time resolution. In [22], 1 min resolution data is used; however, the signals are decomposed, which makes replication of the work difficult. In [25], data of 1 min, 15 min, and 30 min resolution are used; however, the final aggregated configurations are not presented. Lastly, in [26], data resolution of 15 min is used. Of all the references presented, only one [10] uses data with a resolution of 1 min without utilizing appliance-level features. Aside from the time resolution, the other contributions of this work include an additional input feature, which extends the previous work, as well as the method working with small data (3 days training).

Table 1. Literature review comparison.

Ref.	Year	First Author	Horizon	Approach	Data Gran.
[7]	2018	Vats	VSTLF	Linear regression	15 s
[8]	2022	Jiang	VSTLF	Deep-autoformer neural network	15 min
[9]	2020	Faraji	VSTLF	First level: MLP-ANN; second level: FF-ANN	Hourly
[10]	2016	Marino	VSTLF	LSTM with sequence-to-sequence architecture	1 min
[11]	2019	Ploysuwan	VSTLF	CNN for feature learning and LSTM layers for time series	10 min
[12]	2022	Ribeiro	VSTLF	Extreme gradient boosting (XGBoost)	Hourly
[13]	2020	Bessani	VSTLF	Bayesian networks	30 min
[14]	2023	Tong	VSTLF	Attention-based temporal–spatial convolutional network (ACN)	Hourly
[15]	2021	Munkhammar	VSTLF	Markov-chain mixture distribution model (MCM)	30 min
[16]	2019	Shi	VSTLF	Phase space reconstruction (PSR)	5 min
[17]	2021	Lin	STLF	Graph neural-network-based forecasting framework	Hourly
[18]	2023	Botman	LTF	Ensemble: clustering with k-means, prediction with ensemble	30 min
[19]	2019	Ves	VSTLF	Ensemble: gradient boosting regression, MLP and LSTM networks; linear regression	(Agg. to hourly)
[20]	2022	Atef	STLF	6 input feature set scenarios with 4 prediction methods (LR, SVR, Uni-LSTM, and bi-LSTM) that result in analysis of 24 configurations	(Agg. to hourly)
[21]	2019	Rajabi	STLF	Recurrence plots fed to CNN to make the first layer 2-D instead of directly feeding time-series	(Agg. to 15 min)
[22]	2019	Dinesh	STLF	Aggregate power signal is decomposed into individual appliance signals and each appliance's power is forecasted	1 min
[23]	2019	Zheng	STLF	Gated recurrent unit (GRU) neural network	Hourly
[24]	2019	Kong	STLF	LSTM-RNN	30 min
[25]	2018	Din	STLF	FF-DNN and R-DNN	1, 15, 30 min
[26]	2019	Oprea	STLF	7 algorithms: Momentum, Nesterov, Nesterov + backtracking, NARX, DNN, GTB, RF	15 min

The contributions of this paper are summarized as follows:

- Additional input of load demand that is delayed by 10 min;
- Investigation of the DNN structure for VSTLF;
- Dataset considering 1 min residential data;
- Commonly available dataset used;
- Method that relies on small data (only 3 days required).

The current manuscript is organized as follows. The next section provides a brief introduction to neural networks and deep architectures, while Section 3 discusses the challenges of residential loads. Going further, Section 4 presents the materials and methods used, including the developed network architecture. Section 5 presents and discusses the obtained results. Section 6, the last section, summarizes the main points of the paper and provides future research directions.

2. Elements of Deep Neural Networks

Nowadays, artificial intelligence is the driving force of the new industrial revolution that is under way, and together with advanced information and communication technologies, it has transformed our world into a "digital world". There are a variety of AI tools that have been proposed and have been applied in autonomous systems and decision making. Among those tools, artificial neural networks (ANN) are the most widely used, mainly because of their universal approximation property, which makes them ideal solutions for various complex problems where little is known about the internal structure of the process [28].

ANN is an information processing system that takes an input and, through a series of computations, provides an output. The structural unit of an ANN is the artificial neuron, whose architecture mimics that of a biological neuron [29]. The inputs of the neuron are multiplied with their respective weights, and a set of factors are obtained. The factors are summed and are then fed to an activation function, whose output coincides with the final output of the neuron.

The neural network is constructed by interconnecting individual neurons. Interconnection of neurons is implemented by forwarding the output of a neuron as an input to another neuron. Grouping those neurons based on their interconnections allows the neural network to form layers [29].

The input layer is not a computing layer; its role is to take the input and forward it to the next layer. The output layer is the set of neurons that provides the final output of the neural network, while between the input and the output, there is a hidden layer. The number of hidden layers can be more than one (theoretically, a network may have infinite hidden layers but only one input and one output). It should be mentioned that based on the number of hidden layers, neural networks are characterized as shallow or deep. In particular, a shallow network contains up to two hidden layers, while any network with three or more is characterized as deep [28]. The overall number of layers is called the *depth* of the network [29].

Notably, the depth of a deep neural network is at least five: an input, three hidden, and an output layer. Deep networks have revolutionized the information processing domain by offering new opportunities for large-scale processing and have found wide use in image processing—in the form of convolutional networks.

The main strength of the deep architectures is that each layer provides a different abstraction of the input data. In other words, the hidden layers provide different information (representation abstraction) of the input, thus implementing a hierarchical type of representation. This type of representation is useful in representation in the analysis of complex systems, as each layer models a different level of complexity.

Load forecasting is a complex process in which stochasticity prevails. There is no specific model representation of the load demand given that it depends on a set of random factors. Regarding residential load, these factors become even more stochastic—depicted in the form of high volatility in the load pattern—given that the human activities and habits in the residential building cannot be accurately predicted. However, we expect that the DNN and its hierarchical representation will be able to capture the load dynamics of the residential consumer in the very-short-time horizon [30].

3. Residential Load

Load demand patterns are classified into three categories based on the type of consumer: industrial, commercial, and residential. Each of the three categories exhibits a set of features that is unique for every type of consumer. On one hand, industrial and commercial consumers exhibit high load demand at specific times of the day, while their pattern shows seasonality and low variability. On the other hand, residential load patterns depend on various stochastic factors and are subject to human behavior—for instance, activities and habits such as cooking, washing laundry, and other uses of appliances and electronics. There is a high variety of activities that are performed at a residence that do not fall under specific timelines; those activities contribute to the overall volatility that is present in a residential load pattern [26]. For visualization purposes, two examples of daily load demand patterns of a residence are presented in Figure 1, where the volatility in demand is apparent.

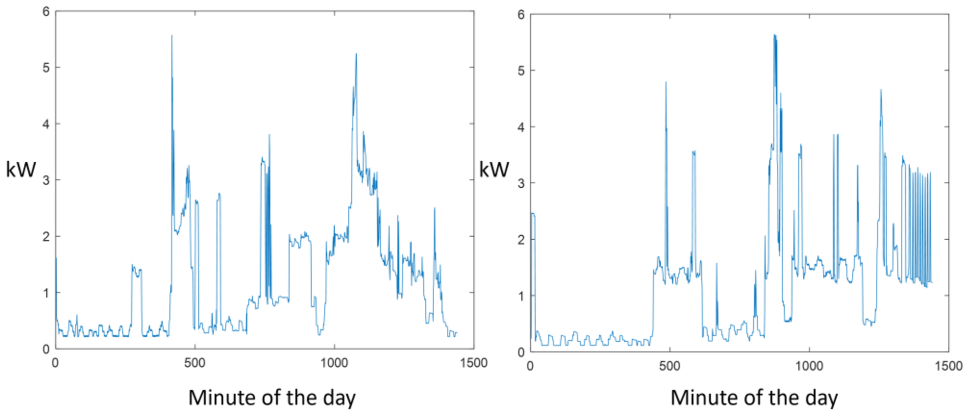

Figure 1. Example electric load demand patterns for a residential consumer.

With the increasing penetration of AI as well as new energy sources into our daily activities, the residential load demand pattern only becomes more stochastic. Smart appliances equipped with AI capabilities can make autonomous decisions with little or no input from the human consumer [2,31]. Furthermore, the adoption of power generation technologies, such as PV panels, has transformed the consumer into a prosumer: the residential consumer is able to generate part of their consumed electricity. This transformation has resulted in new load dynamics concerning the residential consumer, thus altering the load demand pattern. In particular, part of the load demand is satisfied internally; thus, the amount of electricity required to be purchased from the grid diminishes [32].

From the above discussion, forecasting of residential loads is a highly challenging process that becomes even more challenging with the increasing penetration of advanced technologies into our everyday lives. It should be noted that residential forecasting becomes even more challenging for very short prediction horizons (e.g., 1 min ahead of time), given that variability and uncertainty is higher at a smaller time resolution.

Under these conditions, accurate forecasting of the residents' load demand will allow them to make optimal decisions regarding their load management. This makes forecasting methods highly important for the consumer in order to satisfy their maximum demand and minimize their electricity purchasing costs [3,33]. It should be emphasized that in an information-rich environment, such as those of smart grids, where the time granularity of the time decision grows thin, purely data-driven methods prevail as the main forecasting methods. AI and machine learning offer the necessary tools to build methods that are accurate as well as fast in the forecasting of residential load demand with no input or interference from the human resident [3].

4. Materials and Methods

This section describes the materials and methods used to obtain results. The neural network toolbox in Matlab 2020b is used to train and test the deep neural network. A PC with Intel Core i9 is used with 32 GB RAM. The structure of the network is shown in Figure 2; it is comprised of: 2 inputs, 3 hidden layers with 10 neurons each, and 1 output layer with 1 neuron which gives a single output.

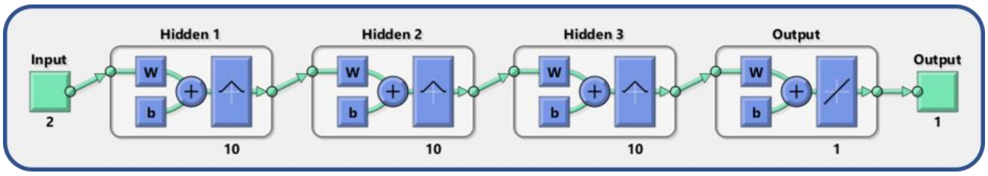

Figure 2. Deep neural network structure.

4.1. Deep Neural Network Very-Short-Term Residential Load Forecasting

The goal of this section is to present the developed method for residential load forecasting. The proposed method is tailored to make predictions for 1 min ahead of time, which is characterized as very short term. The cornerstone of the proposed method is the utilization of a DNN architecture and its training with the inputs of minute-of-day and load values (resolution of 1 min). The minute-of-day captures previous load habits of the resident, while the delayed load demand captures the most recent load dynamics. The block diagram of the proposed method is given in Figure 3.

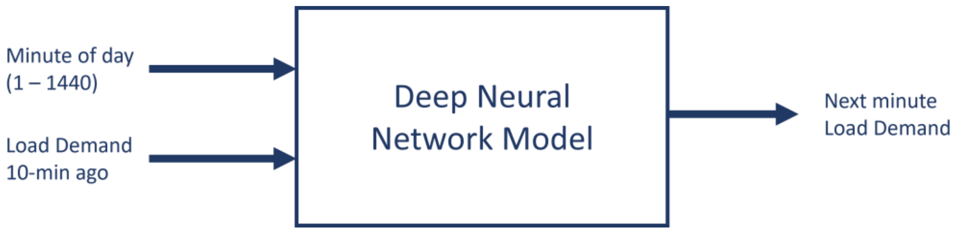

Figure 3. Block diagram of the proposed very-short-term load forecasting.

We observe that the deep network takes two inputs: minute-of-day, which is a single value ranging from 1 to 1440, representing each minute of the day; and the load demand, which was measured 10 min ago. Likewise, the output provides a single value, which is the forecast demand of the next minute.

Training of the DNN architecture is performed by utilizing the 1 min real load demand of: one day before, two days before, and the respective day one week before. Figure 4 exhibits an example of the training dataset assembly process adopted in the current work. The goal of the assembled dataset is to capture the most recent load dynamics (recent days), as well as the general behavior of the resident (week before) [4,27]. It should be noted, we refer here to the "target day" as the day for which a specific network is trained. We choose this term over "test day" because it is the only day for which the network is trained. Each time the target day changes, the training data is re-assembled, and the network is re-trained. Moreover, the "target day" is separate from the target data used in training, which comes from the three training days marked in blue.

Figure 4. Training dataset assembly for training the DNN.

The training and testing procedure is explained through the flowchart in Figure 5. First, the target day for the network is selected. Second, the respective raw training data for the target day is collected, as shown in the example in Figure 4. The minute-of-day input is constructed from a numeric vector, which increases from 1 to 1440 and is repeated for each day. The load values of the three days are taken as both input and output to the network, which are delayed appropriately in the next step. To create the delays, the first 10 samples of the output load data are removed. Similarly, the first 10 min-of-day inputs are removed. Finally, the last 10 samples of the input load data are removed. An example of the input and outputs, both raw and after applying the delay, is shown in Figure 6. Once the delays are applied, the size of the training input/output is reduced from 4320 samples to 4310 samples. The network is then trained using the Levenberg–Marquardt training algorithm. Next, a prediction is made for the target day by feeding the minute-of-day and delayed load inputs using the same delaying procedure, which produces test inputs of a size of 1430 samples. Lastly, the performance of the network for the target day is calculated using the performance metrics.

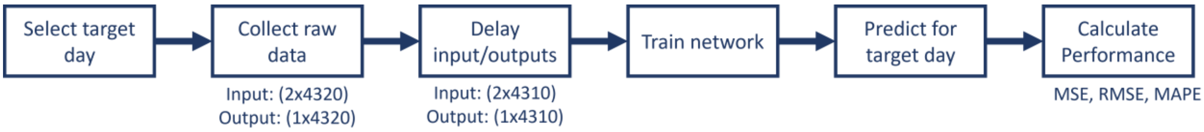

Figure 5. Flowchart of methodology.

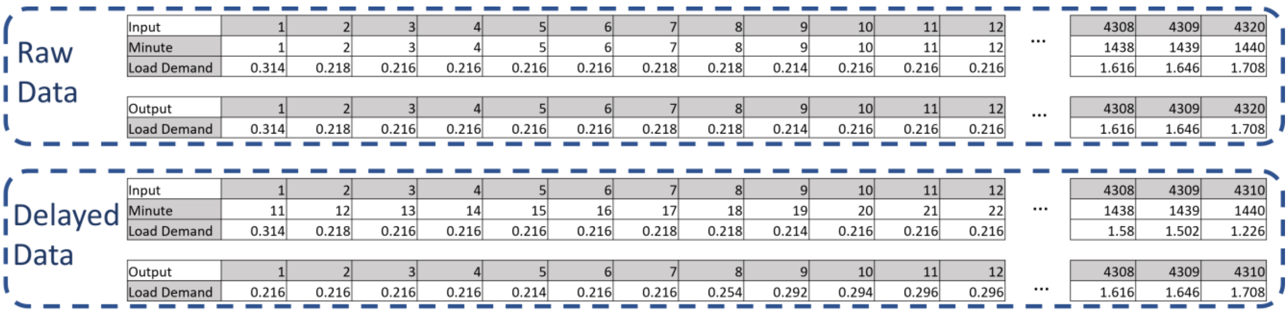

Figure 6. Sample dataset showing raw and delayed data.

Since the network is valid for one day only, training of each network would occur on a daily basis. In this work, 32 days are chosen as target days. One of the benefits of the proposed method is that it requires a smaller amount of data in comparison to other works. The data division ratio used is 90% for training, 5% for validation, and 5% for testing. The actual target day is used as testing data to evaluate the performance of each trained network. This also tests the generalization capability of each trained network by testing on unseen data. In this work, no days that have missing data are utilized. Sporadically missing values could be interpolated without significantly affecting the results. For a large amount of missing data, in the range of hours or days, alternative days can be used for training. Investigation into alternative training days in the case of large amounts of missing data is left as future work.

4.2. Residential Load Datasets

The testing of the forecasting method is performed on a set measured from a residential household building, which is found in [34]. Specifically, the dataset contains load measurements from a single household taken in Sceaux, which is located near the city of Paris in France. Overall, the dataset contains 1 min measurements for the period of December 2006 to November 2010 [34]. The data used in this work is from the year 2007, which is the same year used in [10]. One thing to note is that the important feature of the data is the high variability of the residential load, which would also be present in a more recent dataset.

The assembled test set consists of 32 days containing 4 weeks, one from each season—that is, 28 days—and 4 special days from the year 2007. In particular, the test dataset contains the week of 11–17 February (winter season), 13–19 May (spring season), 12–18 August (summer season), 11–17 November (fall season), and the special days of Easter (8 April), Memorial Day (28 May), Thanksgiving (22 November), and Christmas Day (25 December).

4.3. Deep Neural Network Parameter Analysis

Analysis is performed to determine the optimum number of layers and number of neurons for each layer. The procedure for evaluating the network parameters is shown in Figure 7. A network parameter (e.g., layers or activation function) is first changed, and then the network is trained and evaluated for 32 target days. The performance metrics are calculated for the prediction of each target day and then averaged over the total 32 days. Averaged performance metrics results are then compared to select the parameter for the network. A total of 8 layer and neuron combination scenarios are tested, as well as 15 different activation functions.

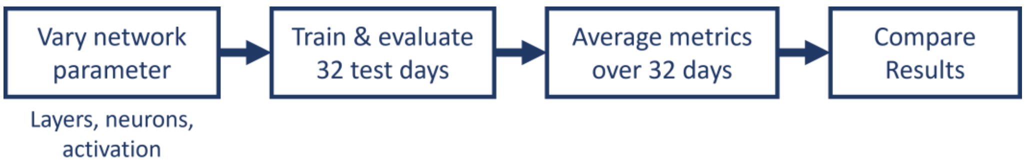

Figure 7. Network parameter selection procedure.

5. Very-Short-Term Forecasting Results

In this section, the presented DNN is applied on a set of load data taken from a real-world residential customer. The obtained results are discussed, and the main observations are highlighted below.

5.1. Performance Metrics

The presented DNN method is applied to the test datasets, and the results are recorded. The accuracy of prediction is quantified utilizing three different error measures that are given below [35]:

(i) Mean square error (MSE):

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(R_t - P_t)^2 \tag{1}$$

(ii) Root mean square error ($RMSE$)

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(R_t - P_t)^2}{n}} \tag{2}$$

(iii) Mean absolute percentage error ($MAPE$)

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{R_t - P_t}{R_t}\right| \quad (3)$$

where R_t is the real value, P_t is the predicted value at timepoint t, and n represents the number of forecasts.

Three different measures are adopted in order to provide a holistic description of the forecasting performance. Each of the measures has some properties and disadvantages. At this point, it should be noted that MAPE is the widely used error metric for load forecasting, given that it scales down the error in the interval 0–100%, which is well understood. However, the problem of MAPE is that it is defined as a ratio, and there is a possibility that the denominator may be zero, in which case MAPE is not defined. MSE and RMSE do not share that problem, but MSE may become very large, while RMSE is prone to outliers (if one point provides a very large error and the rest very small, RMSE will be large).

5.2. Parameter Analysis Results

Parameter analysis for the number of layers and number of neurons is shown in Table 2. There are eight scenarios consisting of various combinations of layers and numbers of neurons. The first column describes a vector that contains the number of neurons for each layer. The rows which are highlighted green show the least value achieved in each metric. It can be seen that the best result achieved in terms of lowest MSE and RMSE is the scenario consisting of three layers with five neurons in each layer. This structure is then chosen for the next step in parameter analysis.

Table 2. Analysis of layers and number of neurons.

Layers	MSE	RMSE	MAPE
[10]	0.6111	0.7422	0.5072
[5 5]	0.5810	0.7230	0.4782
[10 10]	0.7271	0.7946	0.5429
[5 5 5]	0.5355	0.7050	0.4841
[10 10 10]	0.7789	0.8325	0.5607
[20 20 20]	1.4297	1.0954	0.7507
[10 10 10 10]	0.7887	0.8379	0.5866
[10 20 10]	2.0240	1.1113	0.7311

The next step in parameter analysis is the analysis of activation functions. Fifteen different activation functions are considered, and the process in Figure 7 is followed to obtain the average metrics for the 32 target days considered. Table 3 shows the results for each activation function, which is applied on the set of 32 target days. Rows which are highlighted green show the least value achieved in each metric. The activation function that resulted in the lowest MSE and RMSE is "satlin", which is a positive saturating linear transfer function and is chosen as the activation function for the proposed deep network.

5.3. Results and Discussion

Results showing a comparison of the proposed method to other prediction methods are shown in Table 4. Rows which are highlighted green show the least value achieved in each metric. Results of the proposed DNN with three layers of five neurons each and using the "satlin" activation function are compared to three other prediction methods, namely fine tree regression, fine Gaussian support vector machine (SVM), and the ANN with a single layer of 10 neurons, which is considered a shallow network. It can be observed that the best results in terms of least MSE and RMSE are obtained by the proposed DNN.

Table 3. Analysis of activation functions.

Activation Function	MSE	RMSE	MAPE
compet—competitive transfer function	0.9925	0.9799	1.5295
elliotsig—Elliot sigmoid transfer function	0.5246	0.6965	0.4950
hardlim—positive hard limit transfer function	1.2279	1.0757	1.5222
hardlims—symmetric hard limit transfer function	1.2020	1.0652	1.4746
logsig—logarithmic sigmoid transfer function	0.5631	0.7160	0.4870
netinv—inverse transfer function	3.1747	1.2561	0.6894
poslin—positive linear transfer function	0.5165	0.6858	0.4616
purelin—linear transfer function	0.5015	0.6793	0.5595
radbas—radial basis transfer function	0.5709	0.7211	0.4754
radbasn—radial basis normalized transfer function	0.6365	0.7612	0.5240
satlin—positive saturating linear transfer function	0.4954	0.6748	0.4658
satlins—symmetric saturating linear transfer function	0.5384	0.7039	0.4914
softmax—soft max transfer function	0.5658	0.7177	0.4923
tansig—symmetric sigmoid transfer function	0.6436	0.7719	0.5533
tribas—triangular basis transfer function	0.5832	0.7350	0.5886

Table 4. Comparison of different prediction methods.

Method	MSE	RMSE	MAPE
Regression (fine tree)	0.8808	0.9064	0.6413
SVM (fine Gaussian)	0.6429	0.7672	0.4737
ANN (10 neurons)	0.6111	0.7422	0.5072
DNN ((5 5 5) neurons, satlin)	0.5082	0.6832	0.5040

Detailed results of the proposed method are explained next. Table 5 provides the descriptive statistics of the error for each week, as well as for the whole test dataset. Table 5 contains the minimum (min), maximum (max), and mean values for the time period under study—the four weeklong periods, and the lumped special days.

By inspecting Table 5, it can be seen that the interval in which MSE lies is 0.0386–1.2649, while the mean MSE for the whole test set is equal to 0.5082 kW. This error is characterized as low [2,36], highlighting the efficiency of the method for forecasting the 1 min load demands of the residential consumer.

Likewise, RMSE values lie within a short range of values, with the maximum value being equal to 1.1267 kW and the minimum value equal to 0.1965 kW. The minimum value is observed in the summer week, while the maximum is observed in the winter. The summer provides the lowest RMSE, as was the case for MSE. Generally, for both MSE and RMSE, we observe that the weeks with the highest variation of error values are in summer and winter. The overall mean RMSE can also be categorized as low at 0.6832 kW.

Lastly, by checking Table 5 regarding the MAPE values, our observations differ from those made for MSE and RMSE. Notably, the MAPE values lie within a shorter interval when compared to MSE and RMSE, but the overall tested mean values are still very low (i.e., the mean is equal to 0.5040%). Thus, MAPE further confirms the efficiency of our method in very-short-term residential load forecasting. Furthermore, as opposed to MSE and RMSE, which show the lowest error in the summer season, the lowest MAPE is computed for the fall week.

Table 5. Descriptive statistics of the three error measures obtained for the test dataset.

Test Set	MSE			RMSE			MAPE		
	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
Winter week	0.3396	1.2694	0.5784	0.5828	1.1267	0.7408	0.3652	0.6956	0.4942
Spring week	0.1239	0.7427	0.5404	0.3520	0.8618	0.7173	0.4496	0.7296	0.5154
Summer week	0.0386	0.9874	0.3274	0.1965	0.9937	0.5138	0.4004	1.0794	0.6138
Fall week	0.3171	0.8513	0.5554	0.5631	0.9226	0.7372	0.2623	0.7152	0.4087
Special days	0.1712	0.8967	0.5629	0.4137	0.9470	0.7248	0.3933	0.5927	0.4762
Overall	0.0386	1.2694	0.5082	0.1965	1.1267	0.6832	0.2623	1.0794	0.5040

The obtained results per week and per special day for each measure are given in Figures 8–10 in the form of comparative bar graphs of the MSE, RMSE, and MAPE, respectively. Similar to Table 5, it can be observed in Figures 8 and 9 that the week with the lowest MSE and RMSE is in summer. The special day with the lowest MSE and RMSE is Easter Day.

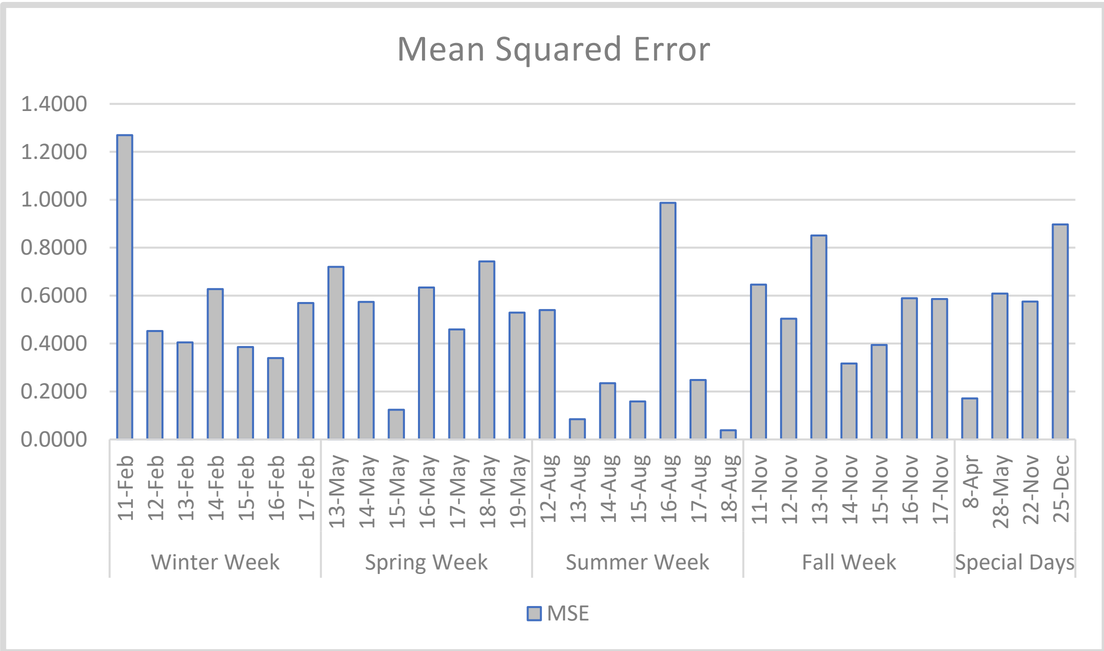

Figure 8. Forecasting error results obtained with MSE.

The lowest MAPE of all weeks is for the fall week, which is evident in Figure 10. Winter, summer, and spring weeks exhibit similar mean MAPE of 0.4942%, 0.5154%, and 0.6138%. The lowest MAPE among the special days is for Thanksgiving Day. We conclude that even though the special days exhibit different load dynamics—given that the behaviors of residential consumers change—the proposed DNN method attained a high accuracy of forecasting (average MAPE: approximately 0.5040%).

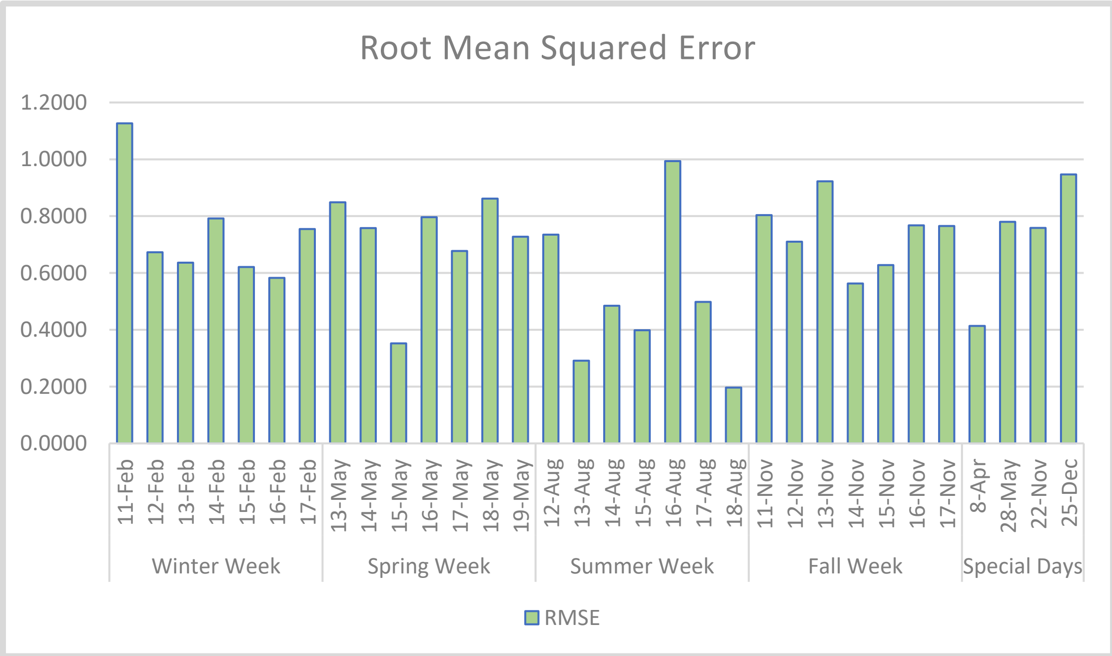

Figure 9. Forecasting error results obtained with RMSE.

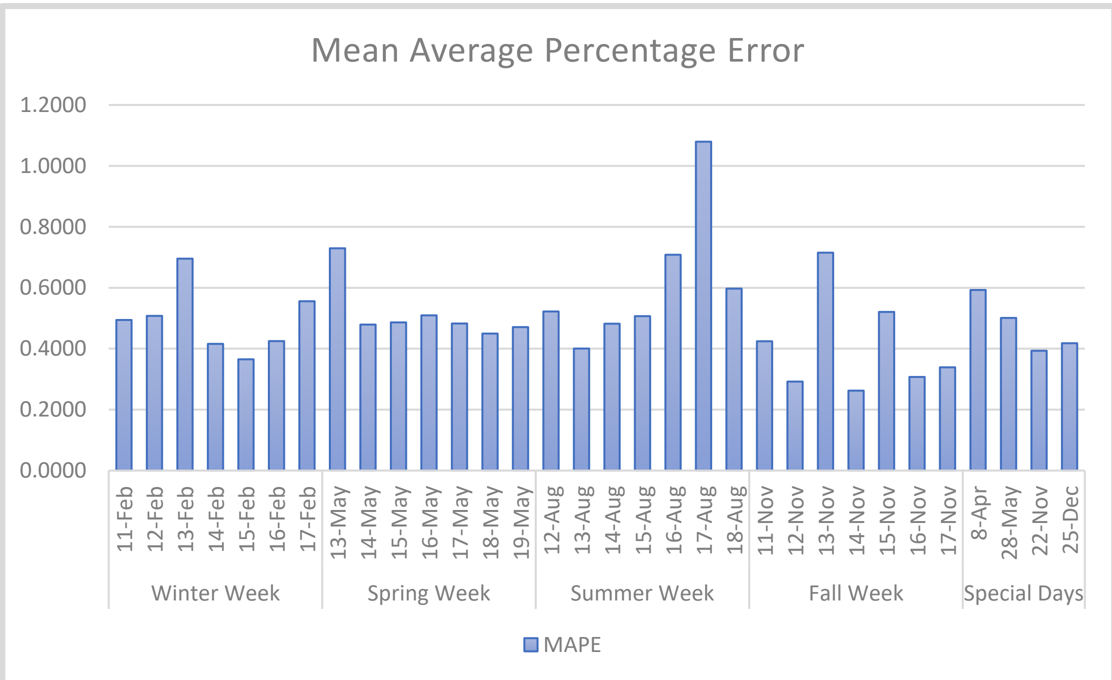

Figure 10. Forecasting error results obtained with MAPE.

Lastly, for visualization purposes, in Figure 11 we provide the forecasted curves against the real demand for the summer week test days. It is also evident that the DNN architecture provides forecasts that closely follow the 1 min real load demand (i.e., 1440 forecasts per day). It should be noted that the summer week plots exhibit a higher error during weekends than weekdays. This is justified by the common fact that the residents spend a higher amount of time at home and perform a higher number of consuming activities (i.e., cooking) on weekends, which makes the electricity load pattern more volatile as compared to weekdays. It is likely that this volatility may be captured with a different assembled training dataset.

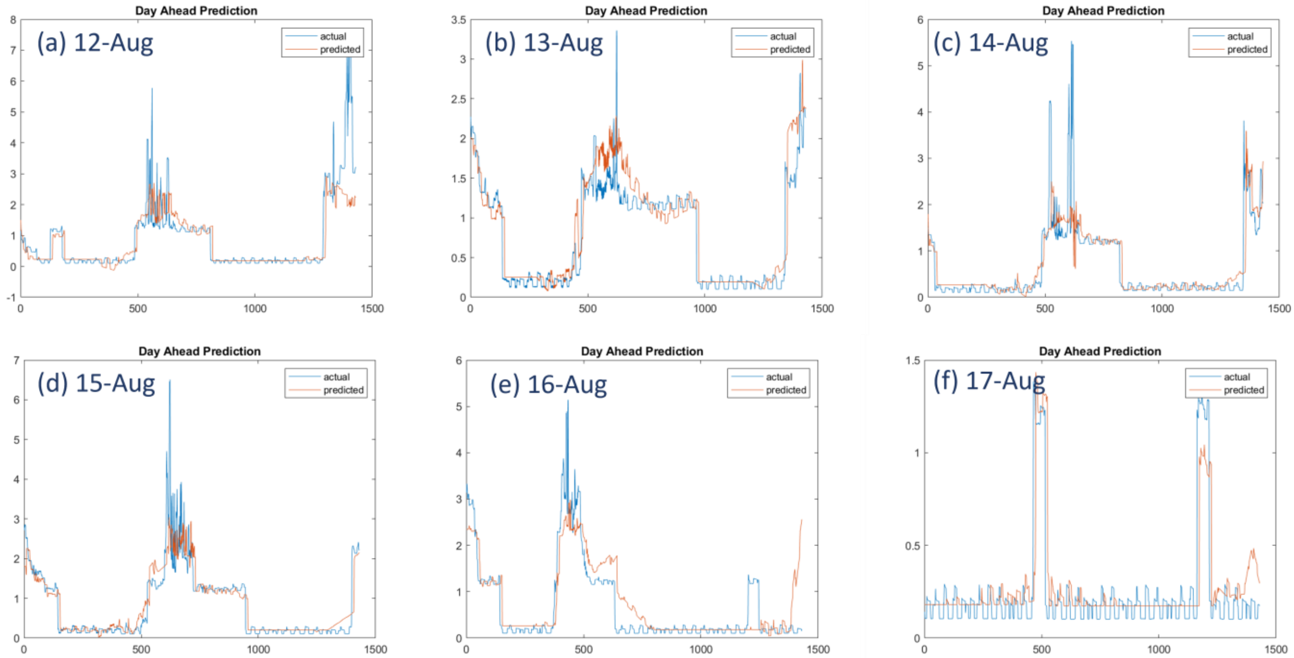

Figure 11. Daily forecasted curve against actual residential load demand for fall week.

It should be emphasized that in Figure 11, we clearly see the stochasticity of the residential load as well as the abrupt changes in loads. More particularly, the real curve (blue line in Figure 11) exhibits high variation, while the peak load appears to be shifted from day to day. For instance, there is a peak around the 520th min of the day on 12 August, while the next day, this peak is shifted to the 560th min, and two days after, there are two peaks, one at the 520th min and a second at the 550th min. Further shifting occurs for 15 August, where the peak appears around the 530th min, and for 16 August the peak has been shifted to the 440th min. Lastly, for 17 and 18 August, which coincide with a weekend, there is a peak around the 500th min, while those peaks have changed shape and are not as high as those of the weekdays. Additionally, to further illustrate the stochasticity of the residential load, we observe that there is a period of high variation load in the middle of the day for the weekdays, while this intense variation is not present on Saturday (17 August) and appears in the last third of the day on Sunday. Overall, based on observation of the load patterns in Figure 11, we observe that there are a few areas that exhibit smooth load demand.

6. Conclusions

In this paper, the implementation of a deep neural network architecture for performing very-short-term residential load forecasting was presented. Research on this horizon length for residential forecasting is very premature, and this paper contributed to that end. DNNs have been successfully used in various types of load forecasting, and the current work exhibits the ability of the proposed DNN architecture to provide highly accurate forecasts of residential demand.

The method was tested on a set of real-world data taken from a residence in France for 4 weeks from 4 different seasons and for 4 special days. Testing included the forecasting of 1 min ahead of time residential load for the whole day, providing an overall set of 1430 forecasts per day. Results obtained in the form of MSE, RMSE, and MAPE showed that the DNN provides low forecasting errors, while the obtained error values are close to each other (i.e., lie within short ranges).

The proposed method shows that DNN is effective for use in VSTLF and how accurate predictions can be made while utilizing a smaller amount of data. The method would be useful for pricing markets and demand-side management. For example, an EV charging algorithm may utilize load forecast to determine the best (most cost-efficient) time to charge an EV. Optimization algorithms for distribution network analysis also stand to greatly benefit from VSTLF. By using forecasted load data, optimization could be performed on future network conditions, leading to better decision making.

Future work will focus on testing the proposed DNN using different training dataset assemblies. In addition, the integration of DNN with fuzzy logic [37] to capture data uncertainty will also be investigated. Lastly, one of the main future directions would be the implementation of a DNN method that is adaptive to the type of forecasted day (weekday vs. weekend) and can capture abrupt changes in the data with the use of a time window.

Author Contributions: Conceptualization, M.A. and R.G.; methodology, R.G.; software, R.G.; validation, S.A. and M.A.; data curation, R.G.; writing—original draft preparation, M.A. and S.A.; writing—review and editing, S.A. and M.A.; supervision, M.A. and S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This project and the preparation of this paper were funded in part by monies provided by CPS Energy through an agreement with The University of Texas at San Antonio. (Project# 1000004785).

Data Availability Statement: The data used in this work is titled "Individual household electric power consumption Data Set" and is openly available from the UCI Machine Learning Repository at: https://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption (accessed on 3 October 2022) [34].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Doumen, S.C.; Nguyen, P.; Kok, K. Challenges for large-scale Local Electricity Market implementation reviewed from the stakeholder perspective. *Renew. Sustain. Energy Rev.* **2002**, *165*, 112569. [CrossRef]
2. Alamaniotis, M.; Karagiannis, G. Elm-fuzzy method for automated decision-making in price directed electricity markets. In Proceedings of the 2019 16th International Conference on the European Energy Market (EEM), Ljubliana, Slovenia, 18–20 September 2019.
3. Alamaniotis, M.; Gatsis, N.; Tsoukalas, L.H. Virtual Budget: Integration of electricity load and price anticipation for load morphing in price-directed energy utilization. *Electric. Power Syst. Res.* **2018**, *158*, 284–296. [CrossRef]
4. Alamaniotis, M.; Ikonomopoulos, A.; Tsoukalas, L.H. Evolutionary multiobjective optimization of kernel-based very-short-term load forecasting. *IEEE Trans. Power Syst.* **2012**, *27*, 1477–1484. [CrossRef]
5. Sha, H.; Xu, P.; Lin, M.; Peng, C.; Dou, Q. Development of a multi-granularity energy forecasting toolkit for demand response baseline calculation. *Appl. Energy* **2021**, *289*, 116652. [CrossRef]
6. Rodrigues, F.; Cardeira, C.; Calado, J.M.F. The daily and hourly energy consumption and load forecasting using artificial neural network method: A case study using a set of 93 households in Portugal. *Energy Procedia* **2014**, *62*, 220–229. [CrossRef]
7. Vats, V.K.; Rai, S.; Bharti, D.; De, M. Very Short-Term9 Short-Term and Mid-Term Load Forecasting for Residential Academic Institute: A Case Study. In Proceedings of the 4th IEEE International Conference on Computing Communication and Automation (ICCCA), Greater Noida, India, 14–16 December 2018.
8. Jiang, Y.; Gao, T.; Dai, Y.; Si, R.; Hao, J.; Zhang, J.; Gao, D.W. Very short-term residential load forecasting based on deep-autoformer. *Appl. Energy* **2020**, *328*, 120120. [CrossRef]
9. Faraji, J.; Ketabi, A.; Hashemi-Dezaki, H.; Shafie-Khah, M.; Catalao, J.P. Optimal day-ahead scheduling and operation of the prosumer by considering corrective actions based on very short-term load forecasting. *IEEE Access* **2020**, *8*, 83561–83582. [CrossRef]
10. Marino, D.L.; Amarasinghe, K.; Manic, M. Building energy load forecasting using deep neural networks. In Proceedings of the IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 23–26 October 2016.
11. Ploysuwan, T. Deep CNN & LSTM network for appliances energy forecasting in residential houses using IoT sensors. In Proceedings of the 7th International Electrical Engineering Congress (iEECON), Chiang Mai, Thailand, 4–6 March 2019.
12. Ribeiro, A.M.N.C.; Rafael, P.; Carmo, X.; Endo, P.T.; Rosati, P.; Lynn, T. Short- and Very Short-Term Firm-Level Load Forecasting for Learning Models. *Energies* **2022**, *15*, 750. [CrossRef]
13. Bessani, M.; Massignan, J.A.D.; Santos, T.M.O.; London, J.B.A.; Maciel, C.D. Multiple households very short-term load forecasting using bayesian networks. *Electr. Power Syst. Res.* **2020**, *189*, 106733. [CrossRef]
14. Tong, C.; Zhang, L.; Li, H.; Ding, Y. Attention-based temporal–spatial convolutional network for ultra-short-term load forecasting. *Electr. Power Syst. Res.* **2023**, *220*, 109329. [CrossRef]
15. Munkhammar, J.; van der Meer, D.; Widén, J. Very short term load forecasting of residential electricity consumption using the Markov-chain mixture distribution (MCM) model. *Appl. Energy* **2021**, *282*, 116180. [CrossRef]

16. Shi, T.; Mei, F.; Lu, J.; Lu, J.; Pan, Y.; Zhou, C.; Wu, J.; Zheng, J. Phase Space Reconstruction Algorithm and Deep Learning-Based Very Short-Term Bus Load Forecasting. *Energies* **2019**, *12*, 4349. [CrossRef]
17. Lin, W.; Wu, D.; Boulet, B. Spatial-temporal residential short-term load forecasting via graph neural networks. *IEEE Trans. Smart Grid* **2021**, *12*, 5373–5384. [CrossRef]
18. Botman, L.; Soenen, J.; Theodorakos, K.; Yurtman, A.; Bekker, J.; Vanthournout, K.; Bart de Moor, H.B.; Lago, J. A scalable ensemble approach to forecast the electricity consumption of households. *IEEE Trans. Smart Grid* **2022**, *14*, 757–768. [CrossRef]
19. Ves, A.V.; Ghitescu, N.; Pop, C.; Antal, M.; Cioara, T.; Anghel, I.; Salomie, I. A stacking multi-learning ensemble model for predicting near real time energy consumption demand of residential buildings. In Proceedings of the IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 5–7 September 2019.
20. Atef, S.; Nakata, K.; Eltawil, A.B. A deep bi-directional long-short term memory neural network-based methodology to enhance short-term electricity load forecasting for residential applications. *Comput. Ind. Eng.* **2022**, *170*, 108364. [CrossRef]
21. Rajabi, R.; Estebsari, A. Deep learning based forecasting of individual residential loads using recurrence plots. In Proceedings of the IEEE Milan PowerTech, Milan, Italy, 23–27 June 2019.
22. Dinesh, C.; Makonin, S.; Bajić, I.V. Residential power forecasting using load identification and graph spectral clustering. *IEEE Trans. Circuits Syst. II Express Briefs* **2019**, *66*, 1900–1904. [CrossRef]
23. Zheng, J.; Chen, X.; Yu, K.; Gan, L.; Wang, Y.; Wang, K. Short-term power load forecasting of residential community based on GRU neural network. In Proceedings of the International Conference on Power System Technology (POWERCON), Guangzhou, China, 6–8 November 2018.
24. Kong, W.; Dong, Z.Y.; Jia, Y.; Hill, D.J.; Xu, Y.; Zhang, Y. Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Trans. Smart Grid* **2017**, *10*, 841–851. [CrossRef]
25. Din, G.M.U.; Mauthe, A.U.; Marnerides, A.K. Appliance-level short-term load forecasting using deep neural networks. In Proceedings of the International Conference on Computing, Networking and Communications (ICNC), Maui, HI, USA, 5–8 March 2018.
26. Oprea, S.V.; Bâra, A. Machine learning algorithms for short-term load forecast in residential buildings using smart meters, sensors and big data solutions. *IEEE Access* **2019**, *7*, 177874–177889. [CrossRef]
27. Gonzalez, R.; Ahmed, S.; Alamaniotis, M. Deep Neural Network Based Methodology for Very-Short-Term Residential Load Forecasting. In Proceedings of the 13th International Conference on Information, Intelligence, Systems & Applications (IISA), Corfu, Greece, 18–20 July 2022.
28. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
29. Tsoukalas, L.H.; Uhrig, R.E. *Fuzzy and Neural Approaches in Engineering*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1996.
30. Alamaniotis, M. Synergism of deep neural network and elm for smart very-short-term load forecasting. In Proceedings of the IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe), Bucharest, Romania, 29 September–2 October 2019.
31. Alamaniotis, M. Fuzzy Leaky Bucket System for Intelligent Management of Consumer Electricity Elastic Load in Smart Grids. *Front. Artif. Intell.* **2020**, *3*, 1–14. [CrossRef]
32. Dimitroulis, P.; Alamaniotis, M. Residential energy management system utilizing fuzzy based decision-making. In Proceedings of the IEEE Texas Power and Energy Conference (TPEC), College Station, TX, USA, 6–7 February 2020.
33. Dimitroulis, P.; Alamaniotis, M. Multimodal energy management system for residential building prosumers utilizing various lifestyles. *Electr. Power Syst. Res.* **2022**, *213*, 108732. [CrossRef]
34. Machine Learning Repository. Individual Household Electric Power Consumption Data Set. Available online: https://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption (accessed on 1 February 2023).
35. Alamaniotis, M.; Ikonomopoulos, A.; Tsoukalas, L.H. A Pareto optimization approach of a Gaussian process ensemble for short-term load forecasting. In Proceedings of the 16th International Conference on Intelligent System Applications to Power Systems, Hersonisos, Greece, 25–28 September 2011.
36. Nichiforov, C.; Martinez-Molina, A.; Alamaniotis, M. An Intelligent Big Data Analytics Method for Two-Dimensional Non-Residential Building Energy Forecasting. *Intell. Decis. Technol.* **2022**, *16*, 691–698. [CrossRef]
37. Ghenai, C.; Al-Mufti, O.A.A.; Al-Isawi, O.A.M.; Amirah, L.H.L.; Merabet, A. Short-term building electrical load forecasting using adaptive neuro-fuzzy inference system (ANFIS). *J. Build. Eng.* **2022**, *52*, 104323. [CrossRef]