energies

MDPI

Article

A Neural Network Forecasting Approach for the Smart Grid Demand Response Management Problem

Slim Belhaiza [1,*,†] and Sara Al-Abdallah [2,†]

[1] Department of Mathematics and IRC for Smart Logistics and Mobility, College of Computing & Mathematics, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia
[2] Department of Mathematics, College of Computing & Mathematics, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia; saraalabdallah94@gmail.com
* Correspondence: slimb@kfupm.edu.sa
† These authors contributed equally to this work.

Abstract: Demand response management (DRM) plays a crucial role in the prospective development of smart grids. The precise estimation of electricity demand for individual houses is vital for optimizing the operation and planning of the power system. Accurate forecasting of the required components holds significance as it can substantially impact the final cost, mitigate risks, and support informed decision-making. In this paper, a forecasting approach employing neural networks for smart grid demand-side management is proposed. The study explores various enhanced artificial neural network (ANN) architectures for forecasting smart grid consumption. The performance of the ANN approach in predicting energy demands is evaluated through a comparison with three statistical models: a time series model, an auto-regressive model, and a hybrid model. Experimental results demonstrate the ability of the proposed neural network framework to deliver accurate and reliable energy demand forecasts.

Keywords: forecasting; neural networks; smart grid

Citation: Belhaiza, S.; Al-Abdallah, S. A Neural Network Forecasting Approach for the Smart Grid Demand Response Management Problem. *Energies* **2024**, *17*, 2329. https://doi.org/10.3390/en17102329

Academic Editors: Marcin Sosnowski, Jaroslaw Krzywanski, Karolina Grabowska, Dorian Skrobek and Ghulam Moeen Uddin

Received: 29 March 2024
Revised: 4 May 2024
Accepted: 8 May 2024
Published: 11 May 2024

1. Introduction

Smart grids (Figure 1) play a crucial role in ensuring the safe, efficient, and reliable operation of systems, contributing to the reduction of power loss in the electricity network. However, modern smart grids face various economic and technical challenges as they strive to deliver energy securely and cost-effectively to consumers. Among the most significant obstacles are load-flow analysis, scheduling, and electric energy system control. Achieving optimal operation and planning of the power system requires an accurate prediction model.

Over the past decade, load forecasting has emerged as a rapidly developing field of interest and research within smart grids. Numerous forecasting techniques for power system load have been proposed, with mathematical models demonstrating success. These techniques aim to minimize estimation errors between predicted and measured future values in energy demands.

The importance of smart grids' demand forecasting is evident in several key aspects:

- Reducing unit production costs and preserving the efficiency of power facilities.
- Monitoring high-risk maintenance operations and managing energy reserves.
- Providing crucial data for planning and ensuring effective power delivery.

1.1. Literature Review on Forecasting Models for Load Forecasting

One of the most common forecasting models is categorization, which distinguishes between linear and non-linear models (Raza et al. [1]). Linear models are separated into statistic time series and dynamic time series models.

Figure 1. Smart grid as a network of intelligent devices.

Time series forecasting is the analysis of time series data using statistics and modeling to generate predictions. The forecast is generally based on data with a historical time stamp. Time series forecasting techniques can be categorized according to numerous parameters [2]. Categorization according to the forecasting horizon distinguishes between the time frame of the forecast :

- Very short-term load forecasting (VSTLF): from a few seconds to several hours. These models are commonly used in flow management.
- Short-term load forecasting (STLF): ranging from hours to weeks. These models are commonly used to balance supply and demand.
- Medium-term and long-term load forecasting (MTLF and LTLF): between months and years normally. These models are used to plan resource usage.

The magnitude of the variables employed is the key distinction between these three forecasting horizons, not taking into consideration the model that was used (Hernandez et al. [3]). Meanwhile, there are some limitations on time series forecasting. For instance, time series are not useful for all situations, as not all models can fit all sets of data. It is up to data teams and analysts to understand the limits of their analysis and what their models can support.

Dynamic models are such that factors and random inputs are taken into consideration dynamically. They make use of time series for modeling dynamical behavior. They are divided into auto-regressive and moving average models and state-space models.

- Auto-regressive and moving average (ARMA) models combine auto-regressive and moving average models. Auto-regressive models use the previous values to forecast future values. Moving average (MA) models calculate the residuals or errors of previous values and determine future values. In ARMA models, residuals and the effects of previous values are taken into account when predicting future values. Many modifications to the ARMA model can be found in the literature under other names like Auto-Regressive Integrated Moving Average (ARIMA), which is quite similar to the ARMA model in the use of previous values and residuals to predict future values, other than the fact that it includes one more factor known as Integrated (I).
- State-space models are used when dealing with dynamic time series issues. They use a set of input, output, and state variables to represent a physical system mathematically. The state variables are employed to describe a system with a set of first-order differential equations. State–space models are commonly used to analyze ecological and biological time-series data.

A comprehensive table outlining the findings of the review of 47 publications detailing 264 forecasting models from 1997 to 2018 is presented in Czapaj et al. [4]. It summarizes the status of research on short-term power demand forecasting for power systems using auto-regressive and non-auto-regressive approaches and models. In addition, the authors provide a new method for creating literature reviews when choosing the most probable forecasting models. An analysis was conducted on the effectiveness of the forecasting models as determined by the Mean Average Percentage Error (MAPE) measure. Table 1

lists the top 10 forecasting techniques and models: DEA, FR, GRM, GA, ANFIS, ANN, FGRM, WANN, ANN, and FL, where the repeated ANN belongs to different models [4]. FGRM and GRM employ the explanatory variables, while the remaining eight models are auto-regressive.

Table 1. Top 10 rank of the forecasting techniques and models.

Ranking	Authors	Year	Method, Model	MAPE
1	Kheirkhah et al. [5]	2013	DEA (Data Envelopment Analysis)	0.01%
2	Kheirkhah et al. [5]	2013	FR (Fuzzy Regression)	0.08%
3	Wang et al. [6]	2018	GRM (General Regression Model)	0.10%
4	Kheirkhah et al. [5]	2013	GA (Genetic Algorithm)	0.14%
5	Kheirkhah et al. [5]	2013	ANFIS (Adaptive Neuro Fuzzy Inference System)	0.15%
6	Kheirkhah et al. [5]	2013	ANN (Artificial Neural Network)	0.16%
7	Wang et al. [6]	2018	FGRM (Full General Regression Model)	0.20%
8	Rana et al. [7]	2016	WANN (Wavelet Artificial Neural Network)	0.27%
9	Rana et al. [7]	2016	ANN (Artificial Neural Network)	0.28%
10	Rana et al. [7]	2016	FL (Fuzzy Logic)	0.29%

The results of the reviews supported the auto-regressive approach's great potential for forecasting power demand. The employment of auto-regressive models may assist the transmission system operator in achieving improved forecasting efficiency. The advantage of using the ARIMA model is that it is simple to use and offers good forecasts over a short period of time. It has two basic limitations, which are listed below (Khashei et al. [8]):

- Linear limitation: it is supposed that a variable's future value will be a linear function of several previous data points and random errors. If the implicit mechanism is nonlinear, the ARIMA models' estimation may be completely unsuitable. However, since non-linearity is a common feature of real-world systems (Zhang et al. [9]), it is illogical to assume that a given implementation of a time series is the result of a linear process.
- Data limitation: for ARIMA models to produce the desired results, a lot of historical data are required. Data limitation dictates that ARIMA models need at least 50, and preferably 100 or more, data points to get the required results.

Using hybrid models or combining multiple models can be an efficient strategy to enhance forecasting performance. The fundamental concept behind model combining in forecasting is to remove its limits in order to create a more comprehensive model with more accurate outcomes. The fundamental ideas of ARIMA, ANNs, and fuzzy regression models are used to formulate a new approach to forecasting. In that model, the special ability of ANNs in nonlinear estimating is employed to go over the ARIMA models' linear limitations and create a more accurate model. Additionally, fuzzy logic is used to get beyond the ARIMA models' data limitations and provide a model that is more adaptable [8].

1.2. Literature Review on ANN Approaches for Load Forecasting

Recently, Dewangan et al. [10] provided an extensive examination of load forecasting, encompassing the categorization, performance indicator calculation, data analysis procedures, and utilization of conventional meter data for load forecasting, alongside the technologies employed and associated challenges. This study delved into the significance of smart meter-based load forecasting, exploring various approaches available.

In particular, the application of ANN-based machine learning to estimate electricity consumption dates back to the 1990s, with ongoing research. Park et al. [11] demonstrated the superiority of ANN-based techniques over traditional forecasting methods. Amjady et al. [12] used the ANN modified harmony search algorithm for short-term load forecasting (STLF) with excellent accuracy. Macedo et al. [13] explored DRM in power systems using ANN. Baliyan et al. [14] conducted a survey on ANN applications for short-term load forecasting. Muralitharan et al. [15] proposed a neural-network-based

optimization model for energy demand forecasting, employing genetic algorithms and particle swarm optimization. Li [16] utilized a machine learning-based forecasting model for short-term load forecasting, integrating data mining and de-noising methods. Jha et al. [17] employed the LSTM and random forest methodologies for electricity load forecasting. Through meticulous comparison with models employing analogous parameters, they ascertained the superior reliability and suitability of our model for long-term forecasting. Their model exhibits an exemplary performance, boasting an average overall accuracy of 96%. Sharadga et al. [18] compared various methods for predicting time series, including both statistical techniques and artificial intelligence-based approaches for forecasting phtovoltaic (PV) power output. Additionally, the study examines how altering the prediction time frame impacts the performance of these algorithms. The BI-LSTM algorithm proves to be a highly precise model for predicting power output in large-scale PV plants, surpassing various neural networks and statistical models in accuracy.

Da Silva et al. [19] proposed a solution employing a fuzzy-ARTMAP (FAM) artificial neural network (ANN). Historical databases are utilized to extract the fundamental knowledge necessary for training this ANN. In tandem with load forecasting, the FAM-ANN integrates a continuous learning (CL) mechanism, enabling incremental knowledge acquisition through real-time measurement system data. A rapid and highly precise load forecasting (with a mean absolute percentage error of around 2%) is achieved for extended forecast intervals, such as 96 h ahead. Tarmanini et al. [20] used two different forecasting models within machine learning (ML) techniques for load prediction: auto regressive integrated moving average (ARIMA) and artificial neural network (ANN). They evaluated the performance of both methods using mean absolute percentage error (MAPE). Utilizing daily electricity consumption data from 709 randomly selected households in Ireland over an 18-month duration, the study demonstrated that ANN outperforms ARIMA in handling non-linear load data.

1.3. Motivation and Novelties of Our Approach

The utilization of ML-based forecasting models continues to evolve, addressing various aspects of energy demand prediction with advancements in neural network approaches. In this context, this paper aims to identify the most accurate smart grid load forecasting models and neural network architectures.

The key challenges faced by artificial neural networks (ANN) involve obtaining precise results, achieving maximum performance during training, and minimizing overall prediction errors. The main innovation points in our current paper intend to address these challenges. To this aim, we explore three distinct models:

- Time series model, utilizing multiple input measurements (hour, period, day, season, month, number of appliances) to forecast energy consumption.
- Auto-regressive model, relying on past energy consumption (within a specific period range) to predict future energy consumption.
- Hybrid auto-regressive model, incorporating both input measurements and past energy consumption to forecast energy consumption.

The performance of these three models is assessed using various multi-layer neural network architectures. The proposed ANN framework undergoes testing on two real-life smart grid data sets to derive general recommendations.

Our paper introduces novelty through the comparison of three statistical models within various ANN architectures for smart grid load forecasting, a comparison that, to our knowledge, has not been conducted previously. The closest study to ours is by Tarmanini et al. [20], which compares the ARIMA model with an ANN model for short-term load forecasting. However, our work stands out notably as we integrate the time series model, the auto-regressive model, and the hybrid model within the ANN forecasting framework. To the best of our knowledge, this approach has not been published before. The chosen forecasting model and neural network are intended to contribute to energy

conservation, aid in demand and supply management, and facilitate efficient financial planning for users venturing into power generation.

Section 2 outlines the data formatting and pre-processing methods employed in our context. Additionally, it provides a detailed account of the statistical models and the artificial neural network (ANN) approach for DRM. Section 3 presents the experimental results obtained with the three different models, utilizing various ANN architectures on two smart grid data sets. Finally, Section 4 summarizes and concludes the paper.

2. ANN Forecasting Approach for DRM

Neural network (NN) techniques constitute a subset of machine learning methods employed for diverse prediction problems. NNs excel at modeling non-linear data across various domains and can approximate complex functions with reasonable precision. In particular, recurrent neural networks (RNNs) employ training data to acquire knowledge. Their defining feature is their ability to retain and leverage information from previous inputs to influence current input-output relationships [21]. RNNs' outputs are influenced by preceding elements within the sequence. In this paper, RNNs (Figure 2) are used to estimate the uncertain smart grid energy load y^t. To do so, we utilize observed historical values p_r^t, on $t = 1, \ldots, T$ previous observations of $r = 1, \ldots, R$, input variables. The estimated $\hat{y}^t$ of y^t is then employed as the average value approximation for the smart grid demand. The activation function $f_l(.)$ used in each layer l ($l = 1, \ldots, L$) of the ANN may be different. It approximates the output of each neuron in layer l. Each layer l may contain S_l neurons. A composed function $F(p^t)$ transforms the input $p^t = (p_1^t, \ldots, p_r^t, \ldots, p_R^t, 1)$ into a predicted $\hat{y}^t$, such that:

$$\hat{y}^t = F(p^t) = f_L(f_l(\ldots f_1(p^t))).$$

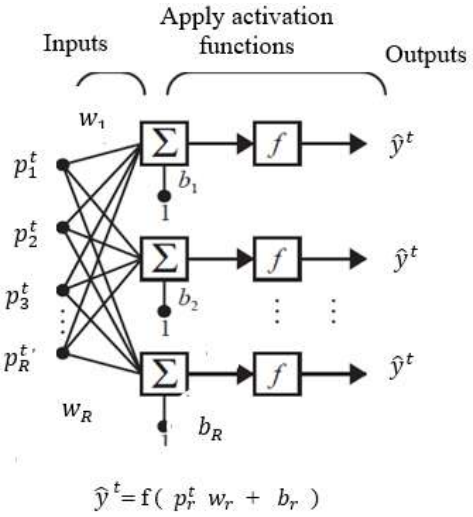

Figure 2. Neural Network Sample Architecture.

2.1. Steepest Descent Methods

In a traditional steepest descent scheme, weight updates occur after each forward pass h, adhering to the following sequence:

$$w_r^{h+1} = w_r^h - \mu \frac{1}{||\nabla E(T)||} \frac{\partial E(T)}{\partial w_r},$$

$$b^{h+1} = b^h - \mu \frac{1}{||\nabla E(T)||} \frac{\partial E(T)}{\partial b}.$$

Here, μ represents a predetermined parameter adjusted based on specific guidelines, while $||\nabla E(T)||$ indicates the magnitude of $\nabla E(T)$. In vector form, the steepest gradient descent scheme is written as:

$$w^{h+1} = w^h - \frac{\mu}{||\nabla E(T)||} \cdot \nabla E(T).$$

In the forthcoming Experimental Results Section 3, we employ a walk-forward optimization on the training set as we assess the effectiveness of our ANN utilizing various gradient optimizers, encompassing adaptive moment estimation (Adam) [22], adaptive gradient algorithm (Adagrad) [23], and Adamax [22] as a further development of Adam.

2.2. Data Sets, Data Formatting and Component Analysis

This paper investigates two real-life data sets: Toronto Data Set 1 and Data Set 2. Toronto Data Set 1 encompasses energy consumption information from 1082 households, collected daily over 36 consecutive months in the past 3 years: 2019, 2020, and 2021. The data include geographic coordinates of each household, its current index in the data set, the 'number of electric appliances' inventoried, the 'day order' within the current year, the 'season', the 'day of the week', and the 'period' of the 'measurement'. Each household contributes only one observation.

Toronto Data Set 2 comprises over 30,000 observations from 6 selected residential areas. The energy consumption information was gathered daily over 60 consecutive months spanning the past 5 years: 2017, 2018, 2019, 2020, and 2021. Similar to Data Set 1, it includes details such as the 'number of electric appliances' inventoried, the 'day order' within the current year, the 'season', the 'day of the week', the 'period', and the 'hour' of the 'measurement'.

The 'day order' represents the order of the measurement day in the corresponding year. Both data sets are segmented into three four-month seasons: Season 1 spans from November to February, Season 2 spans from March to June, and Season 3 spans from July to October. Days of the week are denoted by integers, with Monday assigned the value of 1, Tuesday assigned the value of 2, and so forth. Time periods are also represented by integers: 1 indicates the time between 0:00 and 5:59, 2 indicates the time between 6:00 and 11:59, 3 indicates the time between 12:00 and 17:59, and 4 indicates the time between 18:00 and 23:59. In Data Set 2, measurements are taken over six hours, ranging from 1 to 6, for each time period.

2.3. Data Formatting and Cleaning

The acquired data needed to undergo transformation into input-output time series samples for model training. Specifically, for data set 1, we had to address issues with 14 out of 1082 data entries. These 14 entries generally had a value of 0 in the (kilowatt per hour) KWH electricity consumption column, likely stemming from measurement errors. Subsequently, we replaced the entries in the corresponding cells with the average values observed for demand. This replacement took into consideration factors such as average demand on similar weekdays, seasons, and periods.

Data cleaning involves identifying and rectifying mistakes and discrepancies to enhance the quality of the data. Errors in data entry, information gaps, and other types of inaccuracies can lead to issues with data quality.

2.3.1. Correlation Analysis

The correlation matrix obtained for residential area 1, as shown in Table 2, indicates that the correlation between consumption (KWH) and the number of appliances is the highest. The correlation between consumption and the other features is lower than the correlation with the number of appliances. All the features have a similar positive effect on consumption, but their impact is smaller than the impact of the number of appliances.

Table 2. Correlation matrix: residential area 1.

Cov.	Day Order	Day of Week	Period	Nbr Appliances	Hour	KWH
Day Order	1.000	0.319	0.346	0.450	0.328	0.204
Day of Week	0.319	1.000	0.378	0.485	0.358	0.244
Period	0.346	0.378	1.000	0.524	0.383	0.243
Nbr Appliances	0.450	0.485	0.534	1.000	0.463	0.449
Hour	0.328	0.358	0.383	0.463	1.000	0.186
KWH	0.204	0.244	0.243	0.449	0.186	1.000

2.3.2. Trend and Seasonality

Trend and seasonality decomposition, whether additive or multiplicative, enables the identification of inherent data patterns specific to the problem at hand. An additive decomposition is applied when the variation around the trend cycle or the magnitude of seasonal fluctuations remains constant with the time series level. On the other hand, a multiplicative decomposition is preferred when such variance is found to be proportional to the time series level. The additive formula for time series decomposition is given by:

$$y_t = T_t + S_t + R_t.$$

Meanwhile, the multiplicative formula is expressed as:

$$y_t = T_t \times S_t \times R_t,$$

where y_t represents the observed series, T_t denotes the trend component, S_t stands for the seasonal component, and R_t represents the irregular component (residual) at period t.

For example, the additive seasonality decomposition shown in Figure 3 indicates a significant seasonal component for electricity consumption in data set 1. It also reveals a specific trend related to increased electricity consumption during the first season, corresponding to the harsh winter conditions in Canada. The Dickey-Fuller test, with a p-value of 0.07 (more than 0.05), conducted on electricity consumption confirms the presence of seasonality.

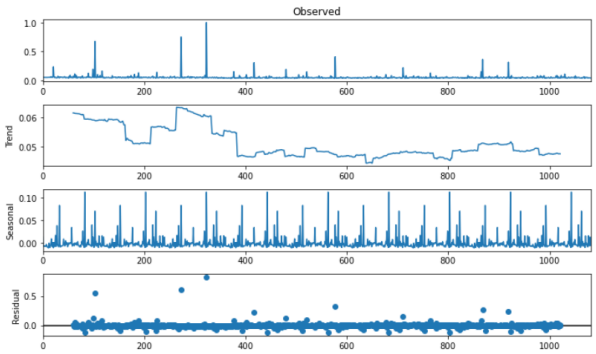

Figure 3. Additive Trend-Seasonality Decomposition Results.

3. ANN Forecasting Experimental Results

We evaluate the performance of our neural network (NN) approach in predicting energy demands by comparing it with three statistical models: a time series model, an autoregressive model, and a hybrid model. For this evaluation, we employ various network architectures, manipulating the number of neurons at each layer, using different activation function types, and employing diverse gradient descent methods, including Adam, Adagrad, or Adamax.

3.1. Time Series Model

Time series models serve as statistical tools for analyzing and predicting data collected over a specific period. Their applications span various fields, encompassing finance, economics, weather forecasting, and more. The main goal of time series modeling is to understand and characterize the inherent patterns and trends within the data, facilitating accurate predictions of future values.

The time series model undergoes testing on both data sets 1 and 2. In the first layer, we apply the activation function to the inputs of the data set. The output from the neurons in the first layer is subsequently used as input for the second layer, and this process continues until reaching the final layer, denoted as L. The computation is represented as follows:

$$\hat{y}^t = F(p^t) = f_L(\dots f_l(\dots f_1(p^t))).$$

3.2. Auto-Regressive Model

Auto-regressive models (AR) belong to the category of time series models that utilize the previous values of a variable to predict its future values. These models posit that the current value of the variable results from a combination of its past values and a random error term. The order of an AR model, denoted as 'r', indicates the number of preceding values used to forecast the present value. For example, an AR(1) model uses the previous value of the variable to predict the current value, while an AR(2) model incorporates the last two values. In this paper, our auto-regressive model is expressed as follows:

$$\hat{y}^t = F(y^{t-r}) = f_L(\dots f_l(\dots f_1(y^{t-r})))$$

Here, r denotes the number of preceding values of y utilized, representing the time delay in hours between the last measurement and the output forecasting.

For data set 1, where only a single observation is recorded for each household, this model cannot be applied. Consequently, the auto-regressive model is exclusively tested on data set 2, with a maximum delay of $r = 6$ h, corresponding to a complete period.

3.3. Hybrid Model

Hybrid models, also known as mixed models, are models that incorporate multiple types of models to formulate predictions. A hybrid time series model can integrate various models, such as combining an auto-regressive model with a moving average model or merging an auto-regressive integrated moving average (ARIMA) model with a neural network model. The purpose of a hybrid model is to leverage the strengths of different models and mitigate their respective limitations. For example, an auto-regressive model might perform well for short-term predictions but may fall short for long-term predictions. In such a case, a hybrid model that combines an auto-regressive model with a moving average model could yield superior results. Our hybrid model combines time series and auto-regressive models, considering:

$$y^t = F(p^t, y^{t-r}) = f_L(\dots f_l(\dots f_1(p^t, y^{t-r})))$$

Similar to the auto-regressive model, the hybrid model is exclusively tested on data set 2, with a maximum delay of $r = 6$ h, corresponding to a complete period.

3.4. Experimental Results

Our experiments were conducted using Python 3 on Jupyter Notebook (Jupyter Project). The computations were executed on 64-bit workstations equipped with 2.9 GHz Intel Core i7-7600 processors and 8 GB RAM, running under MS Windows 11.

The accuracy results of the three-layer ANN are summarized in the following tables: The results were obtained by varying the number of neurons in each layer (S_1, S_2, and S_3) and the activation functions f_1 and f_2. The entries in the table indicate the root mean

squared error (RMSE) of the residuals calculated with a training set size that remains constant at 3/4 of the entire data set while predicting household consumption.

Though the scientific community may not have definitively resolved the debate concerning the ideal size ratio between the training and validation sets, we opt to leave this question open for future exploration, as it does not align with the primary objectives of our current study.

RMSE is one of the most popular measures for assessing the accuracy of predictions. It represents the Euclidean distance between measured true values and forecasts. To compute RMSE, we first calculate the residual (the difference between prediction and true value) for each data point, then compute the norm of the residuals for each data point. Finally, we compute the mean of the residuals and take the square root of that mean using the formula:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{T}(y^t - \hat{y}^t)^2}{N}}.$$

All the results are obtained with a fixed number of epochs (50). An epoch refers to a single forward and backward pass through the entire training data set.

The results in Table 3 reveal that the last architecture, with f_1: relu and f_2: relu as activation functions, generally provides the best predictions and the lowest RMSE averages. In particular, the lowest RMSE (1.201) is obtained with $S_1 = 64$ neurons in the first layer, $S_2 = 32$ neurons in the second, and $S_3 = 16$ neurons in the third layer with the Adam optimizer. Adam is more stable than Adagrad and Adamax when S_1 varies from 32 to 256, S_2 varies from 16 to 128 neurons, and S_3 varies from 8 to 64 neurons across the four architectures. The third architecture, with f_1: tanh and f_2: log-sigmoid as activation functions, yields a larger RMSE on average. Its best RMSE (1.793) is slightly better than that obtained by the best accuracy provided by the first architecture. More detailed results are available upon request.

Figure 4 depicts the consumption values in the y-axis (blue) against the corresponding predicted values (green) using the proposed three-layer ANN on data set 1. The neural network is configured with f_1: relu, f_2: relu, and the Adam optimizer. The x-axis represents the index of observations.

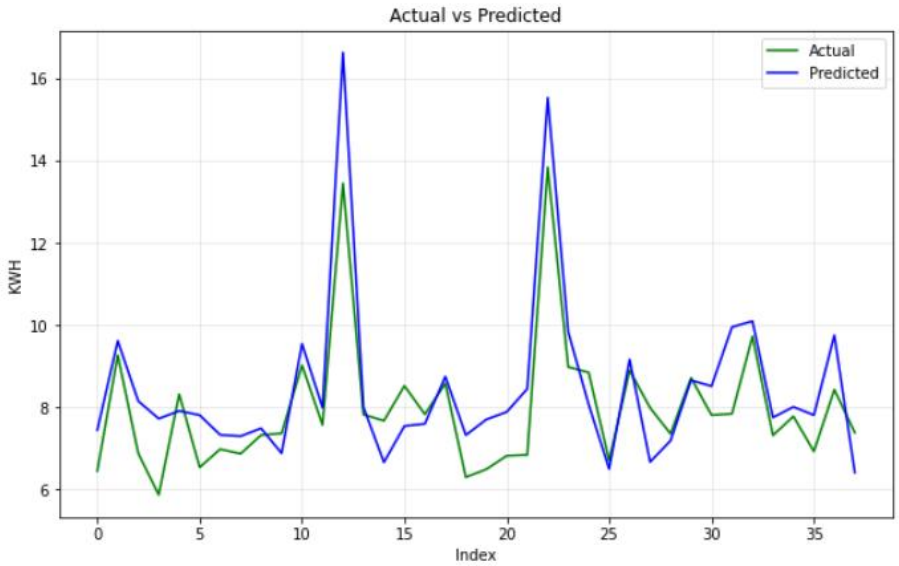

Figure 4. Sample Measurement vs. Prediction on Data Set 1.

Table 3. Three-Layer ANN Forecasting Results on Data Set 1.

			f_1: Log-Sigmoid; f_2: Selu; f_3: Relu			
S_1	S_2	S_3	Adam	Adagrad	Adamax	Avg.
32	16	8	**1.828**	7.984	7.194	5.669
64	32	16	2.004	7.244	6.967	5.405
128	64	32	2.426	7.214	5.636	5.092
256	128	64	2.987	7.04	4.315	**4.781**
		avg.	**2.311**	7.371	6.028	5.237
			f_1: Selu; f_2: Tanh; f_3: Relu			
S_1	S_2	S_3	Adam	Adagrad	Adamax	Avg.
32	16	8	2.907	7.297	6.099	5.434
64	32	16	2.372	7.162	4.500	4.678
128	64	32	3.480	6.889	3.276	4.548
256	128	64	4.002	5.982	**2.187**	**4.057**
		avg.	**3.190**	6.833	4.016	4.679
			f_1: Tanh; f_2: Log-Sigmoid; f_3: Relu			
S_1	S_2	S_3	Adam	Adagrad	Adamax	Avg.
32	16	8	3.060	7.217	7.149	5.809
64	32	16	3.628	7.200	7.160	5.996
128	64	32	**1.793**	7.199	7.031	5.341
256	128	64	2.177	7.182	6.917	**5.425**
		avg.	**2.665**	7.200	7.064	5.643
			f_1: Relu; f_2:Relu ; f_3: Relu			
S_1	S_2	S_3	Adam	Adagrad	Adamax	Avg.
32	16	8	1.817	7.560	6.501	5.293
64	32	16	**1.201**	7.375	5.830	4.802
128	64	32	2.275	7.125	4.313	4.571
256	128	64	1.307	6.465	2.736	**3.503**
		avg.	**1.650**	7.131	4.845	4.542

Table 4 showcases the results for residential area 1 in data set 2. In the time series model, the most effective architecture, featuring f_1: relu and f_2: relu as activation functions, achieves the lowest RMSE (0.977). This result is obtained with $S_1 = 256$ neurons in the first layer, $S_2 = 128$ neurons in the second layer, $S_3 = 64$ neurons in the third layer, and the Adamax optimizer.

For the auto-regressive model, utilizing f_1: relu and f_2: relu as activation functions, the minimum RMSE (0.807) is attained with $S_1 = 32$ neurons in the first layer, $S_2 = 16$ neurons in the second layer, and $S_3 = 8$ neurons in the third layer, with the Adam optimizer.

In the hybrid model, employing f_1: relu and f_2: relu as activation functions, the lowest RMSE (0.697) is achieved with $S_1 = 256$ neurons in the first layer, $S_2 = 128$ neurons in the second layer, $S_3 = 64$ neurons in the third layer, and the Adamax optimizer.

Across all three models and four architectures, both Adam and Adamax demonstrate greater stability than Adagrad. Notably, as S_1 varies from 32 to 256, S_2 ranges from 16 to 128 neurons, and S_3 varies from 8 to 64 neurons. Overall, the hybrid model consistently yields the lowest RMSE among the three models.

Table 4. Forecasting Results on Residential 1.

			f_1: Sigmoid; f_2: Selu ; f_3: Relu											
			Time-Series				**Hybrid**				**Auto-Regressive**			
S_1	S_2	S_3	Adam	Adagrad	Adamax	Avg.	Adam	Adagrad	Adamax	Avg.	Adam	Adagrad	Adamax	Avg.
32	16	8	1.076	1.629	1.211	1.305	0.766	0.988	0.812	0.855	0.843	1.041	0.894	0.926
64	32	16	**0.998**	1.56	1.071	**1.201**	0.736	0.902	0.784	**0.807**	0.840	0.939	0.890	0.890
128	64	32	1.027	1.54	1.043	1.203	**0.727**	0.871	0.787	0.795	0.890	0.891	0.904	0.895
256	128	64	1.03	1.538	1.038	1.202	0.775	0.863	0.756	0.798	**0.838**	0.894	0.878	**0.870**
		Avg.	**1.032**	1.567	1.091	1.228	**0.751**	0.906	0.785	0.814	**0.853**	0.941	0.892	0.895
			f_1: Tanh; f_2: Sigmoid ; f_3: Relu											
			Time Series				**Hybrid**				**Auto Regressive**			
S_1	S_2	S_3	Adam	Adagrad	Adamax	Avg.	Adam	Adagrad	Adamax	Avg.	Adam	Adagrad	Adamax	Avg.
32	16	8	1.002	1.568	1.031	1.2	0.745	0.994	0.787	0.842	0.848	1.029	0.850	0.909
64	32	16	1.007	1.484	1	1.164	0.710	0.907	0.731	0.783	0.855	0.911	0.882	0.883
128	64	32	1.005	1.432	0.993	**1.143**	**0.706**	0.870	0.727	**0.768**	**0.824**	0.901	0.840	**0.855**
256	128	64	**0.981**	1.555	0.997	1.178	0.809	0.851	0.760	0.807	0.862	0.900	0.895	0.886
		Avg.	**0.999**	1.510	1.005	1.171	**0.743**	0.906	0.751	0.800	**0.847**	0.935	0.867	0.883
			f_1: Tanh; f_2: Selu ; f_3: Relu											
			Time Series				**Hybrid**				**Auto-Regressive**			
S_1	S_2	S_3	Adam	Adagrad	Adamax	Avg.	Adam	Adagrad	Adamax	Avg.	Adam	Adagrad	Adamax	Avg.
32	16	8	0.998	2.221	1.035	1.418	0.764	0.842	0.749	0.785	0.850	0.876	0.833	0.853
64	32	16	0.985	1.335	1.001	1.107	0.749	0.825	0.752	0.775	0.825	0.870	0.821	0.839
128	64	32	**0.981**	1.286	0.99	**1.086**	0.734	0.820	**0.705**	**0.753**	**0.813**	0.862	0.819	**0.831**
256	128	64	1.004	1.226	1.006	1.079	0.733	0.812	0.728	0.758	0.837	0.864	0.877	0.859
		Avg.	**0.992**	1.517	1.008	1.173	0.745	0.825	**0.734**	0.768	**0.831**	0.868	0.838	0.846
			f_1: Relu; f_2: Relu ; f_3: Relu											
			Time Series				**Hybrid**				**Auto-Regressive**			
S_1	S_2	S_3	Adam	Adagrad	Adamax	Avg.	Adam	Adagrad	Adamax	Avg.	Adam	Adagrad	Adamax	Avg.
32	16	8	0.981	1.381	1.007	1.123	0.701	0.847	0.759	0.769	**0.807**	0.891	0.843	0.847
64	32	16	0.994	1.346	1.003	1.114	0.727	0.816	0.729	0.757	0.833	0.863	0.840	0.845
128	64	32	0.985	1.263	0.978	1.075	0.702	0.805	0.731	0.746	0.811	0.858	0.833	**0.834**
256	128	64	0.986	1.158	**0.977**	**1.04**	0.699	0.779	**0.697**	**0.725**	0.839	0.865	0.839	0.848
		Avg.	**0.987**	1.287	0.991	1.088	**0.707**	0.812	0.729	0.749	**0.823**	0.869	0.839	0.844

For residential area 4, the results presented in Table 5 indicate that the last architecture, featuring f_1: relu and f_2: relu as activation functions, generally offers the best predictions with the lowest average RMSE across all three models.

In the time series model, the lowest RMSE achieved by the three-layer ANN is 1.640, obtained with $S_1 = 32$ neurons in the first layer, $S_2 = 16$ neurons in the second, and $S_3 = 8$ neurons in the third layer, utilizing the Adam optimizer.

In the auto-regressive model, the lowest RMSE, amounting to 0.158, is attained with $S_1 = 128$ neurons in the first layer, $S_2 = 64$ neurons in the second, and $S_3 = 32$ neurons in the third layer, employing the Adamax optimizer.

Concerning the hybrid model, the lowest RMSE, totaling 1.064, is achieved with $S_1 = 256$ neurons in the first layer, $S_2 = 128$ neurons in the second, and $S_3 = 64$ neurons in the third layer, utilizing the Adamax optimizer.

Across all four architectures and three models, both Adam and Adamax exhibit greater stability compared to Adagrad, as S_1 varies from 32 to 256, S_2 ranges from 16 to 128 neurons, and S_3 varies from 8 to 64 neurons. Overall, the hybrid model consistently yields the lowest RMSE among the three models.

Table 5. Forecasting Results on Residential 4.

			f_1: Sigmoid; f_2: Selu ; f_3: Relu											
			Time Series				**Hybrid**				**Auto-Regressive**			
S_1	S_2	S_3	Adam	Adagrad	Adamax	Avg.	Adam	Adagrad	Adamax	Avg.	Adam	Adagrad	Adamax	Avg.
32	16	8	1.695	3.213	2.120	2.343	1.113	1.625	1.167	1.302	1.206	1.665	1.228	1.366
64	32	16	1.692	3.042	2.033	2.256	1.080	1.459	1.139	1.226	1.207	1.470	1.184	1.287
128	64	32	1.676	3.037	1.724	2.146	**1.076**	1.330	1.133	1.180	1.193	1.395	1.203	1.264
256	128	64	**1.657**	3.034	1.708	**2.133**	1.090	1.284	1.112	**1.162**	**1.183**	1.290	1.194	**1.222**
		Avg.	**1.680**	3.082	1.896	2.219	**1.090**	1.425	1.138	1.218	**1.197**	1.455	1.202	1.285
			f_1: Tanh; f_2: Sigmoid ; f_3: Relu											
			Time Series				**Hybrid**				**Auto-Regressive**			
S_1	S_2	S_3	Adam	Adagrad	Adamax	Avg.	Adam	Adagrad	Adamax	Avg.	Adam	Adagrad	Adamax	Avg.
32	16	8	1.667	2.978	1.679	2.108	1.084	1.613	1.138	1.278	1.186	1.606	1.181	1.324
64	32	16	1.677	3.043	1.697	2.139	1.080	1.331	1.102	1.171	**1.171**	1.317	1.186	1.225
128	64	32	1.686	3.003	1.665	2.118	1.074	1.263	1.090	1.142	1.205	1.270	1.175	**1.217**
256	128	64	1.696	2.597	**1.651**	**1.981**	**1.072**	1.246	1.073	**1.113**	1.197	1.260	1.211	1.223
		Avg.	1.682	2.905	**1.673**	2.087	**1.078**	1.363	1.101	1.180	1.190	1.363	**1.188**	1.241
			f_1: Tanh; f_2: Selu ; f_3: Relu											
			Time Series				**Hybrid**				**Auto-Regressive**			
S_1	S_2	S_3	Adam	Adagrad	Adamax	Avg.	Adam	Adagrad	Adamax	Avg.	Adam	Adagrad	Adamax	Avg.
32	16	8	1.675	2.919	1.740	2.111	1.078	1.339	1.127	1.181	1.186	1.286	1.209	1.227
64	32	16	1.696	2.625	1.673	1.998	1.075	1.256	1.087	1.139	**1.162**	1.235	1.183	1.193
128	64	32	1.665	2.498	1.674	1.946	1.080	1.229	1.084	1.131	1.193	1.201	1.179	**1.191**
256	128	64	1.662	2.388	**1.653**	**1.901**	1.106	1.183	**1.067**	**1.119**	1.182	1.213	1.187	1.194
		Avg.	**1.675**	2.608	1.685	1.989	**1.085**	1.252	1.091	1.142	**1.181**	1.234	1.190	1.201
			f_1: Relu; f_2: Relu ; f_3: Relu											
			Time Series				**Hybrid**				**Auto-Regressive**			
S_1	S_2	S_3	Adam	Adagrad	Adamax	Avg.	Adam	Adagrad	Adamax	Avg.	Adam	Adagrad	Adamax	Avg.
32	16	8	**1.640**	2.866	1.680	2.062	1.090	1.376	1.086	**1.084**	1.169	1.251	1.177	1.199
64	32	16	1.694	2.599	1.663	1.986	1.095	1.259	1.095	1.150	1.161	1.253	1.178	1.197
128	64	32	1.666	2.408	1.652	1.909	1.080	1.183	1.080	1.114	1.168	1.215	**1.158**	**1.180**
256	128	64	1.643	2.089	1.642	**1.791**	1.089	1.149	**1.064**	1.101	1.186	1.203	1.160	1.183
		Avg.	1.661	2.491	**1.659**	1.937	1.089	1.242	**1.081**	1.137	1.171	1.231	**1.168**	1.190

3.5. Summary of Results

For both data sets 1 and 2, the optimal accuracy with our three-layer ANN is consistently achieved using the last architecture with f_1: relu and f_2: relu as activation functions. Specifically, for data set 1, this is typically realized with the Adam optimizer, while for data set 2, the preference is generally for the Adamax optimizer.

In the case of data set 2, within the time series model, the Adamax optimizer yields the most accurate results for residential areas 1, 2, 3, 5, and 6. Residential area 4, on the other hand, achieves the best accuracy with the Adam optimizer. In the auto-regressive model, the Adamax optimizer is superior in accuracy for residential areas 2, 3, 4, 5, and 6, while residential area 1 achieves the best accuracy with the Adam optimizer. For the hybrid model, the Adamax optimizer consistently provides the best accuracy across all residential areas. Notably, the hybrid model exhibits the highest accuracy (lowest RMSE value) among the three models.

Stochastic Gradient Descent (SGD) proves ineffective with both data sets 1 and 2. The underlying concept of SGD relies on random subsets of the gradient, but due to sparse gradients and numerous zeros in each gradient subset, it fails to perform effectively.

4. Conclusions

This paper proposed a neural network forecasting approach for demand-side management in smart grids. The presented neural network framework demonstrated the ability to accurately and reliably predict energy consumption, as evidenced by experimental results on two data sets. The paper explored diverse three-layer neural network architectures and three different statistical models.

The experimental results showed that the most effective architecture employs the relu activation function in all three layers. In addition, the Adamax optimizer consistently yielded the highest accuracy for the hybrid model across all residential areas. Notably, among the three models, the hybrid model demonstrated the greatest accuracy.

Regarding computational efficiency, the time series model emerges as the fastest among the three models, albeit with a compromise in accuracy. Conversely, the hybrid model requires more computational time but delivers superior accuracy.

The selected ANN, coupled with the most suitable statistical model chosen by energy providers, has the potential to contribute to energy conservation, demand and supply management, and the effective organization of financial support for individuals initiating power production.

Author Contributions: Methodology, S.B.; Validation, S.B.; Formal analysis, S.A.-A.; Investigation, S.A.-A.; Resources, S.B.; Data curation, S.A.-A.; Writing—original draft, S.A.-A.; Writing—review & editing, S.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

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

References

1. Raza , M.Q.; Khosravi, A. A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. *Renew. Sustain. Energy Rev.* **2015**, *50*, 1352–1372. [CrossRef]
2. Hippert, H.S.; Pedreira, C.E.; Souza, C.R. Neural Networks for Short-Term Load Forecasting: A review and Evaluation. *IEEE Trans. Power Syst.* **2001**, *16*, 44–51. [CrossRef]
3. Hernandez, L.; Baladron, C.; Aguiar, J.; Belen, C.; Sanchez-Esguevillas, A.J.; Lloret, J.; Massana, J. A Survey on Electric Power Demand Forecasting: Future Trends in Smart Grids, Microgrids and Smart Buildings, A Survey on Electric Power Demand Forecasting: Future Trends in Smart Grids, Microgrids and Smart Buildings. *IEEE Commun. Surv. Tutorials* **2014**, *16*, 1460–1495. [CrossRef]
4. Czapaj, J.S.M.; Kaminski, R. A review of auto-regressive methods applications to short-term demand forecasting in power systems. *Energies* **2022**, *15*, 6729. [CrossRef]
5. Kheirkhah, A.; Azadeh, A.; Saberi, M.; Azaron, A.; Shakouri, H. Improved estimation of electricity demand function by using of artificial neural network, principal component analysis and data envelopment analysis. *Comput. Ind. Eng.* **2013**, *64*, 425–441. [CrossRef]
6. Wang, Y.; Bielicki, J.M. Acclimation and the response of hourly electricity loads to meteorological variables. *Energy* **2018**, *142*, 473–485. [CrossRef]
7. Rana, M.; Koprinska, I. Forecasting electricity load with advanced wavelet neural networks. *Neurocomputing* **2016**, *182*, 118–132. [CrossRef]
8. Khashei, M.; Bijari, M.; Ardali, G.A.R. Improvement of auto- regressive integrated moving average models using fuzzy logic and artificial neural networks (anns). *Neurocomputing* **2009**, *72*, 956–967. [CrossRef]
9. Zhang, G.; Patuwo, B.E.; Hu, M.Y. Forecasting with artificial neural networks: The state of the art. *Int. J. Forecast.* **1998**, *14*, 35–62. [CrossRef]
10. Dewangan, F.; Abdelaziz, A.Y.; Biswal, M. Load Forecasting Models in Smart Grid Using Smart Meter Information: A Review. *Energies* **2023**, *16*, 1404. [CrossRef]

11. Park, D.C.; El-Sharkawi, M.A.; Marks, R.J.I.I.; Atlas, L.E.; Damborg M.J. Electric load forecasting using an artificial neural network. *IEEE Trans. Power Eng.* **1991**, *6*, 442–449. [CrossRef]
12. Amjady, N.; Keynia, F. A New Neural Network Approach to Short Term Load Forecasting of Electrical Power Systems. *Energies* **2011**, *4*, 488–503. [CrossRef]
13. Macedo, M.N.Q.; Galo, J.J.M.; de Almeida, L.A.L.; de CLima, A.C. Demand side management using artificial neural networks in a smart grid environment. *Renew. Sustain. Energy Rev.* **2015**, *41*, 128–133. [CrossRef]
14. Baliyan, A.; Gaurav, K.; Mishra, K.S. A Review of Short Term Load Forecasting using Artificial Neural Network Models. *Procedia Comput. Sci.* **2015**, *48*, 121–125. [CrossRef]
15. Muralitharan, K.; Sakthivel, R.; Vishnuvarthan, R. Neural network based optimization approach for energy demand prediction in smart grid. *Neurocomputing* **2018**, *273*, 199–208. [CrossRef]
16. Li, C. Designing a short-term load forecasting model in the urban smart grid system. *Appl. Energy* **2020**, *266*, 114850. [CrossRef]
17. Jha, N.; Prashar, D.; Rashid, M.; Gupta, S.K.; Saket, R.K. Electricity load forecasting and feature extraction in smart grid using neural networks. *Comput. Electr. Eng.* **2021**, *96*, 107479. [CrossRef]
18. Sharadga, H.; Hajimirza, S.; Balog, R.S. Time series forecasting of solar power generation for large-scale photovoltaic plants. *Renew. Energy* **2020**, *150*, 797–807. [CrossRef]
19. da Silva, M.A.; Abreu, T.; Santos-Junior, C.R.; Minussi, C.R. Load forecasting for smart grid based on continuous-learning neural network. *Electr. Power Syst. Res.* **2021**, *201*, 107545. [CrossRef]
20. Tarmanini, C.; Sarma, N.; Gezegin, C.; Ozgonenel, O. Short term load forecasting based on ARIMA and ANN approaches. *Energy Rep.* **2023**, *9*, 550–557. [CrossRef]
21. Hopfield, J.J. Neural Networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* **1982**, *79*, 2554–2558. [CrossRef] [PubMed]
22. Kingma, D.P.; Ba, J.L. Adam: A method for Stochastic Optimization. In Proceedings of the ICLR, San Diego, CA, USA, 7–9 May 2015.
23. Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.