Short-Term Electricity Load Forecasting using Deep Neural Networks

P. Rodrigues * **I. Serro** ** **P. Cavalheiro** ***

* *Faculdade de Ciências e Tecnologia, Universidade do Algarve (e-mail: a36569@ualg.pt)*
** *Faculdade de Ciências e Tecnologia, Universidade do Algarve (e-mail: a64536@ualg.pt)*
*** *Faculdade de Ciências e Tecnologia, Universidade do Algarve (e-mail: a67999@ualg.pt)*

Abstract: This report presents the development of a deep-learning model for short-term electricity load forecasting. The model is trained using a large dataset containing 180 days of electricity load data. A hyperparameter search is conducted to optimize the performance of the model. The results are compared to the literature, showing competitive performance. The model can be employed for a wide range of applications, including grid management and demand-side management.

Keywords: Deep learning, electricity load forecasting, time series prediction, hyperparameter optimization

1. INTRODUCTION

The uncertainty of energy supply and demand is a reality that can bring unwanted side effects to grid stability, such as overvoltage and frequency deviations. There are many factors that contribute to such disbalance, e.g., the aging of conventional power plants, technological advances and cost reduction which are motivating energy transition towards cleaner sources. From the consumer perspective, environmental awareness also plays an important role as more consumers are also becoming producers of energy (prosumers). On the other hand, the electrification of several sectors such as transport and heating is increasing the load demand (Machado et al., 2021).

In the face of such a scenario, accurately predicting electricity load demand (ELD) is crucial for both short-term load allocation and long-term planning of new generation and transmission infrastructures. Such predictions enable better decision-making in terms of cost and energy efficiency (Yoo and Myriam, 2018). Researchers find the field of energy demand forecasting highly appealing and have developed numerous models for this purpose. Commonly utilized techniques in energy demand forecasting studies include Box-Jenkins models, regression models, econometric models and neural networks (Oğcu et al., 2012). Regarding the latter, several types of artificial neural networks (ANN) have been employed to tackle load forecasting, yielding interesting outcomes. Deep neural networks are particularly advantageous for this task, especially as the volume of data increases. Their inherent ability to automatically learn patterns and extract features in the data with multiple input variables makes them a well-suited method to address this challenge (Vanting et al., 2021). Nonetheless, models are usually chosen based on complexity and performance measures (Ferreira et al., 2009).

This report covers a possible modeling strategy of the electricity load demand using a multi-layer perceptron model (MLP), to forecast the subsequent period of the given dataset. In addition, it also addresses a comparison against the strategy and results obtained by Ferreira et al. (2009), where a radial basis function (RBF) neural network was employed, in which the models were trained by the Levenberg-Marquardt algorithm and the number of neurons and input terms evolved using a Multi-Objective Genetic Algorithm (MOGA).

2. EXPERIMENTAL AND COMPUTATIONAL ANALYSIS

2.1 Data processing

The original data set consists of 180 days of electricity consumption in Portugal, where the values were measured in 15-minute intervals, as depicted in the time-series chart of Figure 1.

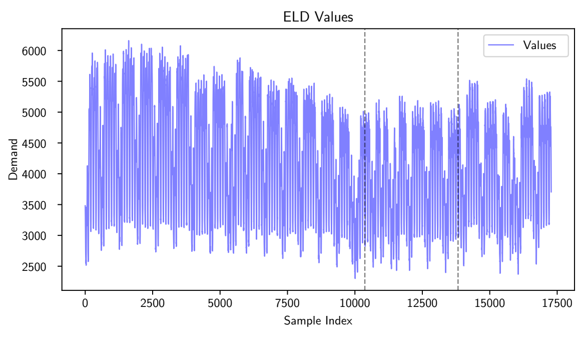

Fig. 1. Electricity consumption measured every 15 minutes during the 180 days with days 60 and 120 shown as dashed lines

From this dataset, we opted for a 60%/20%/20% split of days, i.e., 1-108, 108-144, and 144-180. We split the data into training, validation, and test sets, as shown by the vertical lines in Figure 1. As the name suggests, the training set was used specifically to train the neural network. The validation set allowed us to select the best model and prevent overfitting, and finally, the test set's goal was to evaluate the final model's performance.

We processed the data by normalizing the values between 0 and 1, using the following formula:

$$x_{norm} = \frac{x - \mu}{\sigma} \tag{1}$$

where x is the original value, μ is the mean of the dataset, and σ is the standard deviation of the dataset. This normalization was applied to the entire dataset, i.e., the training, validation, and test sets.

2.2 Model architecture

As already mentioned, the energy supply and demand are dynamic. Many events can affect the consumption of electricity thus disturbing the daily and weekly patterns observed. When selecting a model it is important to understand the focus of the forecasting that we are trying to achieve. For example, if the goal was the ELD forecasting of specific areas perhaps one could consider having the age of the population, temperature, humidity or even the number of households as input variables for the neural network. In our case, we aim for forecasting in a one-step-ahead fashion, i.e., predicting the value of a variable at the next time step, given its current state and previous observations, of the entire country's ELD, relying solely on the historical data of the electricity consumption in a well delimited period and without any particular region focus.

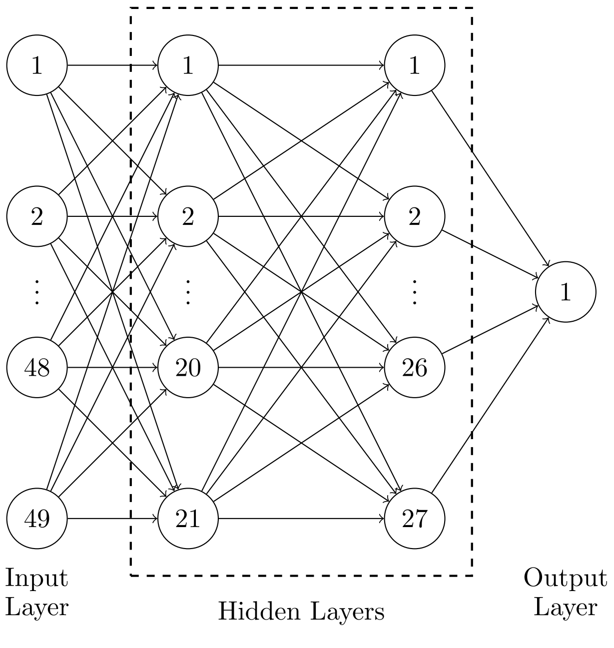

Fig. 2. Simplified neural network representation with 49 input neurons, 21 neurons in the second layer, 27 neurons in the third layer, and 1 output neuron.

A considerable range of models can be employed yielding different levels of accuracy, depending on the type and focus of the forecasting. Due to their generally good performance and simple nature, the Multi-layer perceptron model (MLP) will be used.

Multi-layer perceptrons are a type of artificial neural network that operates in a feed-forward manner. An MLP is made up of a set of sensory units (source nodes or source neurons) that form the input layer, one or more hidden layers consisting of computation nodes and an output layer also consisting of computation nodes. The neurons in each layer of the network are connected to all the neurons in the previous layer through weights (see Figure 2), which are parameters used to adjust the strength of the connections between neurons. The input signal is propagated through the network in a forward direction, moving from the leftmost input layer to the rightmost output layer on a layer-by-layer basis.

We use a single neuron for the output layer (since we only have 1 output value). For the input layer, we use 49 neurons, 27 with the values of the latest 27 readings (going back 405 minutes or 6h45m), 17 with the values of the readings around 7 days before the prediction time (1 neuron exactly 7 days before, 8 before that and 8 after that) and finally 5 with the values of the readings around 14 days before the prediction time (1 neuron exactly 14 days before, 2 before that and 2 after that). The inputs around 7 days before the prediction time exist to add information about past trends, while the inputs around 14 days before the prediction time were added to try to mitigate the effect of outlier days (e.g. holidays), which without further context would mean that the model would "echo" the holiday's trend to the next week.

The number of input nodes, hidden layers and sizes of the hidden layers were decided by random searching, from the following ranges:

- Latest readings: 1 to 128;
- Readings around 7 days before: 1 to 16;
- Readings around 14 days before: 1 to 4;
- Hidden layers: 1 to 3;
- Hidden layer size: 1 to 170.

The best model parameters were then selected for training.

2.3 Training algorithms

The goal of training an artificial neural network is to adjust the internal weights of the network to minimize a predefined error measure which is given by an error function (Christiansen et al., 2014). The training algorithm used was the Adaptive Moment Estimation (Adam), which uses momentum and adaptive learning rates to speed up convergence, and it is computationally efficient and has a low memory footprint (Jin et al., 2021). Regarding the error function, we selected one of the most common measures for regression which is the mean squared error (MSE). All of these were included in PyTorch (Paszke et al., 2019), which was used to implement the model.

Like the parameters for the model, the training parameters were selected by random searching. The number of epochs and patience were selected by manual observation, being

100 and 10 respectively. The following parameters were selected:

- Learning rate: 0.0795;
- Batch size: 32.

The model was trained for 72 hours on 12 parallel tasks, always keeping the best model (based on the validation loss) and stopping the epoch if the validation loss did not improve for 10 epochs. The training was done on a machine with a Ryzen 5 1600 CPU and 32 GB of RAM, dedicated to this task.

The final validation loss arrived upon was 0.17534 (normalized).

3. RESULTS AND DISCUSSION

The model was tested on the test set, which was not used for training or validation. The results are shown in Figure 3 below.

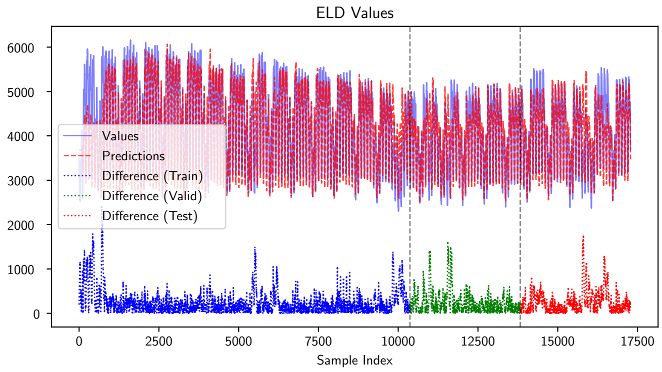

Fig. 3. Results of the model on all the test sets (enlarged version in Appendix A)

The model was trained properly and with success, showing the following average MSE for each set:

Set	Average MSE
Training	147.1488
Validation	134.8012
Test	144.3827

Table 1. Average MSE for each set

Ferreira et al. (2009), as discussed before, proposed more complex models with more information and a larger dataset. As expected, the results are better than ours.

Our approach also focuses on a 24-hour prediction horizon, while the aforementioned paper explores both 24-hour and 48-hour horizons.

The model was able to predict the general trend of the load, even if it was not able to predict the exact values. We also experimented with using the model trained for 24h to predict the load for 48h and the results were still good, even if not as good as the 24h prediction.

As expected, our model did not perform well with horizons not multiples of 24h, since the model was trained to predict the load for the same hour of the next day. It also couldn't handle holidays well, and although the 14 days before inputs were added to try to mitigate this, it still had a ripple effect on the following days.

4. CONCLUSIONS

While we expected Ferreira et al. (2009)'s results to be better, our results are still very promising. The error was smaller than expected, and the model was able to predict the general trend of the load, even if it was not able to predict the exact values. This is a good result, since the model can be used to predict the general trend of the load, which is the most important aspect of the prediction.

A bigger dataset (spanning multiple years or adding other related parameters, like holidays, day of the year or temperature) would have been beneficial since it would allow the model to learn more about the yearly trends of the load.

REFERENCES

Christiansen, N.F., Voie, P.E., Winther, O., and Høgsberg, J.B. (2014). Comparison of neural network error measures for simulation of slender marine structures. *Journal of Applied Mathematics*, 2014, 1–11. doi: 10.1155/2014/759834.

Ferreira, P.L., Ruano, A.E., Pestana, R., and Kóczy, L.T. (2009). Evolving rbf predictive models to forecast the portuguese electricity consumption. *IFAC Proceedings Volumes*, 42(19), 414–419. doi:10.3182/20090921-3-tr-3005.00073.

Jin, X., Zheng, W., Kong, J., Wang, X., Bai, Y., Su, T., and Lin, S. (2021). Deep-learning forecasting method for electric power load via attention-based encoder-decoder with bayesian optimization. *Energies*, 14(6), 1596. doi: 10.3390/en14061596.

Machado, E.C., Pinto, T., Guedes, V., and Morais, H. (2021). Electrical load demand forecasting using feed-forward neural networks. *Energies*, 14(22), 7644. doi: 10.3390/en14227644.

Oğcu, G., Demirel, O.B., and Zaim, S. (2012). Forecasting electricity consumption with neural networks and support vector regression. *Procedia - Social and Behavioral Sciences*, 58, 1576–1585. doi: 10.1016/j.sbspro.2012.09.1144.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J.T., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. volume 32, 8026–8037. URL `https://arxiv.org/pdf/1912.01703.pdf`.

Vanting, N.B., Ma, Z., and Jørgensen, B.B. (2021). A scoping review of deep neural networks for electric load forecasting. *Energy Informatics*, 4(S2). doi:10.1186/s42162-021-00148-6.

Yoo, S.G. and Myriam, H. (2018). Predicting residential electricity consumption using neural networks: A case study. *Journal of Physics*, 1072, 012005. doi: 10.1088/1742-6596/1072/1/012005.

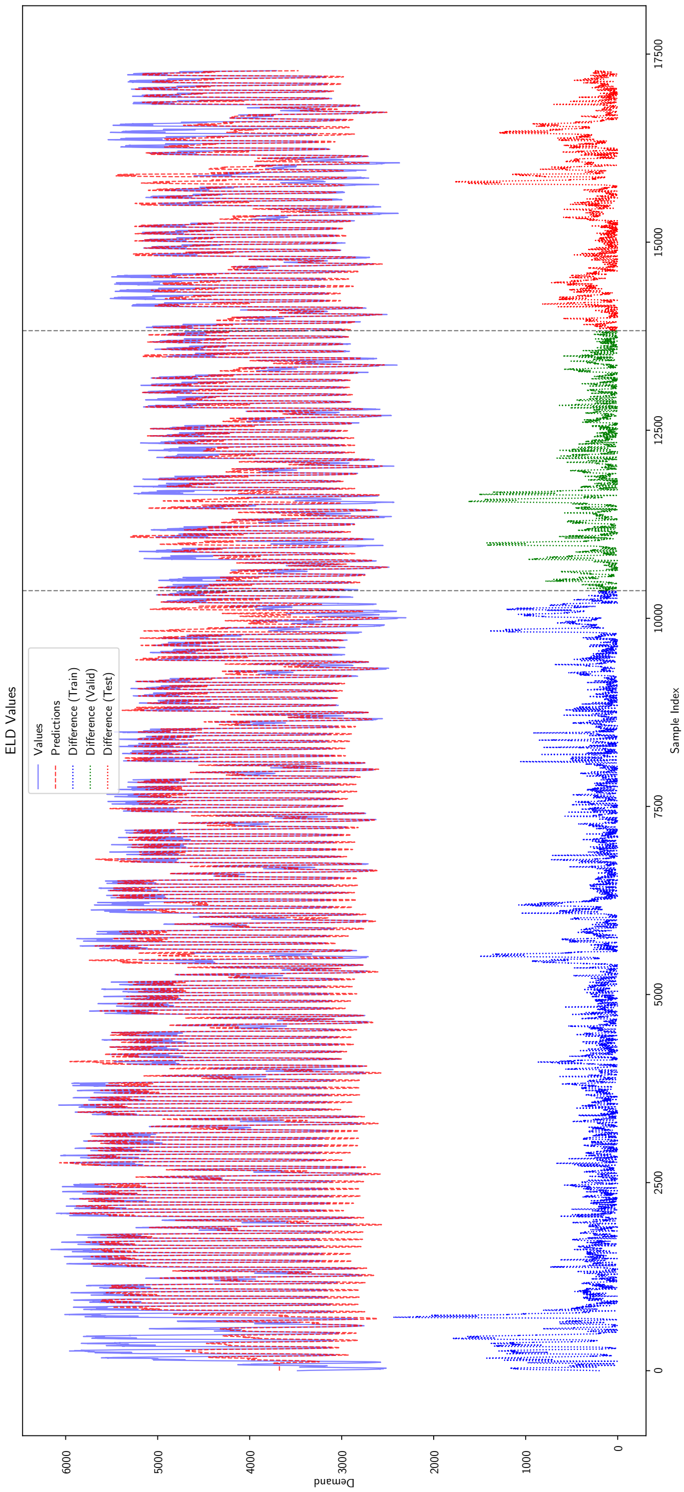

Fig. A.1: Results of the model on all the test sets