The 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
22-25 September, 2021, Cracow, Poland

Forecasting Day-Ahead Electricity Price with Artificial Neural Networks: a Comparison of Architectures

Milutin Pavićević[1], Tomo Popović[2]
[1,2] Faculty of Information Systems and Technologies,University of Donja Gorica,
Oktoih 1, 81000 Podgorica, Montenegro, [1]milutin.pavicevic@udg.edu.me, [2]tomo.popovic@udg.edu.me
https://udg.edu.me

***Abstract*—The spot price prediction for the electric energy markets is a widely approached problem, used by many participants in the market. The ever-shifting rules and regulations, rising percentage of the electricity on the market being produced by solar and wind plants and many stochastic factors influencing it make the market price of electricity very volatile and hard to forecast. Many methods are used to tackle this problem, and their efficiency varies from dataset to dataset. In this work, we use the dataset of hourly day-ahead spot prices from the Hungarian HUPX market, and couple it with weather data for Hungary. We test various types of Dense, Recurrent and Convolutional neural network architectures and report on the results.**

Keywords—Artificial neural networks, day-ahead electricity price, forecasting, convolutional neural networks, recurrent neural networks

I. Introduction

From the mid-2000s, there is a noticeable change in the way electricity is produced in Europe. Feed-in tariffs, subsidies for the producers of renewable energy, and the introduction of tariffs for CO2 emissions have led to the rise of installed capacity of solar and wind power sources and increased the share of the renewable sources in the complete energy traded on the market.

As these sources depend on natural factors (time of year, insolation, wind speed) more than the facilities that use the accumulated material and turn it into electricity when needed, their production is more stochastic [1], which may increase price volatility.

There are many models used to predict electricity spot price today. Weron and Ziel [2] had compared 58 different models and methods on many datasets and concluded that no single method is giving consistently best results, and that the performance depends on many factors, including the dataset itself.

Artificial neural networks (ANN) have been efficiently used in various fields, including time-series prediction. The rise of LSTM (Long Short-Term Memory) RNN (recurrent neural network) architectures in 2009 [3] made them a go-to choice for this task. Lately, temporal convolutional neural networks (CNNs) have proven to be even more precise on a subset of tasks.

This work tests various neural network architectures on the task of forecasting hourly day-ahead electricity prices on the HUPX market and reports on the results, hopefully giving an overview for those starting to work on prediction models on more complex datasets.

II. Dataset

A dataset of hourly prices (in EUR) for the HUPX day-ahead market was used. In the interest of better data windowing, two non-24 hours for every year (due to daylight savings time the last Sunday of October lasts 25, and the last Sunday of March lasts 23 hours) were reported as 24 hours by the HUPX: the price for the missing hour was reported as zero, and the price for the doubled hour was removed from the dataset. It consisted of a total of 82823 hourly values, representing 3451 days between July 10, 2010, and December 31, 2019.

A land area averaged dataset of hourly weather data for the Hungary was provided by the NinjaWeather service and coupled with the market data. As HUPX does not cover only Hungarian market and the weather factors were not the only one influencing the price, this brief dataset was far from complete (and should be expanded for use in final products) but provided a field for comparison of architectures.

TABLE I. HUPX DATASET DESCRIPTIVE STATISTICS

	Mean	STD	Min	50%	Max
Price (EUR)	46.408	20.453	-113.67	44.59	300.1
Precipitation (mm)	0.0726	0.1792	0	0.006	4.03
Temperature (C)	11.528	10.0697	-16.777	11.426	39.128
Snowfall (cm)	0.0065	0.0388	0	0	1.5869
Snow mass (cm)	0.9443	2.6921	0	0	25.074
Cloud cover %	0.5228	0.3223	0	0.541	0.9988
Air density (kg/m^3)	1.2176	0.045	1.1108	1.2137	1.3662

The dataset was expanded with derived data – month, day of month, and weekday values were separately added. As the aim was to compare the general efficiency, and not to maximize the precision by further expert input, no data on holidays or similar expectation markers (such as

This research is supported in part by the EuroCC project, grant agreement grant agreement 951732 EuroCC-H2020-JTI-EuroHPC-2019-2.

negative price or power facility maintenance expectation) were added.

Furthermore, a separate test was conducted on a dataset expanded with naive seasonal data calculation – every hourly row was expanded with a column showing the difference between the same hour on the same date, and the same hour a day after one year ago. This dataset was somewhat shorter, containing values from January 1st, 2011, with EPEX data being used for calculations between January 1, and July 10, 2010 – as the German market was used in place of HUPX prior to its formation. These values have introduced serious overfitting issues to the training process, and their usefulness was defined by how well the model can be optimized to avoid overfitting.

To prepare the dataset for ANN prediction, data frame had been divided into three sets – training, validation, and testing, chronologically (from oldest to most recent) in proportion of 70, 20, and 10, respectively. Training set contained the data used for creating the model, validation set was used for evaluation during training, and test set had been set aside, to test and evaluate models once they were created. As training and validation datasets were used in the training process, we report results only on test dataset.

In addition, the categorical information (day, week and month columns) was turned into one-hot arrays. The rest of the columns were standardized (normalized) into Z-values (standard score). This way of data preprocessing helps the prediction[4] and optimizes training[5].

III. Software and Data Windowing

The code was written in Python, with Keras and TensorFlow libraries used to build machine learning models, Pandas for data manipulation and Numpy for array and matrices support.

ANN models and data windows were built following the methods described in J. Brownlee's book "Deep Learning for Time Series Forecasting" [6], adapted and expanded for the task at hand. However, the language used to describe some concepts such as "multi-layer perceptrons" differs, in accordance to the advice given by M. Nielsen in the book "Neural Networks and Deep Learning" [7].

The task given to the ANN models can be described as follows:

Given the total of 336 hourly day-ahead prices on the HUPX market for 14 consecutive days, forecast the 24 day-ahead hourly prices for the next day.

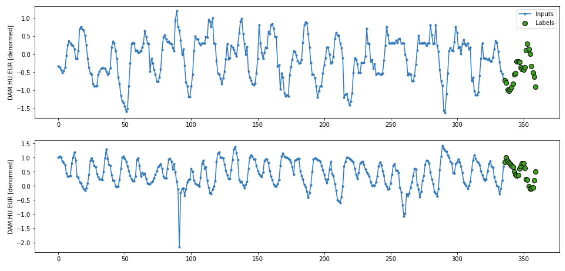

Figure 1. Example data windows

IV. ANN Architectures Tested

Several ANN architectures described below were tested multiple times, with changes and adaptations in-between, to prevent overfitting and increase precision. Three types of layers were the primary focus: densely connected layers (as a tratitional neural network architecture), recurrent LSTM layer (as a go-to standard for time-series prediction), and a temporal CNN (as an emerging approach to time-series).

A. Densely Connected Layers (Dense)

This is a traditional neural network with a hidden layer consisting of one or more columns of fully connected "neurons", with weights and biases determining the configuration. Rectified Linear Unit (ReLU) was used as an activation function, as it has shown to give a fast convergence, and no visible improvement for this dataset could be found by switching to tanh or leaky ReLU.

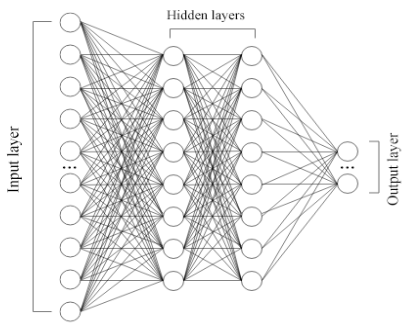

Figure 2. Neural network consisting of densely connected layers

A shallow, single hidden layer network with many neurons (512-1024) has shown to work the best for the initial dataset. The dataset with additional seasoned calculations had overfitting issues that had to be mitigated and has shown some gain from decreasing the number of neurons and adding additional hidden layers. For that dataset, a structure of three 64-neuron layers with two 20% dropout layers in between has given the best results.

B. Convolutional Neural Network (CNN)

Even though their usage is widely connected to image processing, the use of CNNs for time-series related tasks has surged lately [8]. 1D CNNs, sometimes referred to as "Temporal CNNs" or TCNs [9] have shown to efficiently use the feature-extracting ability of CNNs and apply it to time series.

It is worth noting that, due to the electricity demand being driven by societal habits such as work hours and production schedules, the exact hour of prediction is very important. This makes the usual stacking of convolutional and max-pooling layers used in CNNs less efficient. Instead, 24-hour convolutional filters with 24-hour strides have given the best results while testing on this dataset, both as a standalone CNN and as a part of combined networks, where CNN is used for feature extraction in combination with other layers. 128-256 filters and a single CNN layer provided the best results in this testing.

C. Long Short-Term Memory (LSTM)

Instead of feed-forward models, RNNs such as Elman and Jordan were often chosen for time-series prediction, including electricity price forecasting [10]. LSTM cell architecture was introduced to RNNs in order to mitigate the "vanishing gradient" and "exploding gradient" problems, giving recurrent networks more useful forgetting [11] capabilities.

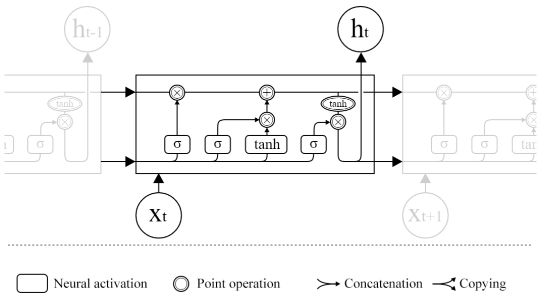

Figure 3. An LSTM cell

A differing number of LSTM cells (8-32) has performed the best for various tasks in these tests, with less cells performing better on the expanded dataset, due to larger LSTMs overfitting more quickly [6].

Gated recurrent units (GRU), another popular RNN gating mechanism similar to the LSTM was not tested in this process but might be of interest to those wishing to further explore the subject, as they have proven to give better results on some datasets [12].

D. Combined Layers and Autoregressive Approach

A few combined-layer structures were used in this test, with two showing promising results: an LSTM stacked between two dense layers (reported as *dense_LSTM_dense*), and a CNN layer followed by a dense layer (reported as *Conv_dense*).

In addition, an *autoregressive LSTM* (reported as AR LSTM) was tested, using a sequence of 24 next-hour predictions while feeding the prediction as a value in next steps.

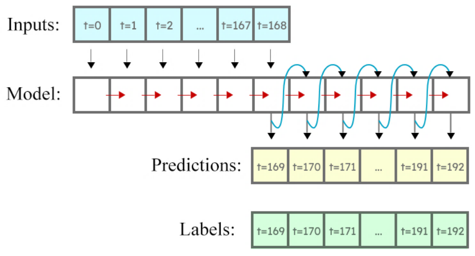

Figure 4. Diagram of an Autoregressive LSTM shown on the problem of predicting 24 hours of labels based upon one week (168 hours) of hourly values.

V. Objective Function, Evaluation and Comparison

Objective function (loss) used in neural network training defines how the neural network treats the errors and estimates their severity. Due to the differences in datasets, and even between time periods within the same dataset, a common way of reporting the accuracy is mean absolute percentage error (MAPE) [13][14]– the mean value, in percent, of an absolute value of forecast error divided by the actual value.

$$MAPE(w,b) = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{y(x_i) - a_i}{y(x_i)} \right| \% \qquad (1)$$

MAPE, however, shows weaknesses with datasets like the one at hand, where labels with zero or close-to-zero values are present [15]. Even with measures put in place to minimize exploding loss when predicting close to zero, when used as an objective (loss) function, MAPE, by its nature encourages the model to underestimate when predicting. The loss is also proportionately reduced when estimating large values – meaning that the punishment for large errors is the smallest when the electricity is the most expensive.

Instead, *mean squared error (MSE) was used as the objective function* during the training of these models. It is the most used regression loss function [16], with a smooth gradient fit for machine learning, eliminating the need for learning rate scheduling.

$$MSE(w,b) = \frac{1}{n} \sum_{i=1}^{n} (y(x_i) - a_i)^2 \qquad (2)$$

In addition to MSE and to better understand and compare the models, we measure and report the following metrics:

- *MAPE measured in EUR*, described as above, with denominators between -1 and 1 being replaced by 1, to increase numerical stability.
- *MAE* (mean average error) measured in *standard deviations and EUR.*
- *MAE represented as a percentage of the mean value (MAE%)* – to give another description of the model precision.

A. Baseline Naive Predictions

To set the baseline for comparison, two naive models were taken into consideration: one that repeats the value of the last (336th) hour 24 times as a prediction (reported as "last"), and one that repeats the last day (24 values) as a prediction (reported as "repeat").

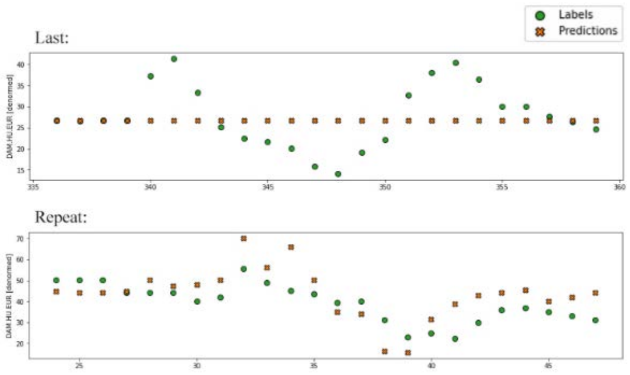

Figure 5. Naive prediction models

B. *Trained Linear Regressor*

As a third comparison point, a linear regression model has been trained, to provide a near-optimal linear combination of the 336 values.

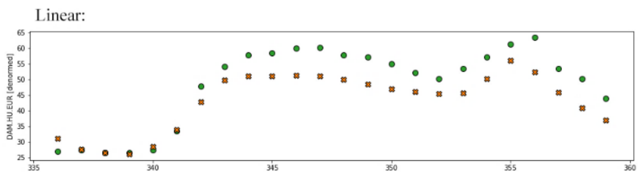

Figure 6. Linear prediction model

VI. Results and Discussion

Two tables below list the performance of all the models, with relevant metrics (as described earlier) in each of the columns. Shaded values mark the smallest error in every metric column.

TABLE II. Model performance on base test dataset

Dataset : Base				**STD**	**Mean**
				20.45392	**46.40822**
Model	MSE (std)	MAE (EUR)	MAE (std)	MAPE	MAE%
Last	1.9603	20.39052	0.9969	72.4896	43.93729
Repeat	1.6676	18.03013	0.8815	50.0480	38.85116
Linear	0.2688	7.74385	0.3786	21.2923	16.68639
Dense	0.2199	6.96251	0.3404	20.1414	15.00276
CNN	0.2186	7.10978	0.3476	20.6052	15.32010
Conv_dense	**0.2115**	6.83161	0.3340	**19.450**	14.72069
LSTM	0.2390	7.06683	0.3455	19.9769	15.22754
Den. LSTM_ den.	0.2177	**6.70070**	**0.3276**	19.5300	**14.43862**
AR LSTM	0.2423	7.21818	0.3529	21.2600	15.55369

TABLE III. Model performance on expanded test dataset

Dataset : Expanded with last year's label difference				**STD**	**Mean**
				20.69935	**46.35115**
Model	MSE (std)	MAE (EUR)	MAE (std)	MAPE	MAE%
Last	2.0287	21.32033	1.0300	72.1765	45.45203
Repeat	1.5803	17.70208	0.8552	51.9993	37.73842
Linear	0.2711	7.83677	0.3786	21.6264	16.70693
Dense	0.2257	7.12264	0.3441	20.1768	15.18450
CNN	0.2252	7.26133	0.3508	21.9086	15.48016
Conv_dense	**0.1894**	**6.53685**	**0.3158**	**19.5733**	**13.93568**
LSTM	0.2127	7.01707	0.3390	19.9300	14.95945
Den. LSTM den.	0.2273	7.25512	0.3505	20.6781	15.46692
AR LSTM	0.2484	7.65668	0.3699	22.4493	16.32301

Following charts show sample prediction of each of the ANN models. All the models' predictions are show on the base dataset, except for *Conv_dense*, shown on the expanded dataset.

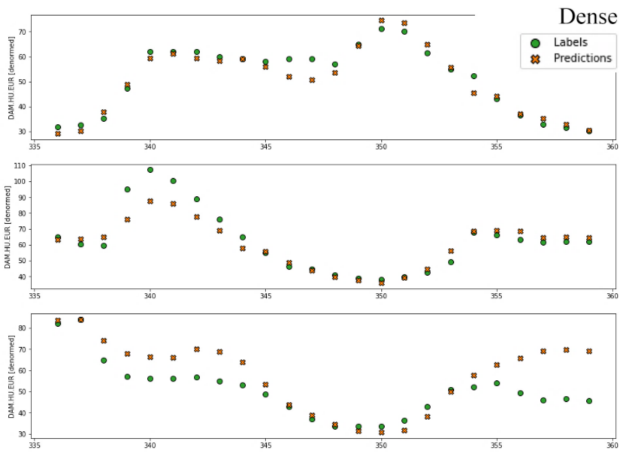

Figure 7. Predictions of the *"Dense"* model on the base dataset

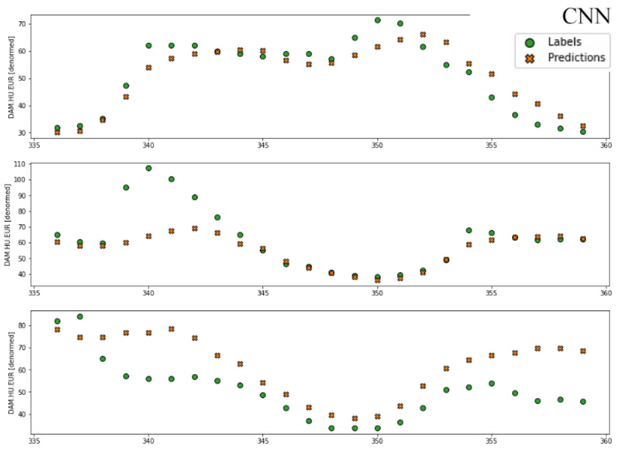

Figure 8. Predictions of the *"CNN"* model on the base dataset

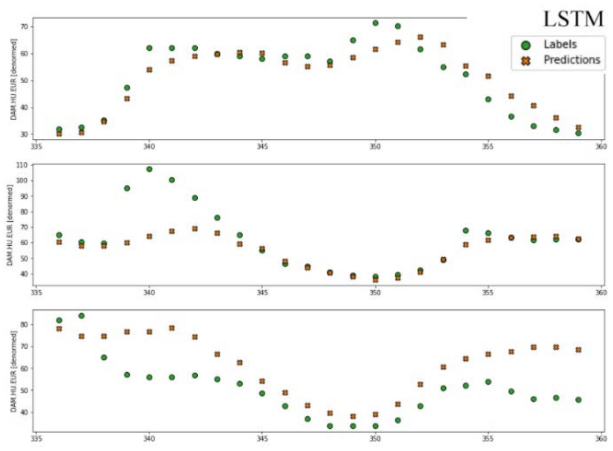

Figure 9. Predictions of the *"LSTM"* model on the base dataset

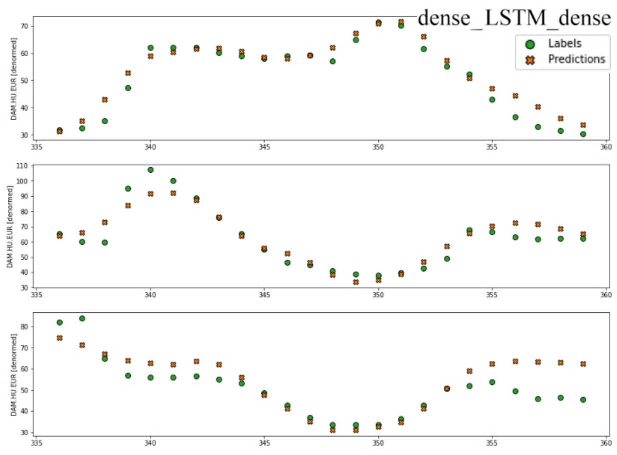

Figure 10. Predictions of the *"Dense_LSTM_dense"* model on the base dataset

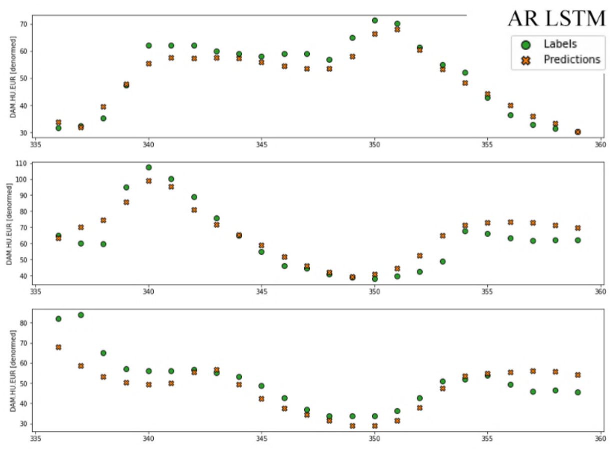

Figure 11. Predictions of the *"Autoregressive LSTM"* model on the base dataset

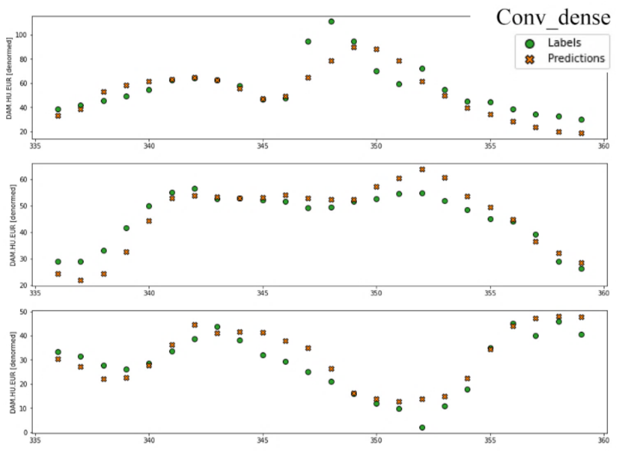

Figure 12. Predictions of the *"Conv_dense"* model on the expanded dataset

When comparing the results, one should have in mind that the expanded and base datasets are not the same, and have their own means and standard deviations, which influences MAE(std) and MSE(std) metrics.

The results show the temporal convolutional networks performing exceptionally well, especially when convolutional layers' feature extraction is combined with a dense layer. Recurrent neural networks with LSTM layers are matching the performance closely.

The models took 8-24 epochs to train. Expanding the dataset with derived data introduced overfitting issues, driving the training processes to sub-optimal local equilibria in the early epochs. This had to be combatted by changing the network structure, simplifying it, dropping dataset columns where possible, or adding drop layers. CNNs were more resistant to overfitting, ultimately extracting most value from the added data, with the *Conv_dense* on the expanded dataset being the best performing model overall. The 24-hour filters with 24-hour strides make the CNNs somewhat reminiscent of a "similar day" method popular among non-ANN prediction [17].

Autoregressive LSTMs have performed very well on most predictions, especially in the later hours of the prediction window, but their overall score was hindered by the tendency to create bigger errors in outlying cases where all models performed poorly, but not as bad. On the expanded dataset that would give context to the outliers (e.g., holiday data, maintenance schedule...) AR RNN models might perform significantly better.

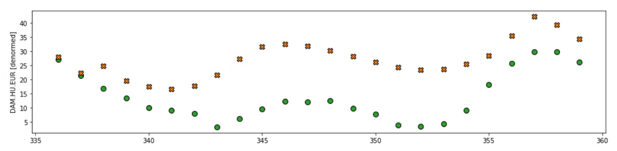

Figure 13. Failed predictions of the *"Autoregressive LSTM"* model

Traditional ANNs with fully connected layers of neurons (*Dense*) have given competitive results. With ease of their implementation, relatively fast training on today's hardware and a variety of ways to easily tweak, test and optimize them, they remain an important tool for creating and training quick and robust prediction models.

These tests were conducted as a comparative benchmark of various neural network models on the dataset and should be considered as starting points for developing practically usable models. Even with that in mind, the results are promising, and comparable to the tools in use on the nearby markets [10].

Expanding the dataset with more variables should be the next step toward achieving increased accuracy. A more inclusive weather data should be provided, to better cover the entirety of HUPX exchange geographics, preferably expanded with the wind data. In addition, but not limited to, the following factors used by prediction models and experts should be considered:

- Production and consumption figures [17]
- Holiday one-hot dummy or holiday type [17] [12]
- Near-holiday markers [17]
- Separately estimated supply and demand [18]
- Weather forecast data [12]
- Gas price [15][13]
- Oil, uranium, and coal price [13]
- Dew point, wet bulb, and dry bulb temperature [19]

Gated recurrent units (GRU) should be tested against LSTM cells as they have shown to be more efficient for the Turkish day-ahead market dataset [12].

Testing the effect of choosing Huber loss [14] for the objective function instead of MSE, as well as introducing Nesterov momentum [20] to gradient descent might provide interesting outcomes.

Re-training the entire model on the full dataset (training plus validation plus test) might lead to more efficient forecasting, as the most recent period usually contains the most useful information to predict the future. This is contrary to the machine learning best practices, but might be essential for some time-series, especially ones where the market rules change often. Alternatively, a roll-forward partitioning model might be considered [21].

VII. Conclusions

This paper describes Forecasting of Day-Ahead spot electricity price on the HUPX market, and compares various ANN architectures on the same dataset and task. Various layer architectures, combined architectures and objective functions were tested, with LSTM and temporal CNN showing the most promising results. The 24-hour filters with 24-hour approach in the CNN-dense combined architecture delivers good performance and might prove interesting for other time-series prediction models where "similar day" approach shows traditionally good results.

Future work includes testing the architectures on more advanced datasets, testing the performance of GRU cells and testing similar models on the problem of load prediction.

Acknowledgment

Many thanks to Nebojša Ćorović, Jovan Pavićević, Pavle Grbović and Vlado Kilibarda for generous help in obtaining datasets, and understanding the problem of day-ahead spot price prediction and the electricity market.

References

[1] R. Čabarkapa, D. Komatina, G. Ćirović, D. Petrović, and M. Vulić, *Analysis of Day-ahead Electricity Price Fluctuations in the Regional Market and the Perspectives of the Pumped Storage Hydro Power Plant "Bistrica."* Belgrade: JP Elektroprivreda Srbije.

[2] F. Ziel and R. Weron, "Day-ahead electricity price forecasting with high-dimensional structures: Univariate vs. multivariate modeling frameworks," *Energy Econ.*, vol. 70, pp. 396–420, 2018.

[3] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, *A Novel Connectionist System for Unconstrained Handwriting Recognition*. IEEE, 2009.

[4] C. Guo and F. Berkhahn, "Entity Embeddings of Categorical Variables," Apr. 2016, Accessed: Jul. 25, 2021. [Online]. Available: https://arxiv.org/abs/1604.06737v1.

[5] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao, "Normalization Techniques in Training DNNs: Methodology, Analysis and Application," *Methodol. Anal. Appl.*, vol. 1.

[6] J. Brownlee, *Deep Learning for Time Series Forecasting - Predict the Future with MLPs, CNNs and LSTMs in Python*, 1.3. 2018.

[7] M. Nielsen., *Neural Networks and Deep Learning*. USA: Determination Press, 2015.

[8] S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," Mar. 2018, Accessed: Jul. 20, 2021. [Online]. Available: http://arxiv.org/abs/1803.01271.

[9] B. Tan, "Farewell RNNs, Welcome TCNs. How Temporal Convolutional Networks are… | Aug, 2020 | Towards Data Science." https://towardsdatascience.com/farewell-rnns-welcome-tcns-dd76674707c8 (accessed Jul. 20, 2021).

[10] R. Beigaitė and T. Krilavičius, "Electricity price forecasting for Nord Pool data," 2017, pp. 37–42.

[11] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to Forget: Continual Prediction with LSTM.," *Neural Comput.*, vol. 12, pp. 2451–2471, 2000.

[12] U. Ugurlu and I. Oksuz, *Electricity Price Forecasting Using Recurrent Neural Networks*. MDPI, 2018.

[13] C. Bouley, *Recurrent Neural Networks for Electricity Price Prediction, Time-Series Analysis with Exogenous Variables*. VTT Technical Research Centre of Finland, Tekniikantie 4A, Espoo, Finland, 2020.

[14] I. Klešić, *Prognoza cijene električne energije*. Osijek: Sveučilište Josipa Jurja Strossmayera u Osijeku, 2018.

[15] A. Wagner and S. Schnürch, "Electricity Price Forecasting with Neural Networks on EPEX Order Books," *Appl. Math. Financ.*, vol. 27, pp. 189–206, 2020.

[16] P. Grover, "5 Regression Loss Functions All Machine Learners Should Know," *Heartbeat*, Jun. 2018. https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0 (accessed Jul. 20, 2021).

[17] R. Weron, "Electricity price forecasting: A review of the state-of-the-art with a look into the future," *Int. J. Forecast.*, 2014.

[18] F. Ziel and R. Steinert, "Electricity price forecasting using sale and purchase curves: The X-Model," *Energy Econ.*, vol. 59, pp. 435–454, 2016.

[19] S. M. R. Nateghi, "A Data-Driven Approach to Assessing Supply Inadequacy Risks Due to Climate-Induced Shifts in Electricity Demand," *Risk Anal.*, vol. 39, pp. 673–694, 2019.

[20] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," 2013, vol. 28, pp. 1139–1147.

[21] C. Cochrane, "Time Series Nested Cross-Validation ," *Towards Data Science*, May 19, 2018. https://towardsdatascience.com/time-series-nested-cross-validation-76adba623eb9 (accessed Jul. 20, 2021).