

# Application of Machine Learning Techniques for Forecasting Electricity Generation and Consumption in Ukraine

Anatoliy Doroshenko <sup>1,2</sup>, Dmitry Zhora <sup>1</sup>, Olena Savchuk <sup>2</sup>, and Olena Yatsenko <sup>1</sup>

<sup>1</sup> Institute of Software Systems of the National Academy of Sciences of Ukraine, Glushkov Ave. 40, build. 5, Kyiv, 03187, Ukraine

<sup>2</sup> National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Peremohy Ave. 37, Kyiv, 03056, Ukraine

## Abstract

To achieve successful business operations and optimize business processes, it is necessary to incorporate the request for forecasting as a key component in electricity markets. In the paper, based on the model developed at the Institute for Modeling in Energy Engineering of the National Academy of Sciences, the utilization of machine learning methods to predict the generation and consumption of electricity in Ukraine is proposed. The experiments were carried out involving the use of regression techniques to reconstruct a dataset from bilateral contracts to predict electricity generation and consumption. The results of employing machine learning algorithms on hourly peacetime data revealed the feasibility of accurately forecasting market volumes and thus electricity consumption one hour ahead, extending the predictive capability beyond a one-day planning horizon.

## Keywords <sup>1</sup>

Algorithm, balancing market, bilateral contracts market, day-ahead market, electrical energy, forecasting, intraday market, machine learning, regression.

## 1. Introduction

The issue of security within the energy sector holds significant importance for Ukraine which suffers from Russian aggressive actions. This directly affects the information security, information technology, and telecommunications industries. An electrical network is an area where the state of the network can be monitored, various characteristics can be measured, and the flow of electricity can be redistributed depending on internal and external threats. The objective of surveillance in this domain is to enhance the efficiency of electricity flow among market participants, European allies, and Ukraine. Maintaining a delicate equilibrium between energy producers and consumers is of utmost importance. Both excess and insufficient energy supply pose risks to infrastructure.

The previous wholesale electricity market model, featuring a single buyer, has been substituted with a framework incorporating bilateral, day-ahead, and intraday markets, along with markets dedicated to balancing and auxiliary services. Participants now have the freedom to engage in electricity trading, and energy companies can offer services to guarantee the stability of the power system and supply electricity to the end consumer [1].

Starting from July 01, 2019, in Ukraine the electricity market consists of four components:

- the market of bilateral contracts;
- day-ahead market (DAM);
- intraday market;
- balancing market.

---

*Information Technology and Implementation (IT&I-2023), November 20-21, 2023, Kyiv, Ukraine*

EMAIL: doroshenkoanatoliy2@gmail.com (A. Doroshenko); dmitry.zhora@gmx.com (D. Zhora); savchuk\_11@ukr.net (O. Savchuk); oayat@ukr.net (O. Yatsenko)

ORCID: 0000-0002-8435-1451 (A. Doroshenko); 0009-0006-6073-7751 (D. Zhora); 0000-0003-3176-7952 (O. Savchuk); 0000-0002-4700-6704 (O. Yatsenko);



© 2023 Copyright for this paper by its authors.  
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

It is important to understand the market situation and the structure of supply and demand [2–4] to implement a trade strategy and conduct business successfully, to maximize economic results in the specified market segment. Trading on DAM takes place according to the principle of margin pricing — sellers submit orders at the minimum price at which they are ready to sell, and buyers — at the maximum price at which they are ready to buy. According to the results of auctions on DAM, sellers sell at a price not lower than their ask, and buyers buy at a price not higher than their bid.

The hourly demand for DAM is the hourly amount of electricity that customers declare and want to buy from DAM at the respective hourly prices. This amount directly affects both the marginal price that will be created and the amount of electricity that will be bought and sold. According to the law of supply and demand, the lower the demand for a product, the lower the price; the higher the demand, the higher the price will be. This directly affects the volumes that will be accepted for sale for sellers on DAM. There is a need to forecast the demand for DAM. But now in Ukraine, there are no mechanisms, algorithms, or mathematical models that would allow forecasting with a low forecasting error. The demand for DAM depends on a large number of factors [5–7], in particular:

- volumes of concluded bilateral agreements;
- seasonality;
- blackout due to the destruction of critical energy infrastructure facilities;
- day of the week.

To obtain forecast results as close as possible to real data, the scientists of the G. E. Pukhov Institute for Modeling in Energy Engineering proposed the following [8].

1. Using trading results for the last available day;
2. Using the change in demand between the day on which the actual results are available and the day on which the forecast is made;
3. Taking into account the change in demand, for example, for the previous four weeks between the same days of the week.

A specialized algorithm for short-term hourly-based forecasting of the amount of electrical energy that will be traded on DAM was proposed in [8]. The basis of the developed algorithm is the possibility to formulate the problem in a parametric form, where the forecast and real data of the hourly demand on the DAM were used as indicators. The mathematical specification of the most accurate algorithm is given in terms of a weighted average value for positive and negative changes in demand. The deviation from the arithmetic mean of the absolute values of changes in demand was taken into account to determine atypical days (or outliers in machine learning terminology). The forecast was using statistical data with the amount of electricity purchased on the DAM, using the principle of multi-iteration analysis of changes in demand for previous similar days of previous months (with the possibility of increasing the sampling depth to quarters or years) [9–11]. At the same time, it should be noted that all calculations were performed in Microsoft Excel package. This approach provides availability of all data and calculations, simplicity, and low cost of implementation of such an algorithm.

In previous works [12–14], we've been developing formal and adaptive methods and tools for designing parallel programs that apply artificial intelligence techniques (in particular, genetic algorithms and neuro-evolution methods). In this paper, software facilities for forecasting the market volume one day ahead using machine learning methods are proposed.

## 2. Machine learning techniques used

Machine learning [15, 16] is a set of artificial intelligence methods and is based on analyzing data and identifying patterns within it. The Scikit-learn library [17] is the most common choice for solving classic machine-learning problems. This library provides a diverse set of supervised and unsupervised learning techniques. Supervised learning assumes the presence of a labeled dataset in which the value of the target feature is known. While unsupervised learning does not require the presence of markup in the dataset, it requires learning to extract useful information from arbitrary data. One of the main advantages of this library is that it reuses several common mathematical libraries, and easily integrates them. Another advantage is the wide community and detailed documentation. Scikit-learn is

widely used for industrial systems that use classic machine learning algorithms, for research, as well as for beginners who are just taking their first steps in the field of machine learning.

For its work, Scikit-learn uses the following popular libraries:

- NumPy [18]: mathematical operations and operations on tensors;
- Pandas [19]: data processing, manipulation, and analysis;
- SciPy [20]: scientific and engineering computing;
- Matplotlib [21]: data visualization;
- IPython [22]: interactive console for Python;
- SymPy [23]: symbolic mathematics.

Scikit-learn specializes in machine learning algorithms for solving supervised learning problems: classification (predicting a category with a limited set of valid values) and regression (predicting a feature with real values) [24, 25], as well as for unsupervised learning problems: clustering (partitioning data into classes, which the model will determine itself), dimensionality reduction (representation of data in a lower-dimensional space with minimal loss of useful information) and anomaly detection.

Regression algorithms [26] are used in this paper and are intended to model the relation between input and output data and make predictions based on that relation. Regression can be used for both continuous and categorical input variables. In general, the goal of regression is to build a model that can accurately predict an outcome based on input characteristics and to understand the underlying relation between the input characteristics and the result. Regression analysis is used in various fields, including economics, finance, marketing, and psychology, to understand and predict the relations between various variables. It is a fundamental data analysis and machine learning tool used to make predictions, identify trends, and understand the mechanisms behind data. Regression algorithms are available in many forms, and the use of each algorithm depends on some parameters, such as the type of attribute value, the trendline model, and the number of independent variables. Linear regression is used to predict a value using linear dependency that is based on a set of features and to model the relation between features and a target variable by fitting a line to the data.

The Scikit-learn library implements the following main methods:

- linear: models whose task is to construct a separating (for classification) or approximating (for regression) hyperplane;
- metric: models that calculate the distance using one of the metrics between sample objects, and make decisions depending on this distance ( $k$ -nearest neighbors);
- decision trees: training models based on a set of conditions optimally selected to solve a problem;
- ensemble methods: methods based on decision trees that combine the power of many trees, and thus improve their performance, and also allow for feature selection (boosting, bagging, random forest, majority voting);
- neural networks: a comprehensive nonlinear method for regression and classification problems;
- SVM: a nonlinear method that learns to identify decision boundaries;
- Naive Bayes: forward probabilistic modeling for classification problems;
- PCA: linear method for dimensionality reduction and feature selection;
- t-SNE: a nonlinear dimensionality reduction method;
- $k$ -means: the most common method for clustering, requiring as input the number of clusters into which the data should be distributed;
- cross-validation: a method in which the entire dataset is used for training (as opposed to splitting into train/test samples), but training occurs multiple times, and different parts of the dataset act as a validation sample at each step. The final result is the averaging of the results obtained;
- grid search: a method for finding the optimal hyperparameters of a model by constructing a grid of hyperparameter values and sequentially training models with all possible combinations of hyperparameters from the grid.

In particular, the  $k$ -nearest neighbors algorithm [27] is a metric algorithm for automatic object classification or regression. When using the method for classification, an object is assigned to the class that is the most common among the  $k$  neighbors of a given element, the classes of which are already known. When using the method for regression, an object is assigned an average value over the

$k$  closest objects, the values of which are already known. The algorithm can be applied to samples with a large number of attributes (multidimensional). To do this, the distance function must be determined before use. The classic version of such a function is the Euclidean metric [28], other options are the Manhattan distance [29], cosine measure, and Pearson correlation criterion.

### 3. Forecasting problems for available data

Currently, experiments have been conducted on the application of machine learning methods for forecasting the generation and consumption of electricity using the Scikit-learn open-source software [17].

The volume of each of the four markets mentioned in the introduction is calculated for each hour of the day in megawatts per hour. The example shown in Figure 1 aggregates all data on electricity markets in Ukraine. For comparison, in Europe, market volumes are recorded every 15 minutes.

	A	B	C	D	E	F
1	TradeDate	TradeHour	Bilateral	DayAhead	Intraday	Balancing
8775	2020-06-30	14		3509.7	654.7	-3202.1
8776	2020-06-30	15		3525.4	651.7	-2857
8777	2020-06-30	16		3472.4	668	-2871.5
8778	2020-06-30	17		3448.7	657.7	-2747.5
8779	2020-06-30	18		3418.1	631.4	-2326.3
8780	2020-06-30	19		3420.3	625.4	-1435
8781	2020-06-30	20		3358.9	634.8	-1013
8782	2020-06-30	21		3162.4	637.8	-747.2
8783	2020-06-30	22		3377.9	659.1	-448.1
8784	2020-06-30	23		3430	714.1	-124.2
8785	2020-06-30	24		3277.3	441.9	0
8786	2020-07-01	1	11746.78	2494	1018.2	-2101
8787	2020-07-01	2	11654.98	2697.1	660.3	-2249.5
8788	2020-07-01	3	11606.28	2606.8	624	-2398.8
8789	2020-07-01	4	11637.28	2507.7	614.7	-2681.5
8790	2020-07-01	5	11614.58	2487.2	607.7	-2666.2
8791	2020-07-01	6	11645.48	2629.3	605	-2832
8792	2020-07-01	7	11696.58	2937	610.6	-2455.8
8793	2020-07-01	8	12160.58	3110.5	690.3	-2275.5
8794	2020-07-01	9	12232.48	3399.6	761.4	-2542.5
8795	2020-07-01	10	12357.98	3560.6	770.7	-3038.6
8796	2020-07-01	11	12373.78	3554.5	782.2	-3104.2
8797	2020-07-01	12	12376.88	3505.4	777.1	-3606.8
8798	2020-07-01	13	12398.98	3504.7	809.2	-3770

**Figure 1:** Data on electricity markets in Ukraine. Cells used in the initial version of the algorithm are highlighted in black

We can highlight the following tasks for which it is appropriate to use machine learning methods:

- reproduction of missing values in the database;
- hourly forecasting of the market for 24 hours in advance using previous data for several hours and the sliding window method for their sampling;
- market forecasting with additional use of data for the previous year.

#### 3.1. Restoring the column with data on bilateral agreements

Data from July 2020 to December 2021 were used to form the training and test datasets. The  $k$ -nearest neighbors method was used to populate the Bilateral column. Optimal hyperparameters are the following: the number of neighbors is 14, the metric is Manhattan, and the algorithm takes into

account the distance to each neighbor. In the case of the four input parameters mentioned above, it was experimentally proven that reducing this set worsens the results.

Finding the optimal set of input parameters. The next step is to try using more parameters, in particular from the Bilateral column. The optimal number of input parameters for a machine learning algorithm is always limited. The fact is that when the number of input parameters increases, the algorithm begins to correlate random fluctuations in the input data with the output value. This effect is usually called the curse of dimensionality.

The set of input parameters obtained empirically is shown in Figure 2 and highlighted in black. The values Bilateral1 and Bilateral2 were used at the input of the machine learning model more than once. The optimal value of the hyperparameter in relation to the optimal number of neighbors has changed, now the value is 4. The results of data prediction for various algorithms are shown in Table 2. Figure 3 shows the deviations of the predicted values from the real ones for the test dataset. As can be seen from the graph, the probability distribution of the forecasting error is close to normal.

The intermediate prediction results when increasing the number of cells in the Bilateral column are shown in Figure 4. This research proves that it is optimal to use about 10–11 values from the available historical data for forecasting the bilateral contract market.

**Table 1**  
Results of application of regression algorithms

Regression algorithm	$R^2$ metric (in descending order)	MAPE error (%)
<i>k</i> -nearest neighbors	<b>0,957113</b>	1,830244
Random forest	0,956206	1,809735
Extra trees	0,955950	<b>1,799991</b>
Gradient boosting	0,955913	1,835777
NU support vector machine	0,955321	1,838009
4-layer perceptron	0,953522	1,861381
Linear regression	0,952627	1,855467

Data from July 2020 to December 2021 were used to form the training and test datasets. The *k*-nearest neighbors method was used to populate the Bilateral column. Optimal hyperparameters are the following: the number of neighbors is 14, the metric is Manhattan, and the algorithm takes into account the distance to each neighbor. In the case of the four input parameters mentioned above, it was experimentally proven that reducing this set worsens the results.

Finding the optimal set of input parameters. The next step is to try using more parameters, in particular from the Bilateral column. The optimal number of input parameters for a machine learning algorithm is always limited. The fact is that when the number of input parameters increases, the algorithm begins to correlate random fluctuations in the input data with the output value. This effect is usually called the curse of dimensionality. The set of input parameters obtained empirically is shown in Figure 2 and highlighted in black. The values Bilateral1 and Bilateral2 were used at the input of the machine learning model more than once. The optimal value of the hyperparameter in relation to the optimal number of neighbors has changed, now the value is 4.

The results of data prediction for various algorithms are shown in Table 2.

Figure 3 shows the deviations of the predicted values from the real ones for the test dataset. As can be seen from the graph, the probability distribution of the forecasting error is close to normal.

The intermediate prediction results when increasing the number of cells in the Bilateral column are shown in Figure 4. This research proves that it is optimal to use about 10–11 values from the available historical data for forecasting the bilateral contract market.

### 3.2. Forecasting the market volume one day ahead

The previous section analyses the prediction of bilateral market volume just one hour back. With an available and trained machine learning model, the iterative algorithm was used to restore the

bilateral column for one year range. The results are shown below in Figure 5. The two lines displayed correspond to configurations of input parameters considered earlier.

The chart allows us to conclude that the process is globally sustainable, but eventually, it falls into loops and attractors. Unfortunately, these restored data cannot be reused to train subsequent machine learning models. The left side of the chart is quite different from the real data shown in Figure 6.

	A	B	C	D	E	F
1	TradeDate	TradeHour	Bilateral	DayAhead	Intraday	Balancing
8775	2020-06-30	14		3509.7	654.7	-3202.1
8776	2020-06-30	15		3525.4	651.7	-2857
8777	2020-06-30	16		3472.4	668	-2871.5
8778	2020-06-30	17		3448.7	657.7	-2747.5
8779	2020-06-30	18		3418.1	631.4	-2326.3
8780	2020-06-30	19		3420.3	625.4	-1435
8781	2020-06-30	20		3358.9	634.8	-1013
8782	2020-06-30	21		3162.4	637.8	-747.2
8783	2020-06-30	22		3377.9	659.1	-448.1
8784	2020-06-30	23		3430	714.1	-124.2
8785	2020-06-30	24		3277.3	441.9	0
8786	2020-07-01	1	11746.78	2494	1018.2	-2101
8787	2020-07-01	2	11654.98	2697.1	660.3	-2249.5
8788	2020-07-01	3	11606.28	2606.8	624	-2398.8
8789	2020-07-01	4	11637.28	2507.7	614.7	-2681.5
8790	2020-07-01	5	11614.58	2487.2	607.7	-2666.2
8791	2020-07-01	6	11645.48	2629.3	605	-2832
8792	2020-07-01	7	11696.58	2937	610.6	-2455.8
8793	2020-07-01	8	12160.58	3110.5	690.3	-2275.5
8794	2020-07-01	9	12232.48	3399.6	761.4	-2542.5
8795	2020-07-01	10	12357.98	3560.6	770.7	-3038.6
8796	2020-07-01	11	12373.78	3554.5	782.2	-3104.2
8797	2020-07-01	12	12376.88	3505.4	777.1	-3606.8
8798	2020-07-01	13	12398.98	3504.7	809.2	-3770

**Figure 2:** Data on electricity markets in Ukraine. Cells with empirically obtained parameters are highlighted in black

**Table 2**

Results of application of regression algorithms

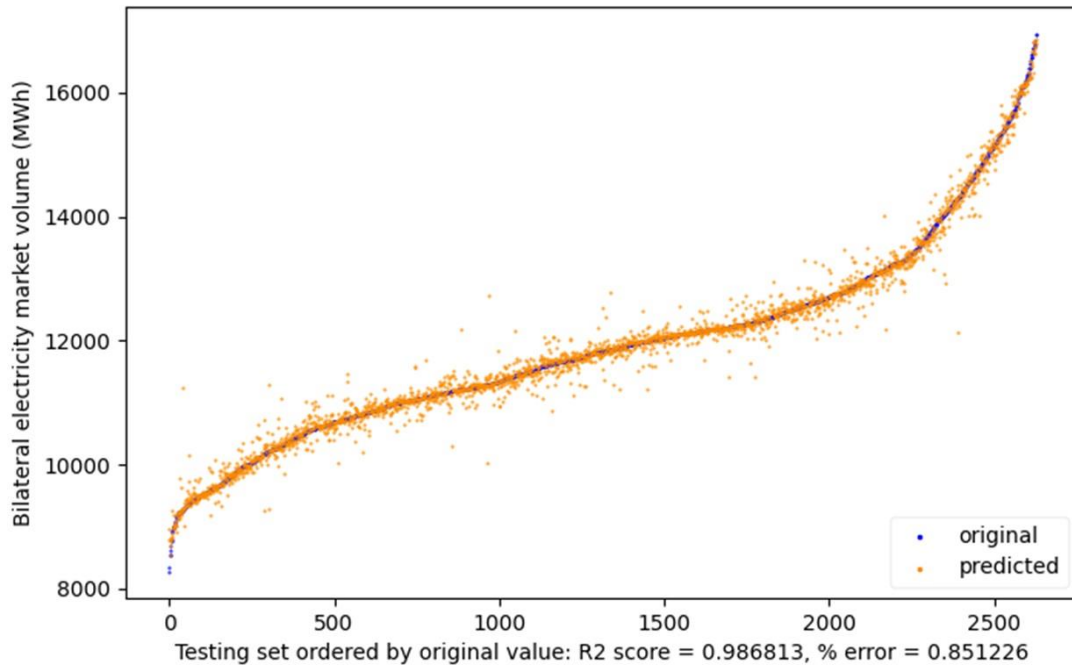
Regression algorithm	$R^2$ metric (in descending order)	MAPE error (%)
<i>k</i> -nearest neighbors	<b>0,986813</b>	<b>0,851226</b>
NU support vector machine	0,982378	1,058797
Gradient boosting	0,982066	1,111225
Extra trees	0,981980	1,076558
Random forest	0,981101	1,113339
4-layer perceptron	0,977275	1,335506
Linear regression	0,969278	1,509372

### 3.3. Forecasting the market volume one day ahead

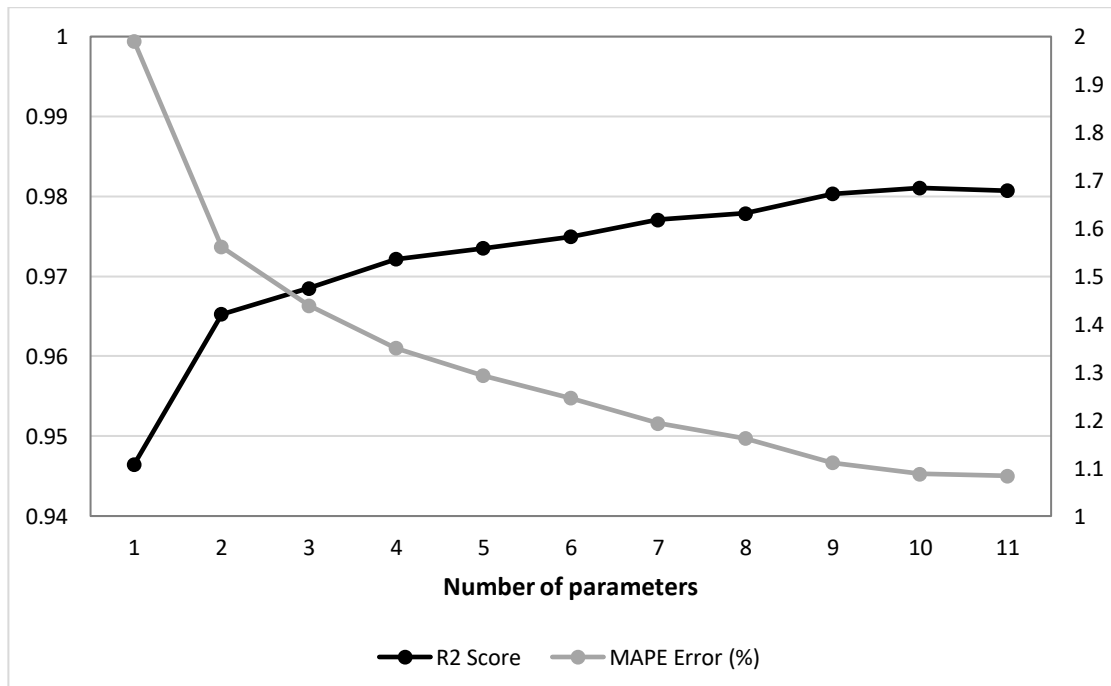
Several issues should be clarified within the framework of the study. First, what is the optimal depth of historical data (in hours) to use for forecasting. This depth may vary for different electricity markets. Another dilemma concerns the iterative approach. It is possible to forecast data 24 times one hour ahead, but it is also possible to forecast for all 24 hours at once, with a fixed state of the input window.

### 3.4. Taking into account time data for the previous year

Using data from the previous year in forecasting can be useful from several points of view. For example, the capacity of solar, wind, and hydro-generating stations depends significantly on the season. The capabilities of solar stations depend even on the time of day. Burning coal for environmental reasons is possible only when other energy sources are already working at full capacity. Data for the previous year already take into account flows of electricity from one market to another. However, the usefulness of last year's data must be determined experimentally.



**Figure 3:** Restoring of the Bilateral column using the  $k$ -nearest neighbors method



**Figure 4:** Dependence of the  $R^2$  metric on the number of parameters from the Bilateral column

## 4. Acknowledgements

We would like to thank the scientists of the G. E. Pukhov Institute for Modeling in Energy Engineering of the National Academy of Sciences of Ukraine for providing the data on electricity markets in Ukraine.

## 5. Conclusion

Effective business operations and the optimization of business processes necessitate the implementation of demand forecasting in electricity markets as a crucial component. We proposed the application of machine learning techniques for forecasting electricity generation and consumption in Ukraine. The experiments were carried out to apply regression methods in reconstructing a column using data from bilateral contracts, specifically to predict electricity generation and consumption. Results from employing machine learning algorithms on hourly peacetime data indicate the potential to forecast market volumes and thus electricity consumption one hour ahead with approximately 1% accuracy. The level of accuracy achieved should enable planning beyond a single day in the future.

## 6. Reference

- [1] Competition market study of Ukraine's electricity sector, OECD, Paris, 2023. URL: <https://read.oecd.org/10.1787/f28f98ed-en?format=pdf>
- [2] G. Klæboe, J. Braathen, A.L. Eriksrud, S.-E. Fleten, Day-ahead market bidding taking the balancing power market into account, *TOP 30* (2022) 683–703. doi:10.1007/s11750-022-00645-1
- [3] X. Ge, K. Li, F. Wang and Z. Mi, Day-ahead market optimal bidding strategy of load aggregator engaging demand response program considering price uncertainty, in: *Proceedings of the 2019 IEEE Power & Energy Society General Meeting (PESGM)*, IEEE, Ney York, NY, 2019, pp. 1–5. doi:10.1109/PESGM40551.2019.8973535.
- [4] R. Qussous, N. Harder, A. Weidlich, Understanding power market dynamics by reflecting market interrelations and flexibility-oriented bidding strategies, *Energies* 15 (2022) 1–24. doi:10.3390/en15020494.
- [5] M. Sharma, N. Mittal, A. Mishra, A. Gupta, Survey of electricity demand forecasting and demand side management techniques in different sectors to identify scope for improvement. *Smart Grids and Energy* 8 (2023) 1–22. doi:10.1007/s40866-023-00168-z
- [6] N. Phuangpornpitak, W. Prommee, A study of load demand forecasting models in electric power system operation and planning, *GMSARN International Journal* 10 (2016) 19–24.
- [7] Y. Guo, Y. Luo, J. He, Y. He, Real-time deep learning-based market demand forecasting and monitoring, *Computers and Electrical Engineering* 100 (2022) 107878. doi:10.1016/j.compeleceng.2022.107878.
- [8] A. Polukhin, V. Evdokimov, Conceptual principles of forecasting demand on the day-ahead market using changes in hourly bidded demand between previous similar days, in: *Proceedings of the 4th International Conference on Sustainable Futures: Environmental, Technological, Social And Economic Matters, Sustainable Energy Workshop (SEnW-2023)*, IOP Conference Series: Earth Environment and Science 1254 (2023) 1–9. doi: 10.1088/1755-1315/1254/1/012035.
- [9] Z. Li, X. Zhang, Y. Li, C. Liu, Day-ahead electricity demand forecasting using a hybrid method, in: *Proceedings of the 4th International Conference on Computer Engineering and Networks, Lecture Notes in Electrical Engineering* 355 (2015) 349–356. doi:10.1007/978-3-319-11104-9\_41.
- [10] M. Pavićević, T. Popović, Forecasting day-ahead electricity metrics with artificial neural networks, *Sensors* 22 (2022) 1–19. doi:10.3390/s22031051.



- [11] O. Trull, J. C. García-Díaz, A. Troncoso, One-day-ahead electricity demand forecasting in holidays using discrete-interval moving seasonalities, *Energy* 231 (2021) 120966. doi:10.1016/j.energy.2021.120966.
- [12] A. Doroshenko, O. Yatsenko, Formal and adaptive methods for automation of parallel programs construction: Emerging research and opportunities, IGI Global, Hershey, PA, 2021. doi: 10.4018/978-1-5225-9384-3
- [13] A. Doroshenko, V. Shymkovich, O. Yatsenko, T. Mamedov, Automated software design for FPGAs on an example of developing a genetic algorithm, in: Proceedings of the 17th Int. Conf. “ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer” (ICTERI 2021), 2021, pp. 74–85. URL: <https://ceur-ws.org/Vol-3013/20210074.pdf>
- [14] A. Y. Doroshenko, I. Z. Achour, O. A. Yatsenko, Parameter-driven generation of evaluation program for a neuroevolution algorithm on a binary multiplexer example, *Radio Electronics, Computer Science, Control*, 1 (2023) 80–88. doi:10.15588/1607-3274-2023-1-8.
- [15] E. Alpaydin, *Machine Learning*, The MIT Press, Cambridge, MA, 2021.
- [16] C. Shah, *A Hands-On Introduction to Machine Learning*, 1st. ed., Cambridge University Press, Cambridge, 2023.
- [17] Scikit-learn 1.3.2. URL: <https://github.com/scikit-learn/scikit-learn/releases/tag/1.3.2>.
- [18] NumPy. URL: <http://www.numpy.org/>
- [19] Pandas. URL: <https://pandas.pydata.org/>
- [20] SciPy. URL: <https://www.scipy.org/>
- [21] Matplotlib. URL: <https://matplotlib.org/>
- [22] IPython. URL: <https://ipython.org/>
- [23] SymPy. URL: <https://www.sympy.org/en/index.html>
- [24] I. McGrath, *Machine Learning — Regression and Classification* (math Inc.): A complete Beginner to Advance level guide to Machine Learning. Hands-on Learning approach with in-depth math concepts, Kindle edition, 2023.
- [25] S. Liu, *Regression: Book One*, Series of Machine Learning with Scikit-Learn, Independently published, 2021.
- [26] J. Frost, *Regression Analysis: An Intuitive Guide for Using and Interpreting Linear Models*, Statistics By Jim Publishing, 2020.
- [27] T. Srivastava, *A Complete Guide to K-Nearest Neighbors* (Updated 2023). URL: <https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/>
- [28] M. M. Deza, E. Deza, *Encyclopedia of Distances*, 4th. ed., Springer, Cham, 2016. doi:10.1007/978-3-662-52844-0.
- [29] R. Suwanda, Z. Syahputra, E. M. Zamzami, Analysis of Euclidean distance and Manhattan distance in the K-means algorithm for variations number of centroid K, *Journal of Physics Conference Series* 1566 (2020) 1–6. doi:10.1088/1742-6596/1566/1/012058

## Appendix A. Fragment of the machine learning algorithm

```

output_names = ['Bilateral']
feature_names = ['Bilateral11', 'Bilateral10',
                 'Bilateral9', 'Bilateral8', 'Bilateral7', 'Bilateral6',
                 'Bilateral5', 'Bilateral4', 'Bilateral3', 'Bilateral2',
                 'Bilateral2', 'Bilateral1', 'Bilateral1', 'Bilateral1',
                 'Bilateral1', 'Bilateral1', 'Bilateral1', 'Bilateral1',
                 'Bilateral1', 'DayAhead1', 'Intraday1', 'DayAhead', 'Intraday']

print(F"Reading the datasets for bilateral column forecasting")
bilateral_dataset = pandas.read_csv('datasets/bilateral-available-dataset.csv')
bilateral_training = pandas.read_csv('datasets/bilateral-available-training.csv')
bilateral_testing = pandas.read_csv('datasets/bilateral-available-testing.csv')

```

```

dataset_inputs = bilateral_dataset[feature_names]
dataset_outputs = bilateral_dataset[output_names]

training_inputs = bilateral_training[feature_names]
training_outputs = bilateral_training[output_names]

testing_inputs = bilateral_testing[feature_names]
testing_outputs = bilateral_testing[output_names]

print("Training the evaluation nearest neighbors regression model")
evaluation_regressor = KNeighborsRegressor(n_neighbors = 4, weights = 'distance',
    algorithm = 'auto', p = 1, metric = 'minkowski', n_jobs = 8)

evaluation_regressor.fit(training_inputs, training_outputs)
predicted_training_outputs = evaluation_regressor.predict(training_inputs)
predicted_testing_outputs = evaluation_regressor.predict(testing_inputs)

print_evaluation_metrics(training_outputs, testing_outputs, \
    predicted_training_outputs, predicted_testing_outputs)

```

## Appendix B. Fragment of the sliding window algorithm

```

print("Populating the missing bilateral market data")

forecast_source_path = 'datasets/bilateral-forecast-source.csv'
print(F"Reading the dataset from file {forecast_source_path}")
forecast_data_frame = pandas.read_csv(forecast_source_path)
forecast_data_frame_length = forecast_data_frame.shape[0]

for i in reversed(range(forecast_data_frame_length)):

    if (i % 100 == 0): print(i)

    if (i == forecast_data_frame_length - 1):
        bilateral11 = forecast_data_frame.iloc[i]["Bilateral11"]
        bilateral10 = forecast_data_frame.iloc[i]["Bilateral10"]
        bilateral9 = forecast_data_frame.iloc[i]["Bilateral9"]
        bilateral8 = forecast_data_frame.iloc[i]["Bilateral8"]
        bilateral7 = forecast_data_frame.iloc[i]["Bilateral7"]
        bilateral6 = forecast_data_frame.iloc[i]["Bilateral6"]
        bilateral5 = forecast_data_frame.iloc[i]["Bilateral5"]
        bilateral4 = forecast_data_frame.iloc[i]["Bilateral4"]
        bilateral3 = forecast_data_frame.iloc[i]["Bilateral3"]
        bilateral2 = forecast_data_frame.iloc[i]["Bilateral2"]
        bilateral1 = forecast_data_frame.iloc[i]["Bilateral1"]

    else:
        bilateral11 = bilateral10
        bilateral10 = bilateral9
        bilateral9 = bilateral8
        bilateral8 = bilateral7
        bilateral7 = bilateral6
        bilateral6 = bilateral5
        bilateral5 = bilateral4

```

```
bilateral4 = bilateral3
bilateral3 = bilateral2
bilateral2 = bilateral1
bilateral1 = bilateral
```

```
forecast_data_frame.loc[i, "Bilateral11"] = bilateral11
forecast_data_frame.loc[i, "Bilateral10"] = bilateral10
forecast_data_frame.loc[i, "Bilateral9"] = bilateral9
forecast_data_frame.loc[i, "Bilateral8"] = bilateral8
forecast_data_frame.loc[i, "Bilateral7"] = bilateral7
forecast_data_frame.loc[i, "Bilateral6"] = bilateral6
forecast_data_frame.loc[i, "Bilateral5"] = bilateral5
forecast_data_frame.loc[i, "Bilateral4"] = bilateral4
forecast_data_frame.loc[i, "Bilateral3"] = bilateral3
forecast_data_frame.loc[i, "Bilateral2"] = bilateral2
forecast_data_frame.loc[i, "Bilateral1"] = bilateral1
```

```
day_ahead1 = forecast_data_frame.iloc[i]["DayAhead1"]
intraday1 = forecast_data_frame.iloc[i]["Intraday1"]
day_ahead = forecast_data_frame.iloc[i]["DayAhead"]
intraday = forecast_data_frame.iloc[i]["Intraday"]
```

```
input_list = [bilateral11, bilateral10, bilateral9, bilateral8, bilateral7, bilateral6,
              bilateral5, bilateral4, bilateral3, bilateral2, bilateral2, bilateral1, bilateral1,
              bilateral1, bilateral1, bilateral1, bilateral1, bilateral1,
              bilateral1, day_ahead1, intraday1, day_ahead, intraday]
```

```
numpy_input = numpy.array(input_list).reshape(1, -1)
numpy_output = production_regressor.predict(numpy_input)
bilateral = round(numpy_output[0][0], 2)
```

```
forecast_data_frame.loc[i, "Bilateral"] = bilateral
```

```
forecast_target_path = 'datasets/bilateral-forecast-target.csv'
print(F"Writing the dataset to file {forecast_target_path}")
forecast_data_frame.to_csv(forecast_target_path, index = False)
```