V. Pliuhin, M. Sukhonos, M. Pan, O. Petrenko, M. Petrenko

IMPLEMENTING OF MICROSOFT AZURE MACHINE LEARNING TECHNOLOGY FOR ELECTRIC MACHINES OPTIMIZATION

Purpose. To consider problems of electric machines optimization within a wide range of many variables variation as well as the presence of many calculation constraints in a single-criteria optimization search tasks. Results. A structural model for optimizing electric machines of arbitrary type using Microsoft Azure machine learning technology has been developed. The obtained results, using several optimization methods from the Microsoft Azure database are demonstrated. The advantages of cloud computing and optimization based on remote servers are shown. The results of statistical analysis of the results are given. Originality. Microsoft Azure machine learning technology was used for electrical machines optimization for the first time. Recommendations for modifying standard algorithms, offered by Microsoft Azure are given. Practical value. Significant time reduction and resources spent on the optimization of electrical machines in a wide range of variable variables. Reducing the time to develop optimization algorithms. The possibility of automatic statistical analysis of the results after performing optimization calculations. References 20, tables 3, figures 7.
Key words: **electrical machines, optimization, algorithm, data set, machine learning, Microsoft Azure, cloud computing.**

Рассмотрены проблемы оптимизации электрических машин при широком диапазоне варьирования многих переменных, наличии большого числа вычисляемых ограничений, в однокритериальных задачах оптимизационного поиска. Разработана структурная модель оптимизации электрических машин произвольного типа с применением технологии машинного обучения Microsoft Azure. Продемонстрированы результаты, полученные с использованием нескольких методов оптимизации из базы Microsoft Azure. Показаны преимущества облачных расчетов и оптимизации на базе удаленных серверов. Приведенные результаты касаются решения однокритериальной задачи оптимизации с двумя переменными. Даны результаты статистического анализа полученных результатов. Даны рекомендации по применению машинного обучения Microsoft Azure в проектировании и оптимизации электрических машин. Библ. 20, табл. 3, рис. 7.
Ключевые слова: **электрические машины, оптимизация, алгоритм, набор данных, машинное обучение, Microsoft Azure, облачные расчеты.**

Introduction. The task of electrical machine (EM) optimal design or a series of EM can be represented as a general non-linear mathematical problem. This problem follows to finding the minimum or maximum of the optimality criterion in the presence of a certain number of independent variables and limiter functions, which are technical or technological requirements-limitations to the project [1-6].

In computer-aided design (CAD) systems, the optimization of an electrical machine consists in multiple calculations of the dependencies between the main indicators given in the form of an equations system, empirical coefficients and graphical dependencies, which can be considered as a design equation [7]. The optimal design of an EM can be represented as the search for optimal parameters by solving this system of equations. The complexity of the calculation algorithm complicates the optimization task.

Reducing the number of independent variables by increasing the number of stages for solving a design problem makes it much easier to find the optimal variant. However, this loses the accuracy of determining the optimal value of the objective function.

Considering CAD in the context of electric machines, it is possible to distinguish the following system components that are used in modern electrical engineering [1]:

1) automated design of an electric machine;
2) search for the optimal version of the designed machine;
3) software implementation of design project and search for optimum;

4) the choice of the optimal variant from the set of effective one, which have been tested for restrictions.

Known methods for searching the optimum version of calculating object, such as the method of coordinate descent, Nelder-Meade, the method of a deformable polyhedron, etc., do not allow performing calculations while changing all configuration variables [8]. As a rule, many methods allow alternating variation of variables with subsequent adjustment of the convergence calculations region [9-12].

Thus, the issue of improving the search for the optimal variation and reducing the time and technical resources, spent on these tasks as well, becomes relevant. In this regard in the paper the development of an optimization model of electric machines, using cloud-based machine learning technology provided by Microsoft Azure services was considered [13, 14].

The aim of the work is the development of a methodology for optimizing electrical machines using Microsoft Azure machine learning technology.

Formulation of the optimization problem. At the optimization stage we assume that the basic version of an electric machine is already calculated (Table 1).

In this case, any electric machine, regardless of its type, turns into a set of initial data (or dataset):
- geometric dimensions;
- winding parameters;
- electrical and magnetic values;
- loss, efficiency, etc.

Table 1
Base machine parameters

Parameter name	Parameter value
Rated power, kW	15
Line voltage, V	380
Rated speed, rev/min	1500
Frequency, Hz	50
Stator core length, mm	130
Stator core inner diameter, mm	185
Efficiency	0.884

The specified dataset, being placed in a one-dimensional vector, can be changed with a given law, obtaining various combinations of the same electric machine. Thus, in order to obtain a machine with the highest efficiency, it was required to find a solution to the following equation:

$$u = f(x_1, x_2,..., x_n),\qquad(1)$$

where $x_1, x_2...x_n$ – varied variables; u – target function.

The search for the optimal value was not limited to finding the extremum of the objective function (1). During the search, candidates were screened out that do not pass the specified restrictions. The number of equality constraints within one project can be arbitrary and is set by the designer:

$$\begin{cases} g_1(x_1,x_2,...,x_n) = 0, \\ g_2(x_1,x_2,...,x_n) = 0, \\ ... \\ g_n(x_1,x_2,...,x_n) = 0. \end{cases}\qquad(2)$$

Inequality constraints are also used:

$$\begin{cases} \alpha_1 \le \gamma_1(x_1,x_2,...,x_n) \le b_1, \\ \alpha_2 \le \gamma_2(x_1,x_2,...,x_n) \le b_2, \\ ... \\ \alpha_k \le \gamma_k(x_1,x_2,...,x_n) \le b_k. \end{cases}\qquad(3)$$

In general case for target function $f(x_1, x_2, …, x_n)$ the minimum m is finding in restricted area D ($x_1, x_2, …, x_n \in D$) [2]. The considered task was replaced by unconditional optimization (minimization) of a one-parameter family of functions:

$$F(x,\beta) = f(x) + \frac{1}{\beta}\varphi(x), \quad x = \{x_1, x_2,..., x_n\},\qquad(4)$$

where $\varphi(x)$ – penalty function; β – penalty factor.

As a penalty function in (4) $\varphi(x)$ was taken, that become zero when the conditions (2) – (3) are fulfilled:

$$\varphi(x) = \frac{1}{\beta}\left\{ \sum_{i=1}^{I} g_i^2(x) + \sum_{j=1}^{J} h_j^2(x)\left[1 - \text{sign } h_j(x)\right] \right\},\qquad(5)$$

$$\beta > 0.$$

In expression (5) the limitations of the equality and inequality types are:

$$g_i(x) = 0, i = 1,2,...,I;$$
$$h_j(x) > 0, j = 1,2,...; x = \{x_1,x_2,...,x_n\}.\qquad(6)$$

The additional (penalty) function $\varphi(x)$ is chosen in the way, when $\beta \to 0$, the solution of the auxiliary problem tends to solve the original one, or that their minimums coincide: min $F(x, \beta) \to m$ while $\beta \to 0$.

To solve the optimization problem, a Java program was written, the functionality of which made it possible to solve the following problems [15]:

- design of the base machine;
- setting restrictions;
- setting a set of varied variables with setting their variation relative to the base value and the step of their change;
- selection of optimality criteria.

When changing only two variable values (stator core length and its internal diameter) in the range ±20 % of the base value, 710000 non-repeating combinations of electric machines were found. Only 441 combinations from this value were passed the restrictions, among which the best option was found. On the Intel Core i3 2.54 GHz processor and 8 Gb RAM, the calculation time was 9 min and 8 s. The results of sampling the selected values are shown on Fig. 1.
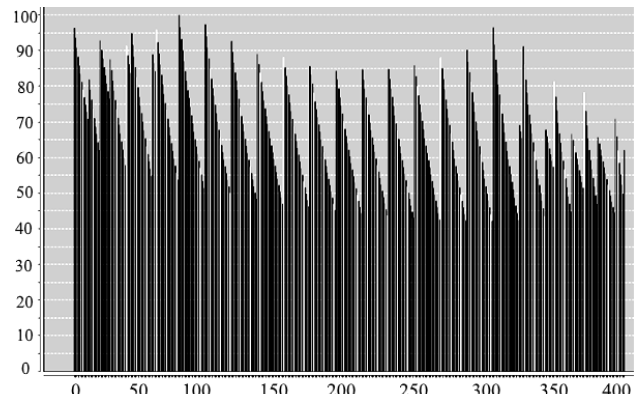

Fig. 1. Sampling combinations diagram of electrical machines in the Java program: Efficiency along the vertical axis and the number of the combination along the X-axis

The obtained optimization results were compared with the experimental data, obtained on two machines with the parameters of the basic and optimized versions, manufactured at «SpesialEnergyService» LLD, Kharkiv, Ukraine. The results of laboratory tests showed a discrepancy with the theoretical no more than 7-8 %.

Performed tests, as well as software solutions of classical optimization methods [2, 3], can be taken for comparison with alternative approaches to optimization.

The disadvantage of the existing method is that the total development time for a Java project was about 3 days (72 hours). In addition, the operating time of the calculated algorithm increases significantly with a change in the range and number of varied values. Fig. 2 shows a comparative chart of the obtained results.

As can be seen from Fig. 2, even at 4 variable variables and the range of their variation ±20 % from the base value, the calculation time was about 8 h.

In real industrial projects of electric machines optimization, it is necessary to vary about 32 parameters, with a range from ±10 % to ±100 % of the base value [3].
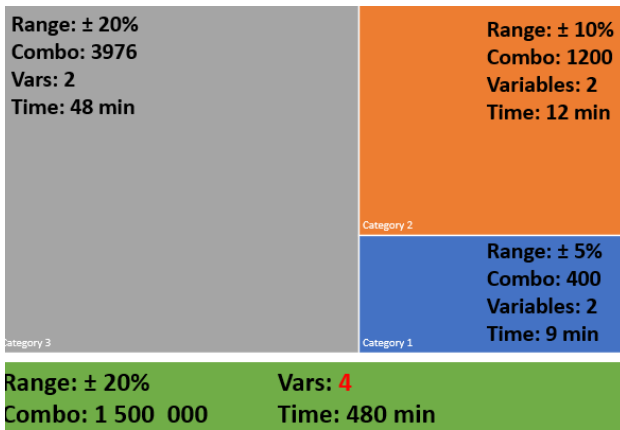
Fig. 2. Comparative chart of time spent on optimization calculations

It is easy to assume that resources of the local PC are not enough to solve such problems, and the debugging time of the project becomes unattainable.

The solution to the problem of operations with large amounts of data (also known as Big Data) and computational operations is the parallelization of calculations and the organization of high-performance computing (HPC) on PC-cluster. However, parallel computing will inevitably entail both changes to the code of an existing program (and an inevitable increase in the debugging time of the program), and will require the presence of the HPC cluster itself.

One solution to this problem is to use the computing power of the Microsoft cloud cluster and machine learning technology based on the Microsoft Azure service.

Developing Microsoft Azure model. Azure Machine Learning enables computers to learn from data and experiences and to act without being explicitly programmed. Customers can build Artificial Intelligence (AI) applications that intelligently sense, process, and act on information – augmenting human capabilities, increasing speed and efficiency, and helping organizations achieve more [16].

Machine Learning finds patterns in large volumes of data and uses those patterns to perform predictive analysis. Microsoft offers Azure Machine Learning, while Amazon offers Amazon Machine Learning and Google offers the Google Prediction API. Software products such as MATLAB support traditional, non-cloud-based ML modeling. There are four steps in the process of finding the best parameter set:

• define the parameter space: for the algorithm, first decide the exact parameter values you want to consider;

• define the cross-validation settings: decide how to choose cross-validation folds for the dataset;

• define the metric: decide what metric to use for determining the best set of parameters, such as accuracy, root mean squared error, precision, recall, or *f*-score;

• train, evaluate and compare: for each unique combination of the parameter values, cross-validation is carried out by and based on the error metric you define. After evaluation and comparison, you can choose the best-performing model.

To iterate on your model design, you edit the experiment, save a copy if desired, and run it again. When you're ready, you can convert your training experiment to a predictive experiment, and then publish it as a web service so that your model can be accessed by others [17].

Elastic cloud infrastructure is the optimal choice for solutions requiring large design capacities in short periods of time. It allows you not to wait for training models for weeks and at the same time not to keep «supercomputers» on balance.

The source data vector (with parameters of the base machine and its non-repeating combinations) for the investigated electrical machine was saved into a *.csv* file (comma separated data) and imported into a block of the Microsoft Azure model. In this table (Table 2) for the test task there were 10 variable values (columns) and 442 combinations (rows).

Table 2

Vector of initial data, imported to Microsoft Azure model

Combo	Diameter	Length	Efficiency	cosφ	…
					…
0	175	120	0.8824	0.8618	…
1	175	121	0.8831	0.8679	…
2	175	122	0.8838	0.8739	…
3	175	123	0.8844	0.8787	…
4	175	124	0.8848	0.8828	…
5	175	125	0.8852	0.8866	…
6	175	126	0.8855	0.8896	…
…	…	…	…	…	…

Statistical analysis of the selected optimality criterion (Efficiency) is performed automatically after importing the source data table into the Microsoft Azure workspace (Table 3).

Table 3

Efficiency statistical performance

Parameter name	Parameter value
Average value	0.8801
Median	0.8817
Minimum value	0.8553
Maximum value	0.8865
Standard deviation	0.0059
Unique values	87
Lost Values	0
Type of analysis	Numeric label

The Microsoft Azure database contains hundreds of computational blocks from which a research task can be made and the complexity of which is limited by the designer's skill [18-20]. Numerous examples of already completed works are available in the Azure cloud. This allow to choose selected one as the basis for the own development.

In this example, the Microsoft Azure project contained the following elements:

• IM_Data – table of parameters;

• Clean Missing Data – deleting of empty rows;

- Select Columns in Dataset – selection of columns of variable parameters;
- Split Data – initial dataset dividing (70% for model teaching in left port and 30% for model analyses using original data in right port);
- Algorithm (Boosted Decision Tree, Multiclass Neural Network);
- Train Model – blocks for model teaching;
- Score Model – block of selection and analysis of the optimality criterion;
- Evaluate Model – block for calculating of statistical information.

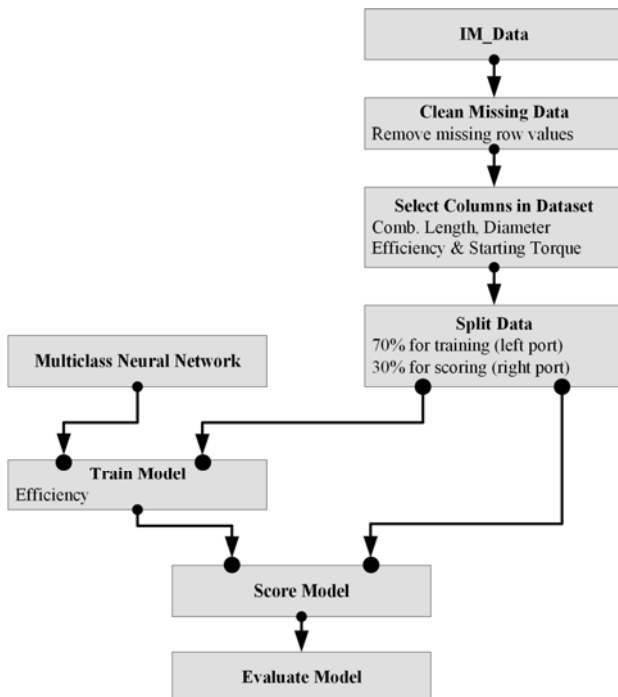The block-scheme of the project is shown on Fig. 3

Fig. 3. The project on the optimization of the electric machine in Microsoft Azure workspace

After performing the calculations, related to the system training, testing the sampling algorithm and searching for the optimum, the final results were obtained. We can get access to these results from «Evaluate Model» block (Fig. 3) and receive various reports. The sample of efficiency data is shown on Fig. 4 and Fig. 5.
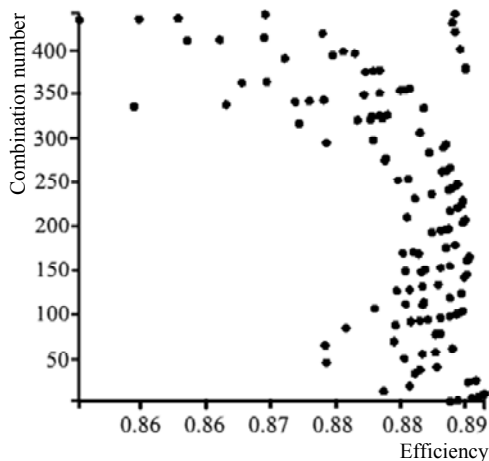
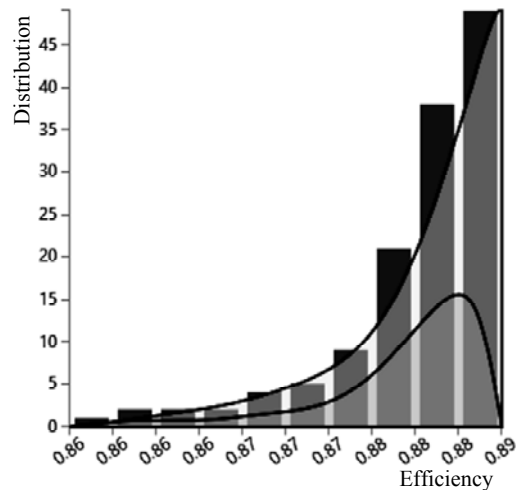Fig. 4 The dependence of the combinations number vs. efficiency

Fig. 5. Efficiency dispersion summary

Fig. 6 shows the report, obtained after analyzing the constructed model in Microsoft Azure using the method of multiclass neural networks. The generated report in tabular format represents the source data sets, followed by columns with calculated deviations from the optimum, as well as statistical indicators (on Fig. 6 only one column (rightmost one) is shown out of 10 available for analysis).

Fig. 7 shows the user project view in the Microsoft Azure workspace, where in addition to the neural network method, the decision tree algorithm, the Poisson regression analysis were included in the analysis of the source data sample.

According to the calculation results, the optimal combination No. 420 was chosen (the choice was made according to the table, the first row of Fig. 6, where the optimization results are sorted in order of increasing error) with the following parameters: stator core length 120 mm, stator core diameter 195 mm, efficiency 0.884.

The computation time was only 1 min 45 s. The metric estimation module built into Microsoft Azure made it possible to determine the quality of the performed calculations. Absolute error 0.000702, standard deviation 0.005926, relative absolute error 0.164582 and relative square error 1.011483 (the lower the value, the better) were obtained.

It should be noted that if the functionality of the embedded Microsoft Azure tools is not enough for some reason, researchers can write and execute their own scripts on *R* or *Python* [13, 14].

Thus, the use of Microsoft Azure in optimizing electrical machines has been demonstrated. In the shown example, only one data vector was used and there were no modules for intermediate processing and data transfer between the modules.

Further research will be focus on creating own *Python* calculating blocks and *R* scripts with a view to transferring to the Microsoft Azure platform not only data set (now this data set is forming based on the results of separate calculations in *Java* program), but also creating a population of source data based on the vector parameters of the base machine.

	rows	columns
	133	131

Efficiency	PowerFactor	Induction	StartCurrent	StartTorque	MaxTorque	Temperature	Scored Probabilities for Class "0.8556"
0.8849	0.9052	0.709	5.65	1.35	2.53	90.6	0
0.8828	0.8793	0.73	5.61	1.39	2.59	92.9	0.000001
0.8778	0.8582	0.729	5.61	1.42	2.66	96.2	0.000792
0.8696	0.8256	0.659	5.12	1.36	2.57	95.9	0.001181
0.8779	0.8524	0.768	5.76	1.48	2.76	98.9	0.00042
0.8811	0.8598	0.769	5.72	1.46	2.72	96.8	0.000045
0.8835	0.9042	0.684	5.54	1.32	2.49	89.7	0
0.8765	0.8627	0.676	5.36	1.35	2.55	93.1	0.000138
0.88	0.8829	0.683	5.48	1.34	2.53	91.6	0.000021
0.8831	0.8805	0.743	5.7	1.41	2.63	93.8	0.000004

Fig. 6. Results of the neural network sampling algorithm in the Microsoft Azure report table
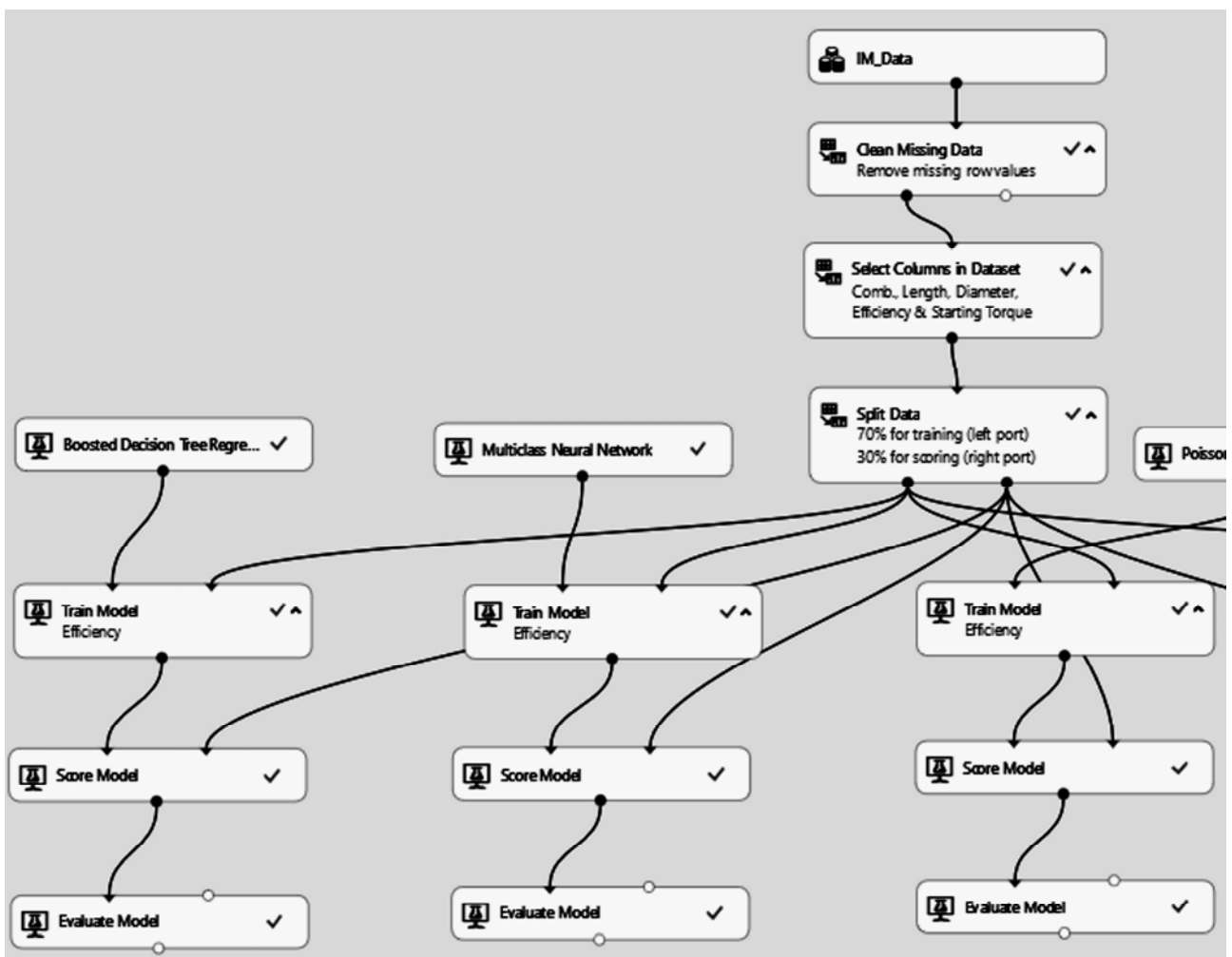(screenshot of the project table in the browser workplace)

Fig. 7. Full project model in Microsoft Azure (Screenshot of the project model in the browser workplace)

Conclusions.

The application of Microsoft Azure machine learning technology in electrical machines optimizing is shown for the first time.

As a result of the performed optimization using Microsoft Azure cloud services, the computation time was reduced by more than 300 times (from 480 minutes to 2 minutes) when solving the same task compared to calculations on a stationary PC.

The project development time of an electric machine, on the example of an induction motor with a squirrel-cage rotor, was reduced from several working days to 30 minutes.

The complexity of project developing (induction motor parameters optimization task) has significantly decreased compared to direct *Java* programming, due to the use of ready-made analysis units provided by Microsoft Azure.

The Microsoft Azure platform for the implementation of machine learning technology can be recommended in solving optimization problems of various electric machine types.

REFERENCES

1. Sen S.K. *Principles Of Electrical Machine Design With Computer Programs*. Oxford, IBH Publishing Company Pvt. Limited, 2006. 415 p.

2. Rekleitis G., Reivindran A., Regsdel K. *Optimizatsiia v tekhnike* [Optimization in technology]. Moscow, Mir Publ., 1986. 351 p. (Rus).

3. Goriagin V.F., Zagriadskii V.I., Sycheva T.A. *Optimal'noe proektirovanie asinkhronnykh vzryvozashchishchennykh dvigatelei* [Optimal design of asynchronous explosion-proof motors]. Kishinev, Shtiitsa Publ., 1980. 200 p. (Rus).

4. Zablodskiy N., Lettl J., Pliugin V., Buhr K., Khomitskiy S. Induction Motor Optimal Design by Use of Cartesian Product. *Transactions on electrical engineering*, 2013, no.2, pp. 54-58.

5. Zablodskiy N., Lettl J., Pliugin V., Buhr K., Khomitskiy S. Induction Motor Design by Use of Genetic Optimization Algorithms. *Transactions on electrical engineering*, 2013, no.3, pp. 65-69.

6. Zablodskii N.N., Pliugin V.E., Petrenko A.N. Using object-oriented design principles in electric machines development. *Electrical engineering & electromechanics*, 2016, no.1, pp. 17-20. **doi: 10.20998/2074-272X.2016.1.03**.

7. Papalambros P.Y., Wilde D.J. *Principles of optimal design*. Cambridge University Press, 2017. **doi: 10.1017/9781316451038**.

8. Severin V.P., Nikulina E.N. *Metody odnomernogo poiska* [Methods of one-dimensional search]. Kharkiv, NTU KhPI Publ., 2013. 124 p. (Rus).

9. Todorov E. Optimal control theory. *Bayesian Brain: Probabilistic Approaches to Neural Coding*, 2006, chap. 12, pp. 268-298. **doi: 10.7551/mitpress/9780262042383.003.0012.**

10. Kappen H.J. Optimal control theory and the linear Bellman equation. *Bayesian Time Series Models*, 2011, pp. 363-387. **doi: 10.1017/cbo9780511984679.018**.

11. Kanemoto Y. *Theories of urban externalities*. Holland, North-Holland Publ., 1980. 189 p.

12. Satit Owatchaiphong, Nisai Fuengwarodsakul. Multi-Objective Design for Switched Reluctance Machines Using Genetic and Fuzzy Algorithms. *The ECTI Transactions on Electrical Engineering, Electronics, and Communications*, 2013, vol.11, no.2, pp. 1530-1533.

13. Chappel D. *Introduction Azure machine learning: a guide for technical professionals*. Chappel & Associates, 2015.

14. Collier M., Shahan R. *Microsoft Azure Essentials*. Microsoft Press, 2016.

15. Bloch J. *Effective Java*. Canada, Sun Microsystems, 2008.

16. *What is Azure machine learning studio?* Available at: https://docs.microsoft.com/en-us/azure/machinelearning/studio/what-is-ml-studio (Accessed 10 May 2018).

17. Hayakawa S., Hayashi H. Using Azure Machine Learning for Estimating Indoor Locations. *2017 International Conference on Platform Technology and Service (PlatCon)*, Busan, 2017, pp. 1-4. **doi: 10.1109/platcon.2017.7883736**.

18. *Azure Machine Learning*. Available at: https://azure.microsoft.com/en-us/services/machine-learning-studio/ (Accessed 22 April 2018).

19. *Dig Deep with Azure Machine Learning*. Available at: https://studio.azureml.net/ (Accessed 16 February 2018).

20. Microsoft Azure Machine Learning: Algorithm Cheat Sheet. Available at: http://aka.ms/MLCheatSheet (Accessed 05 March 2018).

V. Pliuhin[1]*, Doctor of Technical Sciences, Professor,*
M. Sukhonos[1]*, Doctor of Technical Sciences, Professor,*
M. Pan[1]*, Candidate of Technical Sciences, Professor,*
O. Petrenko[1]*, Doctor of Technical Sciences, Associate Professor,*
M. Petrenko[2]*, Candidate of Technical Sciences, Associate Professor,*
[1] O.M. Beketov National University of Urban Economy in Kharkiv,
17, Marshal Bazhanov Str., Kharkiv, 61002, Ukraine,
e-mail: petersanya1972@gmail.com
[2] National Technical University «Kharkiv Polytechnic Institute», 2, Kyrpychova Str., Kharkiv, 61002, Ukraine.