UDC 681.5
DOI: 10.15587/2706-5448.2025.326909

Illia Velychko,
Viktor Sidletskyi

DEVELOPMENT OF A NEURAL NETWORK MODEL FOR AN AUTOMATED HVAC SYSTEM BASED ON COLLECTED DATA

The object of research is ventilation and air conditioning systems, which act as the object of data collection for the development of a neural network model based on them. The main attention is paid to the choice of algorithm, data collection for training a neural network model based on the MATLAB software package, to simplify the model development process.

The main problem that was considered in the study is the complexity of building mathematical models for ventilation and air conditioning systems. Traditional approaches require significant computing resources and in-depth analysis of physical processes, which complicates their development and practical use.

The research results show one of the approaches to creating a model of ventilation and air conditioning systems using neural networks. The proposed approach provides fast training of the model based on real data, which in further studies will allow adapting the system to changing operating conditions and increasing its efficiency.

The obtained results are explained by the fact that, unlike classical mathematical models that require precise formulation of all dependencies and parameters. Neural networks are able to approximate complex nonlinear functions without the need for a complete understanding of physical processes.

The proposed approach can be used for ventilation and air conditioning systems provided that there is a sufficient amount of data for training the neural network. Also important is the integration of such a system with controllers and SCADA systems that provide operational collection of parameters from the environment. The use of neural network models is especially effective in smart buildings, industrial facilities and energy-saving systems, where it is important to optimize energy consumption and provide comfortable conditions for users. In addition, such models can be implemented in cloud platforms for centralized management of climatic parameters in various buildings or production complexes.

Keywords: *microclimate control, HVAC automation, machine learning, energy efficiency, neural networks.*

Received: 19.01.2025
Received in revised form: 12.03.2025
Accepted: 04.04.2025
Published: 18.04.2025

How to cite
Velychko, I., Sidletskyi, V. (2025). Development of a neural network model for an automated HVAC system based on collected data. Technology Audit and Production Reserves, 2 (2 (82)), 21–26. https://doi.org/10.15587/2706-5448.2025.326909

1. Introduction

Modern ventilation and air conditioning systems require accurate modeling to ensure energy efficiency and comfort [1]. Traditional approaches are based on physical equations (e. g., heat transfer equations, CFD modeling [2]). Whereas methods using neural networks offer an alternative based on collected or experimental data.

Comparing traditional mathematical models [3, 4] and models based on artificial neural networks (ANNs) in the context of designing ventilation and air conditioning systems allows to identify their strengths and weaknesses.

Traditional mathematical models are based on physical laws [5] and equations describing the behavior of the system. They require a deep understanding of physical processes, significant computational resources, and accurate determination of system parameters. Building such models can be difficult and time-consuming, especially for systems with a high degree of complexity and nonlinearity.

Models based on artificial neural networks [6] offer an alternative approach. They are able to automatically detect complex dependencies between input and output parameters of the system, without requiring detailed knowledge of physical processes. The methods used to build a neural network model can be used on the basis of experimental (or collected) data, which simplifies the modeling process [7] and allows to adapt to changing operating conditions. In particular, studies [8, 9] show that the use of neural network models for predicting the state of ventilation systems is effective, and show minimal deviations from real data.

Thus, *the aim of the research* is to develop a neural network model for an automated ventilation and air conditioning system based on collected data. The modeling used Function Fitting Neural Network (FFNN) – MATLAB, which allows to predict the dynamics of the ventilation system.

2. Materials and Methods

2.1. Data collection tools and software implementation tools

This paper considers the process of developing a neural network model based on collected data from an automated ventilation and air conditioning system.

The main task of such systems is:

– ensuring the regulatory amount of outside (fresh) air for the premises in which activities are carried out;
– maintaining the required level of air humidity;
– maintaining the required air temperature.

 ISSN-L 2664-9969; E-ISSN 2706-5448

To create a model based on the collected data, the MATLAB environment was chosen – an interactive platform for scientific computing, modeling and programming. The Neural Network Fitting Tool was used – a MATLAB tool for creating, training and testing neural networks designed for function approximation (regression). The Neural Network Fitting Tool is part of the Deep Learning Toolbox and is designed to predict numerical values based on input data. The neural network model was trained using three algorithms to select the best one, namely:

- Levenberg-Marquardt;
- Bayesian Regularization;
- Scaled conjugate gradient.

Each of the selected algorithms was used to train the neural network in two separate experiments, differing in the number of neurons in the hidden (10 and 20 neurons) layer. This approach allowed to assess the impact of the dimension of the hidden layer on the effectiveness of the model.

In order to ensure the collection of experimental data necessary for building the model (Fig. 1), the study used a freely programmable logic controller Carel c.pCO (made in Italy). Based on this controller, an algorithm for controlling ventilation and air conditioning systems was implemented.

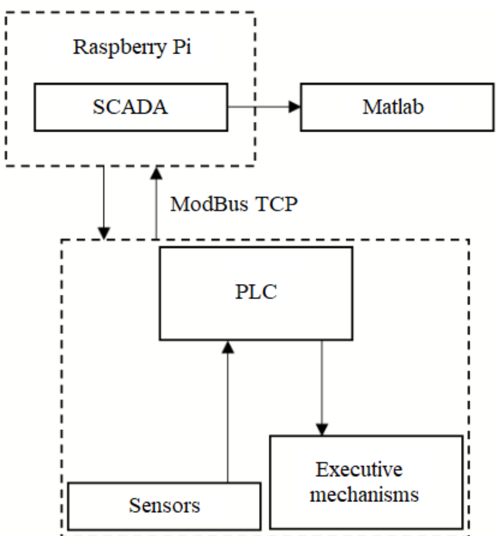

Fig. 1. Data acquisition scheme for training

Data acquisition is provided by a SCADA system (Fig. 2) deployed on the Raspberry Pi platform.

The connection between the controller and the SCADA system is implemented using the industrial data exchange protocol – Modbus TCP.

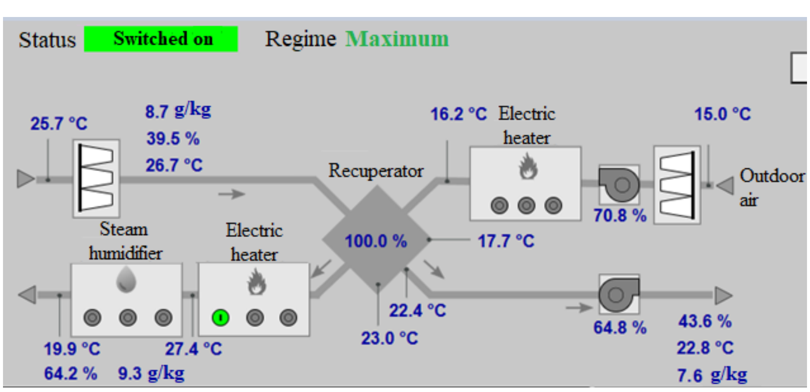

Fig. 2. Mnemonic diagram of the ventilation and air conditioning system

2.2. Overview of the collected data for model training

The data (Table 1) is used in the network training process. From the first to the sixth of July – the "SUMMER" mode and from the first to the sixth of February – the "WINTER" mode. For correct training of the neural network model, it was decided to collect data from the system in two modes of its operation, namely – "WINTER/SUMMER". A total of 24504 data samples were used for training, of which 19603 – the training sample, 2450 – the validation sample, 2450 – the test sample.

This approach allows to build a stable model using a large amount of data, while maintaining a certain amount of independent data to check the generalization ability.

The Neural Network Fitting Tool uses sample data to build a neural network model, as traditional empirical approaches often fail to account for the nonlinearity of temperature and humidity processes. The Neural Network Fitting Tool learns to predict future changes based on historical data.

Table 1

Sample data for training

Input data for training	Control output
Outside air temperature, °C	Electric heater 2 Stage 3 (0 – OFF, 1 – ON)
Supply air temperature after electric heater, °C	Electric heater 2 Stage 2 (0 – OFF, 1 – ON)
Air temperature after heat exchanger – sensor 3, °C	Electric heater 2 Stage 1 (0 – OFF, 1 – ON)
Air temperature after heat exchanger – sensor 2, °C	Electric heater Stage 1 (0 – OFF, 1 – ON)
Air temperature after heat exchanger – sensor 1, °C	–
Supply air temperature after second electric heater, °C	–
Supply air temperature after humidifier, °C	–
Absolute air humidity, g/kg	–
HVAC/System status	–
Supply air humidity setpoint, g/kg	–
Supply air temperature setpoint, °C	–

As part of the study, two experiments were conducted using the Levenberg-Marquardt algorithm, the difference between which was the number of neurons in the hidden layer. The structure of the neural network is presented in Fig. 3. The network architecture includes eleven input parameters that are fed to one hidden layer. Each neuron receives input values that are multiplied by weight coefficients W, the weights are updated according to formula (1), and *a* bias *b* is added:

$$\omega_{k+1} = \omega_k - (J^T \cdot J + \lambda I)^{-1} J^T e, \tag{1}$$

where J – the Jacobi matrix of the derivative errors; is the error vector; λ – the regularization parameter.

The activation function tansig (2) (hyperbolic tangent) is used:

$$f(n) = \tanh(n) = \frac{2}{1+e^{-2\pi}} - 1. \tag{2}$$

In the process of using the second learning algorithm, namely Bayesian Regularization, two experiments were conducted that involved varying the number of neurons in the hidden layer, similar to the approach used in the first algorithm. The Bayesian Regularization algorithm is based on the Bayesian approach aimed at increasing the generalization ability of the neural network. This is achieved by adding a regularization coefficient (3) to the standard error function, which allows improving the stability and accuracy of network training.

$$E = E_D + \lambda E_w = \sum (y_i - \hat{y}_i)^2 + \lambda \sum \omega_j^2 E = E_D + \lambda E_w, \quad (3)$$

where E_D – the prediction error; E_w – the regularization coefficient (sum of squares of weights); λ – the regularization coefficient that balances the accuracy and complexity of the model.

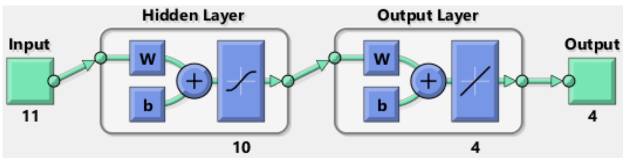

Fig. 3. Neural network structure

In the process of using the third learning algorithm, Scaled Conjugate Gradient, two separate experiments were conducted. The algorithm uses the conjugate gradient method to calculate updates (4) of the weight coefficients.

To ensure comparability of the results with previous experiments, the number of neurons was adjusted accordingly:

$$w_{k+1} = w_k + \alpha_k p_k, \quad (4)$$

where w_k – the vector of weight coefficients at the iteration k; α_k – the step size; p_k – the descent direction.

3. Results and Discussion

3.1. Using the Levenberg-Marquardt algorithm

Based on two (Fig. 4, 5) experiments with different numbers of neurons in the second (hidden) layer, the following results of training the neural network model were obtained. The stability of the obtained mean square error (MSE) values for different samples indicates the effectiveness of the model. The MSE results obtained on the training, validation and test samples are almost the same, which indicates the ability of the model to generalize data properties and avoid overtraining. At the same time, an increased MSE level on the test sample may indicate certain limitations of the model in predicting new data.

Results

	Samples	MSE	R
Training:	19603	1.27906e-1	6.89797e-1
Validation:	2450	1.26901e-1	6.92495e-1
Testing:	2450	1.31651e-1	6.78872e-1

Fig. 4. Results of network training – 10 neurons (Levenberg-Marquardt algorithm)

Below are the results of training a neural network (Fig. 5) with twenty neurons in the hidden layer.

Results

	Samples	MSE	R
Training:	19603	1.25609e-1	6.96917e-1
Validation:	2450	1.28433e-1	6.88387e-1
Testing:	2450	1.26339e-1	6.91503e-1

Fig. 5. Results of network training – 20 neurons (Levenberg-Marquardt algorithm)

After comparing the results, it is possible to say that for this algorithm and with the available collected data, increasing the number of neurons in the hidden layer mainly has a beneficial effect only on the mean square error. The values of the error correlation coefficient do not change significantly.

Analyzing the values of Fig. 4, 5, it is possible to conclude that a neural network trained on a data sample can return output values very close to the expected result using the algorithm. Increasing the number of neurons in the hidden layer does not make significant, positive changes, but increases the time for network training.

3.2. Using the Bayesian regularization algorithm

To analyze the efficiency of the considered neural network model training algorithm, a comparison of the results (Fig. 6, 7) was carried out for different numbers of neurons in the hidden layer.

Results

	Samples	MSE	R
Training:	19603	1.27595e-1	6.90665e-1
Validation:	2450	0.00000e-0	0.00000e-0
Testing:	2450	1.28876e-1	6.87375e-1

Fig. 6. Results of network training – 10 neurons (Bayesian regularization algorithm)

Results

	Samples	MSE	R
Training:	19603	1.24006e-1	7.01202e-1
Validation:	2450	0.00000e-0	0.00000e-0
Testing:	2450	1.25583e-1	6.97391e-1

Fig. 7. Results of network training – 10 neurons (Bayesian regularization algorithm)

Analysis of the obtained data allows to conclude that increasing the number of neurons from 10 to 20 in the hidden layer leads to a slight improvement in the model's performance, but at the same time its computational complexity and potential risk of overtraining increase.

3.3. Using the gradient descent algorithm

To assess the effectiveness of the gradient descent algorithm in training a neural network model, an analysis of the results (Fig. 8, 9) obtained with different numbers of neurons in the hidden layer was conducted.

Results

	Samples	MSE	R
Training:	19603	1.36239e-1	6.64680e-1
Validation:	2450	1.40824e-1	6.50697e-1
Testing:	2450	1.36880e-1	6.61486e-1

Fig. 8. Results of network training – 10 neurons (gradient descent algorithm)

Results

	Samples	MSE	R
Training:	19603	1.37376e-1	6.60951e-1
Validation:	2450	1.38564e-1	6.58088e-1
Testing:	2450	1.34848e-1	6.69052e-1

Fig. 9. Results of network training – 20 neurons (gradient descent algorithm)

ISSN-L 2664-9969; E-ISSN 2706-5448

Analyzing the obtained results, it is possible to conclude that the increase in neurons (Fig. 8) did not have a significant positive impact on the accuracy of training the neural network relative to the neural network with a smaller number of neurons (Fig. 9). The root means square error obtained during training significantly exceeds the indicators achieved using the two previous algorithms. This indicates a less efficient training of the model when using this algorithm.

3.4. Comparison of the efficiency of the neural network under different training algorithms and architectures

The obtained results indicate that increasing the number of neurons in the hidden layer does not lead to a noticeable improvement in the accuracy of the models (Tables 2, 3). The speed of model learning remains an important factor (Table 4), since further studies plan to increase the sample size.

Table 2

Obtained values of the mean square error

Algorithm	Number of neurons	Training MSE	Validation MSE	Testing MSE
Levenberg-Marquardt	10	0.1279	0.1269	0.1316
Levenberg-Marquardt	20	0.1256	0.1284	0.1263
Bayesian regularization	10	0.1276	0.0000	0.1289
Bayesian regularization	20	0.1240	0.0000	0.1258
Gradient descent	10	0.1362	0.1408	0.1316
Gradient descent	20	0.1374	0.1386	0.1348

Table 3

Obtained values of the correlation coefficient

Algorithm	Number of neurons	Training R	Validation R	Testing R
Levenberg-Marquardt	10	0.6898	0.6925	0.6789
Levenberg-Marquardt	20	0.6970	0.6884	0.6915
Bayesian regularization	10	0.6907	0.0000	0.6874
Bayesian regularization	20	0.7101	0.0000	0.6973
Gradient descent	10	0.6649	0.6509	0.6148
Gradient descent	20	0.6691	0.6588	0.6905

Table 4

Learning time during experiments

Algorithm	Levenberg-Marquardt	Bayesian regularization	Gradient descent
Time (10 neurons)	6 s	25 s	2 s
Time (20 neurons)	49 s	146 s	5 s

3.5. Choosing a training algorithm

Having considered the results of the three neural network training algorithms and compared them, it is possible to conclude that for the data sample used for training, it does not matter which of the algorithms will be used. The results of training the neural network model using all methods show almost identical values of the correlation coefficient and the mean square error.

To check the correctness of the work, a set of input data was selected, which was used for training and the Levenberg-Marquardt algorithm. The check was carried out by feeding the values (Table 5) that were used for its training to the input of the neural network and the following results were obtained.

During the training of the neural network model, a script was generated (Fig. 10), which makes it possible to retrain the neural network model, change the number of hidden layer neurons and input/output data for training.

Table 5

Data sample for testing

Parameter name	Parameter value
Outside air temperature, °C	25.9
Supply air temperature after electric heater, °C	26.8
Air temperature after heat exchanger – sensor 3, °C	27.8
Air temperature after heat exchanger – sensor 2, °C	28.8
Air temperature after heat exchanger – sensor 1, °C	27.9
Supply air temperature after second electric heater, °C	27.5
Supply air temperature after humidifier, °C	27.5
Absolute air humidity, g/m^3	12.5
HVAC/System status	1 (ON)
Supply air humidity setpoint, g/m^3	8
Supply air temperature setpoint, °C	20

```
x = data';
t = data_1';
trainFcn = 'trainlm';  % Levenberg-Marquardt backpropagation.
hiddenLayerSize = 10;
net = fitnet(hiddenLayerSize,trainFcn);
net.input.processFcns = {'removeconstantrows','mapminmax'};
net.output.processFcns = {'removeconstantrows','mapminmax'};
net.divideFcn = 'dividerand';
net.divideMode = 'sample';
net.divideParam.trainRatio = 80/100;
net.divideParam.valRatio = 10/100;
net.divideParam.testRatio = 10/100;
net.performFcn = 'mse';
net.plotFcns = {'plotperform','plottrainstate','ploterrhist',
'plotregression', 'plotfit'};
[net,tr] = train(net,x,t);
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)
trainTargets = t .* tr.trainMask{1};
valTargets = t .* tr.valMask{1};
testTargets = t .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)
view(net)
```

Fig. 10. Generated script during training

To check the correctness of the work (Table 6), it is necessary to improve (Fig. 11) the generated script to feed the input values to the neural network and display the received output data.

Analysis of the data presented in Table 6 confirms that the neural network model trained using the Levenberg-Marquardt algorithm provides high accuracy of the output data based on the sample used.

Table 6

Checking the correctness of the output data

Output	Output data (expected)	Results obtained (10 neurons)	Results obtained (20 neurons)
Electric heater 2 Stage 3 (0 – OFF, 1 – ON)	0	0	0
Electric heater 2 Stage 2 (0 – OFF, 1 – ON)	0	0	0
Electric heater 2 Stage 1 (0 – OFF, 1 – ON)	0	0	0
Electric heater Stage 1 (0 – OFF, 1 – ON)	0	0	0

```
op = testoptimizapion1';
a = round(net(op(:,1)),1)
```

Fig. 11. Improving the generated script

In the future, the resulting model can be used as a prediction function in adaptive or nonlinear control systems Model Predictive Control (MPC), which allows replacing the classical mathematical description of the system with a "black box" that can take into account nonlinear relationships and complex dependencies.

3.6. Discussion of the results

In the process of modeling a neural network for a ventilation and air conditioning system, the influence of various learning algorithms and network architectures on the quality of approximation of input and output parameters was analyzed. Six experiments were conducted to develop a neural network model using three learning algorithms and changing the number of hidden layer neurons.

This behavior can be explained by the limited complexity of the studied process. In such cases, additional neurons do not increase the quality of the model, since the data simply do not carry more complexity that could be reflected. It is worth noting that data collection and model training were carried out in conditions of limited time for this time, due to interruptions in the operation of electrical networks and Internet communication. To a large extent, the presence of such conditions reduced the possibility of collecting qualitative data in a larger volume from the existing system due to its unstable operation and access for data collection in general.

The correlation coefficient obtained on the best model (about 0.69) indicates a moderate level of dependence between input and output parameters. On the one hand, this confirms the existence of a relationship between control influences and the results of the model. On the other hand, the relatively low correlation value indicates the complexity of the processes occurring in the ventilation and air conditioning system, and the possible presence of nonlinearities and external influences that were not taken into account in the input data.

It is also worth noting that data from a temperate continental climate were used to obtain the model. Such data determine a very wide range of temperatures, with cold winters and hot summers. Therefore, the obtained results can be used with the highest quality in similar climatic conditions.

The work uses the tools of the MATLAB software package to develop a neural network model designed to control ventilation and air conditioning systems. The use of MATLAB simplifies the development of neural network models for ventilation and air conditioning systems compared to conventional developments that use high-level programming languages and mathematical modeling through the use of tools such as the Neural Network Fitting Tool.

Despite the obtained results, the use of neural network models in ventilation and air conditioning systems has the following disadvantages that must be taken into account:

– *The need for a large amount of data*: High-quality training of models requires the availability of large data sets covering various scenarios of system operation. If the data is incomplete or of low quality, the model may create a distorted picture of the dynamics of the system.

– *Data relevance*: Data can quickly become outdated due to changing operating conditions, equipment modernization, or changing climatic conditions. This can reduce the effectiveness of the model if it is not retrained regularly.

– *Risk of incorrect operation*: If the system operates on the basis of a model that has not been properly tested or updated, this can lead to control errors, which will negatively affect the comfort and efficiency of the equipment.

For further development of the proposed approach, it is necessary to:

– expand the data sample from the real system, increase the sampling time and scaling of the collected data, integrate the neural network model into ModelPredictiveControl;

– develop a mechanism for self-learning the model based on new operational data;

– investigate the possibilities of integrating the neural network model into adaptive or nonlinear control systems Model Predictive Control (MPC).

4. Conclusions

As part of the research, an artificial neural network model for the ventilation and air conditioning system was developed based on data collected through the SCADA system and the Carel c.pCO controller. The network was trained in the MATLAB environment using the Neural Network Fitting Tool. To assess the impact of the architecture and optimization method, three training algorithms (Levenberg-Marquardt, Bayesian Regularization and Scaled Conjugate Gradient) were used in two hidden layer configurations (10 and 20 neurons). A sample of 19,603 values was used for training and 2,450 for validation and testing, respectively. The analysis showed that changing the number of neurons in the hidden layer did not lead to a significant improvement in accuracy. For example, for the Levenberg-Marquardt algorithm, increasing the number of neurons from 10 to 20 resulted in a slight decrease in the mean square error on the test set from 0.1316 to 0.1263, and the correlation coefficient even decreased from 0.6789 to 0.6915, which is not a significant improvement in quality.

The lowest mean square error (0.1256) and the highest correlation coefficient (0.6973) were achieved using a model trained using Bayesian regularization with 10 neurons in the hidden layer. This indicates the ability of algorithms with built-in regularization to effectively generalize information, which is especially important in the context of technical systems.

It was also found that none of the algorithms demonstrated a critical advantage in accuracy, but Bayesian regularization turned out to be more stable and less prone to overtraining (no validation error).

The results obtained demonstrate an important feature – when modeling technical processes using neural networks, algorithms that have built-in mechanisms for controlling the complexity of the model, in particular Bayesian regularization, show high efficiency. This allows achieving a balance between accuracy and generalization ability, which is critically important for applications in ventilation and climate systems, where the training sample does not cover all possible operating modes.

The practical significance of the results is manifested in the ability to use the obtained architecture and training methodology to create digital models of ventilation and air conditioning systems operating under conditions of uncertainty, environmental changes, or insufficient sensory information. This creates a basis for the implementation of intelligent control systems in industrial facilities, as well as for further automation of technical diagnostics or optimization of the operation of microclimate systems.

The results obtained can become the basis for the creation of adaptive microclimate control systems capable of operating in conditions of changes in the external environment and incomplete information. In practice, this can allow reducing energy consumption, increasing the stability of temperature regimes in rooms, avoiding emergency situations (for example, icing of the heat exchanger) and improving comfort for users.

From a theoretical point of view, the results demonstrate the effectiveness of using neural network approaches to modeling physical processes in technical systems, in particular those that are difficult to formalize using conventional analytical methods. The proposed

ISSN-L 2664-9969; E-ISSN 2706-5448

approach can be used as a template for building models in related industries, where data are fragmentary and system behavior is complex or nonlinear.

Conflict of interest

The authors declare that they have no conflict of interest regarding this research, including financial, personal, authorship or other, that could influence the research and its results presented in this article.

Financing

The study was conducted without financial support.

Data availability

The manuscript has linked data in the data repository [10].

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the presented work.

References

1. Asamoah, P. B., Shittu, E. (2025). Evaluating the performance of machine learning models for energy load prediction in residential HVAC systems. *Energy and Buildings, 334,* 115517. https://doi.org/10.1016/j.enbuild.2025.115517
2. Bundo, J., Tola, S., Daci, A. (2024). HVAC/R Systems Modelling: Assessing Mathematical Model for Gas Compressor. *Mathematical Modelling of Engineering Problems, 11 (9),* 2285–2292. https://doi.org/10.18280/mmep.110901
3. Yan, L., Chang, X., Wang, N., Zhang, L., Liu, W., Deng, X. (2024). Comparison of Machine Learning and Classic Methods on Aerodynamic Modeling and Control Law Design for a Pitching Airfoil. *International Journal of Aerospace Engineering, 2024 (1).* https://doi.org/10.1155/2024/5535800
4. Guo, W., Liang, S., He, Y., Li, W., Xiong, B., Wen, H. (2022). Combining EnergyPlus and CFD to predict and optimize the passive ventilation mode of medium-sized gymnasium in subtropical regions. *Building and Environment, 207,* 108420. https://doi.org/10.1016/j.buildenv.2021.108420
5. García Vázquez, C. A., Cotfas, D. T., González Santos, A. I., Cotfas, P. A., León Ávila, B. Y. (2024). Reduction of electricity consumption in an AHU using mathematical modelling for controller tuning. *Energy, 293,* 130619. https://doi.org/10.1016/j.energy.2024.130619
6. Chaudhary, G., Johra, H., Georges, L., Austbø, B. (2025). Transfer learning in building dynamics prediction. *Energy and Buildings, 330,* 115384. https://doi.org/10.1016/j.enbuild.2025.115384
7. Afram, A., Janabi-Sharifi, F., Fung, A. S., Raahemifar, K. (2017). Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems: A state of the art review and case study of a residential HVAC system. *Energy and Buildings, 141,* 96–113. https://doi.org/10.1016/j.enbuild.2017.02.012
8. Zheng, G., Jia, R., Yi, W., Yue, X. (2025). Multi-step fusion model for predicting indoor temperature in residential buildings based on attention mechanism and neural network. *Journal of Building Engineering, 102,* 112057. https://doi.org/10.1016/j.jobe.2025.112057
9. Lopatko, O., Mykytyn, I. (2016). Neural networks as a tool for the temperature value prediction using transition process. *Measuring Equipment and Metrology, 77,* 65–70. https://doi.org/10.23939/istcmtm2016.77.065
10. *ROZROBKA MODELI SYSTEMY VENTYLYATSIYI TA KONDYTSIONUVANNYA DLYA REHULYUVANNYA TEMPERATURY TA VOLOHOSTI ZA DOPOMOHOYU NEYRONNYKH MEREZH NA OSNOVI ZIBRANYKH DANYKH DLYA PROTSESU AVTOMATYZATSIYI KERUVANNYA MIKROKLIMATOM.* Available at: https://drive.google.com/drive/folders/1avfqXfqOSdkn6ZrbnP97oymGjdkCd_Hd?usp=sharing Last accessed: 14.04.2025

✉ ***Illia Velychko,*** *Department of Automation and Computer Technologies of Control Systems named after Prof. A. P. Ladanyuk, National University of Food Technologies, Kyiv, Ukraine, e-mail: velychkoiv@nuft.edu.ua, ORCID: https://orcid.org/0009-0001-8826-4419*

Viktor Sidletskyi, *PhD, Associate Professor, Department of Automation and Computer Technologies of Control Systems named after Prof. A. P. Ladanyuk, National University of Food Technologies, Kyiv, Ukraine, ORCID: https://orcid.org/0000-0003-2606-3651*

✉ *Corresponding author*