Date of publication DEC-31, 2023, date of current version DEC-11, 2023.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.22.4.3358

Energy Consumption of Methods for Pattern Recognition using Microcontrollers

OLEKSANDR OSOLINSKYI[1], KHRYSTYNA LIPIANINA-HONCHARENKO[1], VOLODYMYR KOCHAN[1], ANATOLIY SACHENKO[1,2], DIANA ZAHORODNIA[1]
[1]West Ukrainian National University, 11 Lvivska St, Ternopil, 46009, Ukraine
[2]Kazimierz Pulaski University of Technology and Humanities in Radom, ul. Malczewskiego 29,26-600 Radom, Poland

Corresponding author: Oleksandr Osolinskyi (e-mail: oso@wunu.edu.ua).

ABSTRACT This paper presents the study of energy consumption of the methods for recognizing patterns/anomalies in numerical series, namely, the light sensor values in a smart home system. Methods for analyzing time series, identifying anomalous zones, and testing anomaly recognition algorithms are presented, and the smart system is prototyped. The energy consumption of correlation, comparison, and recognition methods using NNs is measured and analyzed. The case study has confirmed that the most resistant to signal changes and interference is the correlation analysis method. A methodology for applying recognition algorithms for different strategies for using optimal energy consumption is presented.

KEYWORDS energy consumption; microcontrollers; recognition algorithms; recognizing patterns; anomalies detection.

I. INTRODUCTION AND RELATED WORK

NOWADAYS, systems that measure various physical values, are growing. The most popular concept among them is the Internet of Things (IoT). Today, there are approximately 13 billion of such devices [1]. Unlike previous years, it became common to shift data computing from computer systems to small-sized, energy-efficient microcontroller units (MCUs) [2]. This trend has also affected many areas, for example, anomaly analysis of sensor signals [3] and pattern recognition [4]. However, the analysis confirms that developers of such systems pay just a little attention to the energy efficiency of algorithms for pattern and anomaly analysis recognition. On the other side, researchers who analyze and study the energy consumption of instructions and commands by MCU (to increase the battery life), do not consider to a sufficient extent where and how the results of instruction energy consumption will be used [5, 6].

Therefore, it remains an urgent question not only to measure the power consumption of the MCU but also to define what tasks can be running so that the MCU can be working in autonomous mode for the longest time.

Due to the need for a comprehensive approach to the addressed issue let us to divide the related works into two groups: analysis of numerical series for anomalies detection and analysis of MCU instructions energy consumption.

A. METHODS AND TOOLS OF ANALYZING NUMERICAL SERIES FOR ANOMALIES DETECTION

During operation, any measuring system collects and processes the sensor signals and sends the data to a server or stores it in internal memory. If this data is accompanied by the acquisition time, then a numerical series of values of measured physical value is formed. An example of a physical value can be temperature, light level, or the vibration level of an electric motor under extreme loads. Methods for finding patterns in a numerical series help to understand better the processes that take place in the measurement object, their cause, and further development. One of the most used methods for analyzing numerical series is correlation [7, 8].

In [9], a technique for visual recognition of numerical series was proposed and the algorithm for image classification was described. This approach simplifies the analysis of numerical series because the image is recognized as a whole, unlike the methods using Markov chains [10].

In [11], the authors proposed a speech recognition algorithm that does not depend on the programming language and can be used for low-energy computer systems based on MCUs. The proposed algorithm achieves a high accuracy of speech recognition, which is an analog signal converted into a certain numerical series, using a relatively small set of patterns.

In [12], the approach for using a neural network (NN) in systems based on the cheap MCUs was proposed. The technology consists of two parts. Its first intelligent part is created in Matlab-Simulink, where the NN is trained. Then the trained structure of NN is transferred to the MCU as part of its program. As shown in [13], the computing resources required to use the NN are about 10,000 times less than those required to train it. Moreover, it is possible to transfer each layer to a separate MCU, which allows the creation of a high-performance control system without using the expensive equipment.

An even more effective speed solution with a small number of NN inputs was proposed in [14]. Matlab-Simulink calculates the values of the NN for all possible combinations of its inputs and records it to the MCU's extended RAM at the addresses corresponding to the code combinations at the NN inputs. Then the MCU gets from the memory the only value that corresponds to the desired combination of NN inputs.

However, this solution may not be optimal in terms of energy consumption, because the MCU's extended RAM consumes the energy too.

Therefore, a goal of the paper is to investigate the performance and energy consumption of MCUs when using common correlation and comparison methods as well as NN methods for recognizing patterns/anomalies in numerical series.

B. ANALYSIS OF MCU ENERGY CONSUMPTION

The most common technique for measuring the MCU energy consumption (as well as IoT modules) is the discharge control circuit using a voltmeter for measuring the voltage of the MCU power supply, and an ammeter for measuring the current consumption. However, since MCUs are manufactured using CMOS technology, they consume current in short pulses synchronous with the clock generator [15, 16]. Therefore, the series ammeter inclusion in the MCU energy supply circuit impact on active resistance but also a much larger dynamic reactive resistance. This inductive resistance often leads to MCU failures due to a large voltage drop at the higher harmonics of the consumption current pulses. To avoid this problem, it is needed to include a capacitor in parallel to MCU power supply circuit.

In [17], it was proposed to study the power consumption of a microcomputer using special benchmarks. The results show that hardware-based video processing has higher performance, but also increases power consumption. If we use a software implementation, then video processing has less power consumption, but the processing time increases.

In [18], the energy consumption of two popular platforms, Arduino and Raspberry Pi, was analyzed as part of an air quality control system. The results were received without energy consumption of the software and working modes. These measurement results show that although both platforms are positioned as low-current, the Arduino consumes almost three times less energy than the Raspberry Pi, which has significantly more computing resources.

In [19], a simple and inexpensive wattmeter for the investigation of FPGAs and MCUs was presented. It can measure energy consumption both at the level of processor instructions and at the level of system components. According to the authors' claim, this device confirms the data of equipment manufacturers, which indicates the high accuracy of the latter and has an error of about 5%, the error depends on the code being executed or on the set of components.

The analysis of works above show that some researchers investigate the implementation of high-performance data processing methods while others deal with measuring the MCU energy consumption of executing code/instructions.

Therefore, a very little attention is paid to the study of computing methods energy efficiency and program optimization in terms of energy consumption. The reason of this is that this problem is not within the interest's scope of both researchers' groups. This paper proposes to fill this gap and to study the energy efficiency for analyzing numerical series.

In our case, we have for prototype the system of growing plants in a closed space GrowBox [20]. It automatically controls the best conditions for plant growth. All studies were conducted with an open prototype (Fig. 1).

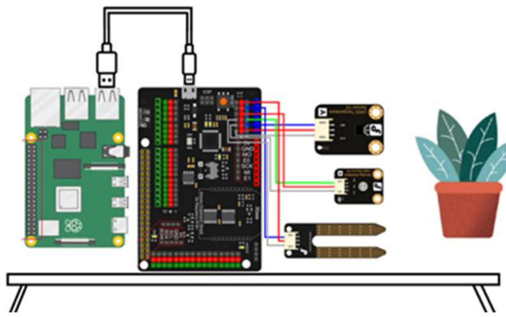
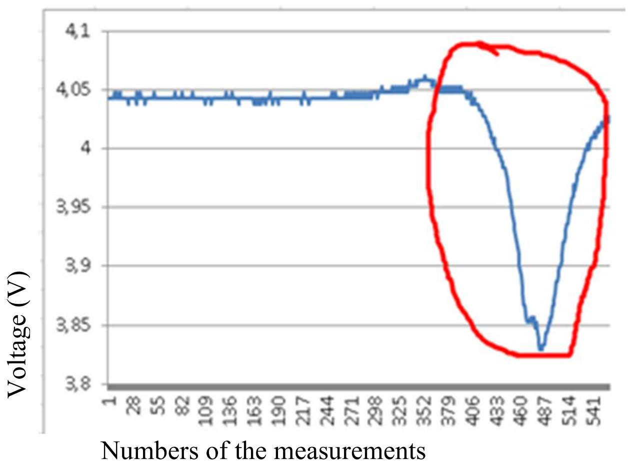


Figure 1. The investigated GrowBox system for growing plants in a closed space.

When testing the system, it has been found that the light sensor reacts to changes in lighting when a person passes by (Fig. 2). As it can be seen in Fig. 2, the waveform is almost unchanged and it does not depend on the illumination level. This feature was used in the microclimate control system to expand the functions for detecting the people in the room. In particular, the modernization does not require any hardware changes. It only requires the software and supplementing with the analysis of numerical series and detection of specified anomalies. Since the lighting in a room changes considerably throughout the day, the method of detecting anomalies by comparing them to a predefined threshold does not allow reliably to recognize this anomaly.

To solve this issue, the three approaches are selected: a) recognition by an artificial neural network (Perceptron); b) correlation method; c) comparison method.



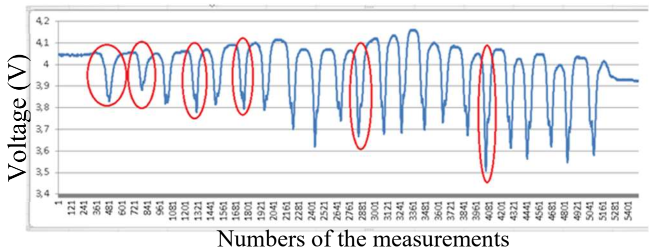


Figure 2. Changes in the output signal of the light sensor when a person passes by

II. ANOMALIES DETECTION IN NUMBER SERIES

Based on the preliminary analysis of methods for data processing authors selected clustering technique [1, 2]. The latter one can sort similar data, revealing unusual or outlier groups of points in a series, which can be useful in finding unusual patterns in energy-efficient data processing techniques using microcontrollers.

To solve the given problem, we need to identify the two clusters: a rest period (no anomaly) and an anomaly period. That is why we select the non-hierarchical clustering, which requires that the initial distribution/number of clusters is known a priori. In non-hierarchical clustering, the data points are divided into k clusters so that the variation inside the cluster is as small as possible for all the clusters. It is expedient here to employ the k-means method [3, 21, 22], which is the most common non-hierarchical clustering approach [23, 24], due to its simplicity and effectiveness in detecting clusters and anomalies in time series. In such case we may run the following steps [25]:

Selecting the k-means method:

Step 1. Determine the k number of clusters.

Step 2. Determine the initialization by randomly assigning each sample to one of the k clusters.

Step 3. Repeat the previous two steps until the cluster assignment stops changing:

Step. 3.1 Calculating the centroid V per each cluster [26, 27]

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} \left| X_j - \mu_i \right|^2, \tag{1}$$

where $i=1,2,....,k$ and μi is the centroid or midpoint of all points of the cluster Si.

Step. 3.2. Assign/reassign each sample to the cluster with the nearest centroid.

Step 4. Display the result of data distribution by clusters.

As input data, we used data presented in the form of a time series. As the result, Fig. 3 illustrates a distribution of clusters, namely, cluster 0 contains anomalies, and cluster 1 contains data during the rest period. As it can be seen in Fig. 3, clusters are detected even with the minor fluctuations in indicators.

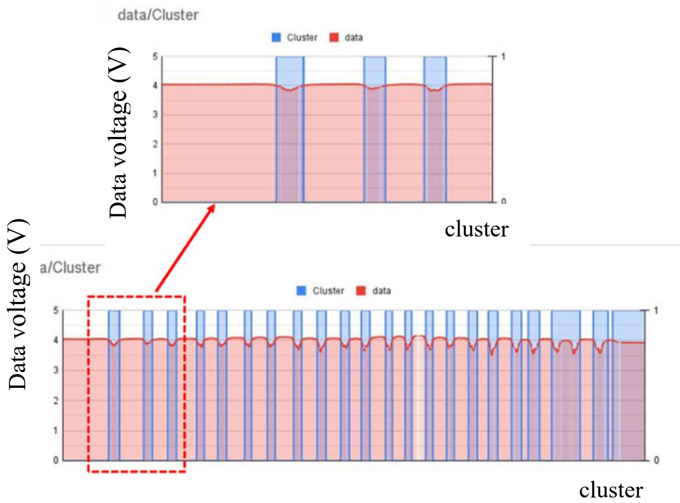


Figure 3. Anomaly detection by cluster analysis

III. SOFTWARE IMPLEMENTATION OF ANOMALY DETECTION METHODS USING LabView

Before uploading the executable code using one of the three specific methods mentioned above, we developed the software using LabView [28]. This environment was selected because it can be employed to rapidly design the systems for data acquisition and analysis. Moreover, the LabView has the built-in virtual tools for working with the Arduino platform, which simplifies development and saves the research time.

As mentioned above, the NN, namely a single-layer perceptron, was used as the first method of anomaly detection [29]. We selected the googlecolab service [30] to implement the perceptron and train it. This service was used separately from the main program for analyzing recognition methods and its main task was to output the weighting coefficients of the trained NN.

The block diagram of the code that implements the perceptron model is given in Fig. 4.a. The two curves are shown in graphical user interface (Fig 4.b). The first curve (see Fig 4.b, top) illustrates how the neural network is trained, and the second one (see Fig 4.b, bottom) shows the test signal that has been recognized.

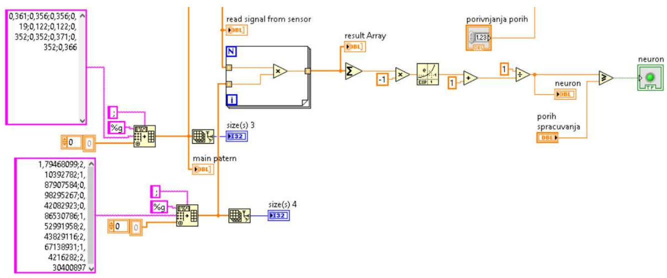


Figure 4.a Block diagram of code for NN implementing.

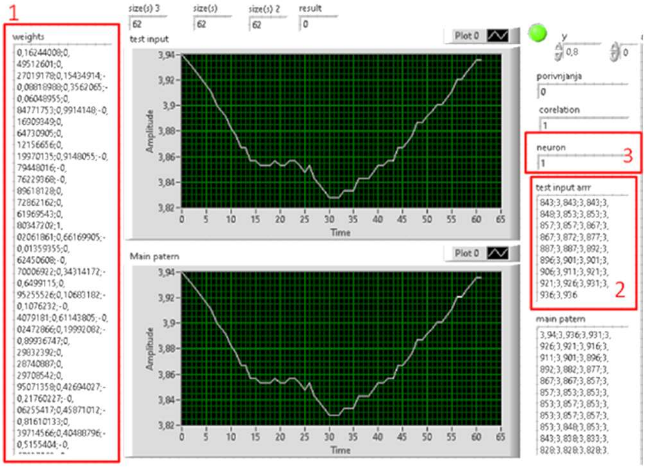


Figure 4.b Graphical user interface for NN implementing 1- weighting coefficients of NN, 2- input signal from the sensor and 3 – results of the NN running

To implement the correlation method [31], the code was developed (Fig. 5.a).The graphical user interface (Fig. 5.b) illustrates the two curves. The upper graph shows the received signal from the sensor (65 measurements). The lower graph shows a set of sample values that are used as a reference

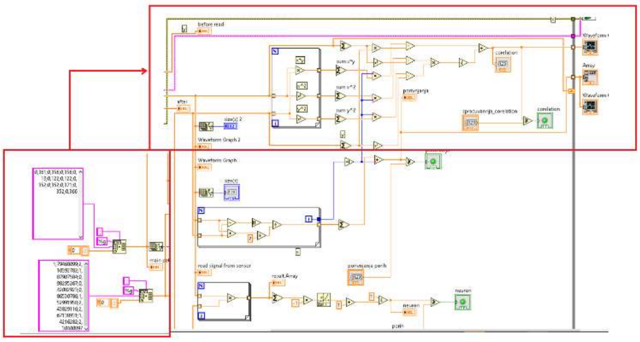


Figure 5.a Block diagram of code for correlation method implementing

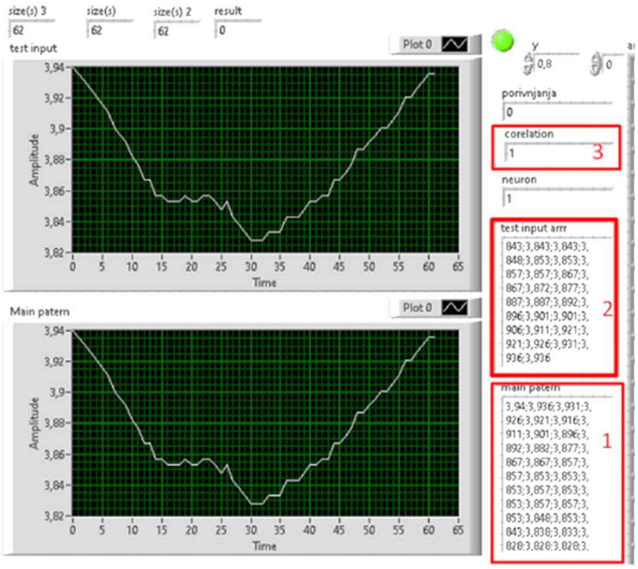


Figure 5.b Graphical user interface for correlation method implementing: 1- the main signal pattern, 2 – the input signal under test and 3- the result of calculating the correlation coefficient r.

To implement the comparison method [32], the code was developed for determining the deviation from the arithmetic mean (Fig. 6.a).

The graphical user interface (Fig. 6.b) illustrates the same two curves as in Fig 4.b.

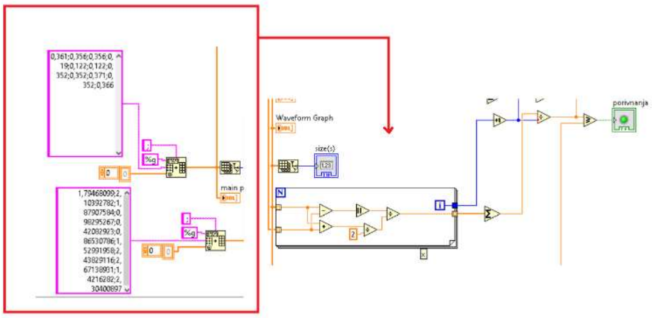


Figure 6.a Block diagram of the code for comparison method implementing.

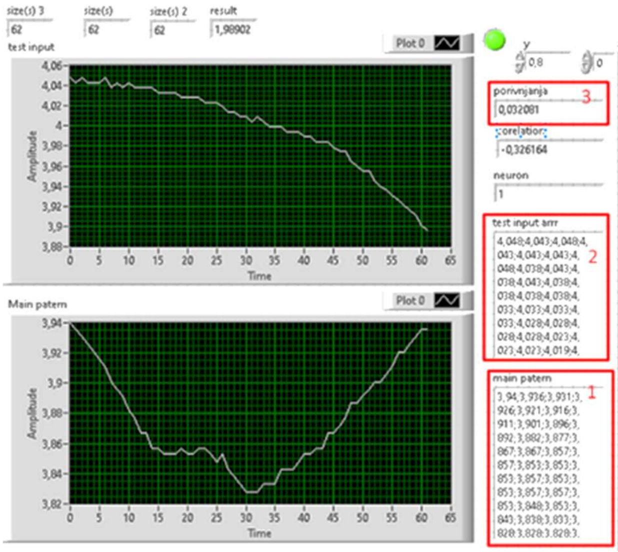


Figure 6.b Graphical interface for comparison method implementing:1 – the main signal pattern, 2- the input signal under test, and 3 – the deviation from the arithmetic mean.

After checking all the anomaly recognition methods above, it has been found that the correlation method has the best results. This method is stable even when the light level changes, but the pattern remains the same.

Then authors investigated the anomaly recognition methods for different ration of input and sample signals. In particular, Fig. 7 shows a comparison of input and sample signals when they have the same amplitude and shape. The curve at the top means the real signal from the sensor, and the curve at the bottom means that the signal is used as a reference. The test results are labeled with the following numbers (see Fig. 7, right):

1) comparison method. Zero deviation means a complete match, for example, 0.2 is a deviation of 20%;

2) correlation method. One means that two signals are 100% correlated with each other;

3) recognition using NNs. One means that the root mean square error tends to be zero.

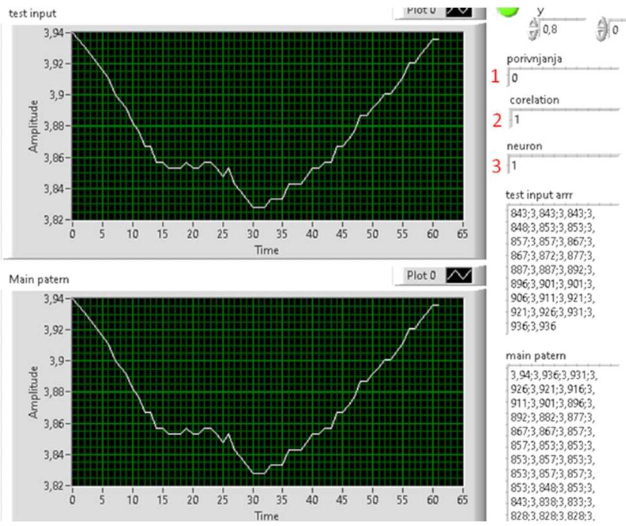


Figure 7. Comparison of the input and sample signal.

Fig. 8 illustrates the effect of amplitude deviations of +/-20%. When the signal amplitude was increased by 20% (see the upper graph of Fig. 8.a), the NN and the correlation method showed stability and the anomaly recognition remained within acceptable limits. When the amplitude was reduced by 20%, similar results were obtained (see the upper graph of Fig. 8.b). The comparison method showed deviations of about 20%, respectively, although the waveform remained unchanged. Thus, when the difference in the amplitude of the input and sample signals increases, the comparison method stops working.

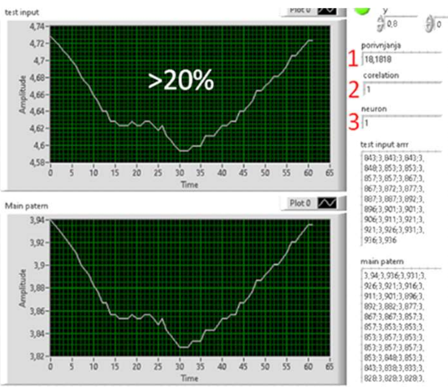


Figure 8.a. Dependence of signal on the amplitudes increase amplitude by 20%

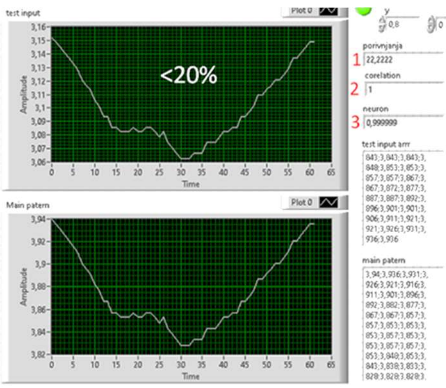
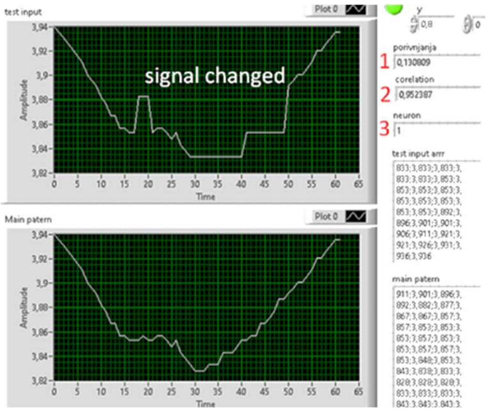


Figure 8.b. Dependence of signal on the amplitudes reduce amplitude by 20%: 1 – the result of the comparison method, 2 – the result of correlation, 3 – the result of recognition using a neural network

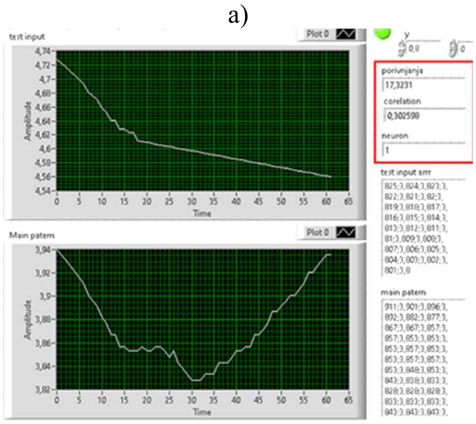
Fig. 9 shows the response of methods for comparison, correlation, and NN to changes in the input signal. For example, the top graph in Fig. 9, *a* shows some signal distortion. As a result, there are deviations of methods: comparison – 13% (label 1), correlation – 0.95 (label 2) and the NN – 1 (label 3).

It should be noted that label 1 in the NN window shows the recognition probability rounded to an integer. In fact, the root mean square error is slightly less than 50%.

Thus, with different types of input signals, the correlation method remains the most accurate in detecting anomalies. So, it can be used in portable and autonomous systems, such as IoT modules or low cost measurement systems of different applications.



a)



b)

Figure 9. Response of comparison, correlation, and NN methods to changes in input signal: a) slight change in the signal; b) significant change in the signal

IV. ANALYSIS OF ENERGY CONSUMPTION FOR ANOMALY RECOGNITION METHODS

Having the tests results, we may estimate the energy efficiency of the proposed methods because it is expected to employ them as part of IoT modules that operate autonomously from the battery. Extending the life of the module, the cost of its maintaining in a remote location will be reduced.

The following recommendations were used for preliminary measurements [33] and consisted of the MCU under test, a precision multimeter [34], and a PC for data collection and analysis.

This system for preliminary measurements includes the multimeter with the MCU under test and a PC for analysis. The communication between the PC and the multimeter uses an infrared interface, which reduces the influence of interference from the PC.

After completing measurements by the 3 developed methods for anomaly detection, the following results were obtained (Fig. 10).

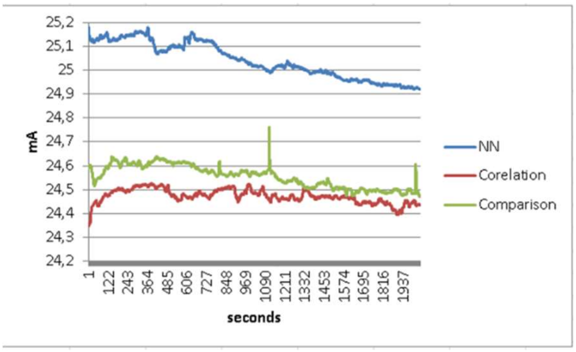


Figure 10. Graphical interface for working with the user

As it can be seen in Figure 10, the NN method is the best one, with an average consumption of 25.042 mA, followed by the comparison method with 24.554 mA correspondingly. The lowest energy consumption was shown by the correlation method at 24.471 mA.

V. CONCLUSIONS

This work demonstrates a possibility of using anomaly/pattern recognition methods in the IoT modules working as part of low cost measurement systems.

Received results have confirmed that the most resistant to signal changes and interference is the correlation analysis method. Changes in signal amplitude and shape by 20% did not affect the accuracy and were in the range of 1. The analysis of the methods' power consumption showed that the correlation analysis method is also the least power-consuming, with an average of 24.471 mA.

The NN method was the most power-consuming, averaging 25.042 mA. In addition, this method provided a lower recognition accuracy compared to the correlation analysis method, 1 vs. 0.98. This can be explained by the fact that the simplest perceptron was studied as an NN, and maybe more complex NNs can have the better recognition accuracy.

The comparison method confirmed the average power consumption of 24.554 mA, but it was very sensitive to changes in signal amplitude and shape.

Thus, the correlation analysis method can be recommended for use in low-cost measurement modules that are a part of the IoT for anomaly recognition.

In addition, the research shows that there are deviations in power consumption, but they are insignificant due to the fact that a small dataset is used.

In terms of autonomy, the correlation and comparison methods gave approximately 31-31.5 days, and the NN method gave 29-30 days of battery life.

In future, authors are going to explore machine learning methods and various types of neural networks [26, 35] for a more detailed analysis in energy efficiency terms and power consumption optimization of autonomous MCU units in IoT.

References

[1] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, "A review on outlier/anomaly detection in time series data," *ACM Comput. Surv.*, vol. 54, issue 3, Article 56, 33 pages, 2021. https://doi.org/10.1145/3444690.

[2] A. S. Alghawli, "Complex methods detect anomalies in real time based on time series analysis," *Alexandria Engineering Journal*, Elsevier, vol. 61, issue 1, pp. 549–561, 2022. https://doi.org/10.1016/j.aej.2021.06.033.

[3] K. P. Sinaga and M.-S. Yang, "Unsupervised k-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80716-80727, 2020, https://doi.org/10.1109/ACCESS.2020.2988796.

[4] E. Garcia-Breijo, et al., "A comparison study of pattern recognition algorithms implemented on a microcontroller for use in an electronic tongue for monitoring drinking waters." *Sensors and Actuators A: Physical*, vol. 172, issue 2, pp. 570-582, 2011. https://doi.org/10.1016/j.sna.2011.09.039.

[5] M. Al-Kofahi, M. Al-Shorman, & O. Al-Kofahi, Osameh, "Toward energy efficient microcontrollers and IoT systems," Preprint, pp. 1-23, 2019, https://doi.org/10.1016/j.compeleceng.2019.106457.

[6] L. Lin, et al., "Power management in low-power MCUs for energy IoT applications," *Journal of Sensors*, vol. 2020, pp. 1-12, 2020. https://doi.org/10.1155/2020/8819236.

[7] R. Krishnamurthi, A. Kumar, D. Gopinathan, A. Nayyar, B. Qureshi, "An overview of IoT sensor data processing, fusion, and analysis techniques," *Sensors*, vol. *20*, article 6076, 2020. https://doi.org/10.3390/s20216076.

[8] M. D. Ahmad, et al., "Lux meter integrated with internet of things (IoT) and data storage (LMX20)," *Proceedings of the 2021 IEEE International Conference in Power Engineering Application (ICPEA)*, 2021. https://doi.org/10.1109/ICPEA51500.2021.9417762.

[9] P. A. Waghmare and J. V. Megha, "Efficient pattern recognition in time series data," *Proceedings of the 2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, 2018, pp. 436-441, https://doi.org/10.1109/ICIRCA.2018.8597250.

[10] A. Wilinski, "Time series modeling and forecasting based on a Markov chain with changing transition matrices," *Expert Systems with Applications*, vol. 133, pp. 163-172, 2019, https://doi.org/10.1016/j.eswa.2019.04.067.

[11] A. Udal, A. Riid, M. Jaanus, K. Parnamets and M. Lokuta, "Development and testing of a compact voice command recognition algorithm for limited complexity microcontroller devices," *Proceedings of the 2018 22nd International Conference Electronics*, 2018, pp. 1-5, https://doi.org/10.1109/ELECTRONICS.2018.8443645.

[12] K. Dokic, B. Radisic, & M. Cobovic, "MicroPython or Arduino C for ESP32-efficiency for neural network edge devices," *Proceedings of the International Symposium on Intelligent Computing Systems*, March 2020, pp. 33-43). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-43364-2_4.

[13] X. Xu, Y. Ding, S. X. Hu, et al., "Scaling for edge inference of deep neural networks," *Nat Electron*, vol. 1, pp. 216–222, 2018. https://doi.org/10.1038/s41928-018-0059-3.

[14] M. Komar, V. Kochan, L. Dubchak, S. Bezobrazov, I. Romanets, "High performance adaptive system for cyber attacks detection," *Proceedings of the 2017 IEEE 9th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS'2017*, 2017, vol. 2, pp. 853–858. https://doi.org/10.1109/IDAACS.2017.8095208.

[15] M. Malinowski, M. Kampik, and K. Musioł, "Software for automation of measurements with Keysight E4980A LCR meter," *Measurement Systems in Research and in Industry*, Proceedings of the 13th Scientific Conference, 2020.

[16] V. Konstantakos, A. Chatzigeorgiou, S. Nikolaidis and T. Laopoulos, "Energy consumption estimation in embedded systems," *Proceedings of the 2006 IEEE Instrumentation and Measurement Technology Conference Proceedings*, Sorrento, Italy, 2006, pp. 235-238, doi: 10.1109/IMTC.2006.328405. https://doi.org/10.1109/IMTC.2006.328405.

[17] Q. He, B. Segee and V. Weaver, "Raspberry Pi 2 B+ GPU power, performance, and energy implications," *Proceedings of the 2016 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2016, pp. 163-167, https://doi.org/10.1109/CSCI.2016.0038.

[18] C. A. Okigbo, A. Seeam, S. P. Guness, X. Bellekens, G. Bekaroo, V. Ramsurrun, "Low-cost air quality monitoring: comparing the energy consumption of an Arduino against a raspberry Pi based system," *Proceedings of the 2nd International Conference on Intelligent and Innovative Computing Applications ICONIC'20*, September 2020, Article no. 36, pp. 1–8, https://doi.org/10.1145/3415088.3415124.

[19] M. Al-Shorman, M. Al-Kofahi, & O. Al-Kofahi, "A practical microwatt-meter for electrical energy measurement in programmable devices," *Measurement and Control*, vol. 51, 002029401879435, 2018. https://doi.org/10.1177/0020294018794350.

[20] V. Turchenko, I. Kit, O. Osolinskyi, D. Zahorodnia, P. Bykovyy and A. Sachenko, "IoT based modular grow box system using the AR," *Proceedings of the 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T)*, 2020, pp. 741-746, https://doi.org/10.1109/PICST51311.2020.9467949.

[21] P. Fränti, & S. Sieranoja, "How much can k-means be improved by using better initialization and repeats?" *Pattern Recognition*, *Elsevier*, vol. 93, pp. 95–112, 2019. https://doi.org/10.1016/j.patcog.2019.04.014.

[22] C. Yuan, H. Yang, "Research on k-value selection method of k-means clustering algorithm," *Scientific Multidisciplinary Journal MDPI*, vol. 2, issue 2, pp. 226-235, 2019. https://doi.org/10.3390/j2020016.

[23] A. Kanavos, S. A. Iakovou, S. Sioutas, V. Tampakas, "Large scale product recommendation of supermarket ware based on customer behaviour analysis," *Big Data and Cognitive Computing*, vol. 2, issue 2, 11, 2018. https://doi.org/10.3390/bdcc2020011.

[24] P. Fränti, S. Sieranoja, "K-means properties on six clustering benchmark data sets," *ApplIntell, Springer*, vol. 48, pp. 4743–4759, 2018. https://doi.org/10.1007/s10489-018-1238-7.

[25] H. Lipyanina-Goncharenko, V. Brych, S. Sachenko, T. Lendyuk, P. Bykovyy, & D. Zahorodnia, Method of Forming a Training Sample for Segmentation of Tender Organizers on Machine Learning Basis. Proceedings of the Proceedings of the 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2021). Volume I: Main Conference Lviv, Ukraine, April 22-23, 2021, pp. 1843-1852. [Online]. Available at: https://ceur-ws.org/Vol-2870/paper134.pdf.

[26] A. Dovbysh, V. Liubchak, I. Shelehov, J. Simonovskiy, A. Tenytska, "Information-extreme machine learning of a cyber attack detection system," *Radioelectronic and Computer Systems*, no. 3, 2022, pp. 121-131, https://doi.org/10.32620/reks.2022.3.09.

[27] N. Malekar, "Survey of PCA, DWT, k-means and novel k- means algorithm for image processing," 2014. [Online]. Available at: https://api.semanticscholar.org/CorpusID:219571757.

[28] H. Alshareefi, C. Lupu, L. Ismail, & L. D. Luu, "The design and testing of a neural controller based on artificial neural network theory using LABVIEW facilities," *UPB Sci. Bull. Ser. C Electr. Eng. Comput. Sci.*, vol. 83, issue 3, pp. 35-46, 2021.

[29] S. Chakraverty, D. M. Sahoo, N. R. Mahato, "Perceptron learning rule," *Concepts of Soft Computing: Fuzzy and ANN with Programming*, Springer, Singapore, pp. 183-188, 2019. https://doi.org/10.1007/978-981-13-7430-2_13.

[30] W. Vallejo, C. Díaz-Uribe, & C. Fajardo, "Google colab and virtual simulations: practical e-learning tools to support the teaching of thermodynamics and to introduce coding to students," *ACS Omega*, vol. 7, issue 8, pp. 7421-7429, 2022. https://doi.org/10.1021/acsomega.2c00362.

[31] S. Chatterjee, "A new coefficient of correlation," *Journal of the American Statistical Association,* 116.536, pp. 2009-2022, 2021. https://doi.org/10.1080/01621459.2020.1758115.

[32] S. Khalilpourazari, and S. H. R. Pasandideh, "Modeling and optimization of multi-item multi-constrained EOQ model for growing items," *Knowledge-Based Systems*, vol. 164, pp. 150-162, 2019. https://doi.org/10.1016/j.knosys.2018.10.032.

[33] S. Sebastian, "A survey about power consumption for Arduino," *Carpathian Journal of Electrical Engineering*, vol. 14, issue 1, pp. 105-109, 2020.

[34] Digital multimeter; USB; LCD; (40000); Bargraph: 40segm.10x/s Manufacturer part number: UT71E. [Online]. Available at: https://www.tme.eu/en/details/ut71e/portable-digital-multimeters/uni-t/

[35] N. Lutsiv, T. Maksymyuk, M. Beshley, A. Sachenko, L. Vokorokos, J. Gazda, "Deep semisupervised learning-based network anomaly detection in heterogeneous information systems," *Computers, Materials and Continua*, vol. 70, issue 1, pp. 413–431, 2021. https://doi.org/10.32604/cmc.2022.018773.

OLEKSANDR OSOLINSKYI, ***an Associate Professor, Department for Information Computing Systems and Control, West Ukrainian National University, Ternopil, Ukraine.***

Research interests: Computer networks, Microprocessor systems, Embedded Systems, SCADA, IoT, Distributed Systems, High Performance Computing, Operating Systems.

KHRYSTYNA LIPIANINA-HONCHARENKO, ***an Associate Professor at Department of Information Computer System and Control, West Ukrainian National University, Ternopil, Ukraine.***

Research interests: machine learning, cloud technologies, data mining, modern information technologies, corporate information technologies, information systems and technologies in jurisprudence, digital economy, etc.

VOLODYMYR KOCHAN, ***a Professor at Department of Information Computer System and Control, West Ukrainian National University, Ternopil, Ukraine.***

Research interests: Microcontroller's temperature measurement, distributed sensor network etc.

ANATOLY SACHENKO, ***a Professor, Research Institute for Intelligent Computer Systems, Director, West Ukrainian National University, Ternopil, Ukraine.***

Research interests: Distributed Sensor Networks, Artificial Intelligence Application, Image Processing, Wireless Sensor Networks, Project Management etc.

DIANA ZAHORODNIA, ***an Associate Professor at Department of Information Computing Systems and Control, West Ukrainian National University, Ternopil, Ukraine.***

Research interests: Information technologies, image processing and video surveillance systems, optimization methods.

• • •