https://doi.org/10.31891/csit-2024-2-10
УДК 005.21:005.8:004.8.

Sergii LYSENKO, Yevheniia SHPULIAR
Khmelnytskyi National University

METHOD FOR CREATING SVM CLASSIFIER FOR DATA ANALYSIS ON FPGA

The paper explores the use of SVM classifier method for data analysis on FPGA, which, despite its effectiveness, may face challenges related to limited resources and data processing speed. In this context, there is a need to develop new methods for integrating SVM classifiers with high-performance computing hardware. The increasing demand for speed and energy efficiency requires new approaches to implementing machine learning methods. One of the key tools for data classification and analysis is the Support Vector Machine (SVM), widely used in business, science, medicine, and many other fields. Developing an efficient and optimized method for creating SVM classifiers for FPGA requires further research and development, as existing methods may be suboptimal in terms of speed and FPGA resource utilization. The article provides an overview of known hardware solutions to this problem, proposed in the current scientific literature. Additionally, the effectiveness of combining hardware and software components to achieve significant acceleration of the data analysis process is discussed. The article emphasizes the need for further research and improvement to fully realize the transformative potential of machine learning classification methods.

The work resulted in the development of a new, specialized, and optimized hardware accelerator based on FPGA for the Support Vector Machine (SVM) method using convex optimization (CO) on embedded platforms. The proposed embedded architectures are designed to be universal, parameterized, and scalable. This means that these embedded solutions can accommodate different datasets of varying sizes and can be implemented on various embedded platforms, including those equipped with the latest FPGAs. They are also capable of handling both linear and nonlinear discrimination across multidimensional datasets.

Keywords: SVM classifier, machine learning, convex optimization, FPGA, data analysis methods, data classification, support vector machine.

Сергій ЛИСЕНКО, Євгенія ШПУЛЯР
Хмельницький національний університет

МЕТОД СТВОРЕННЯ SVM-КЛАСИФІКАТОРА ДЛЯ АНАЛІЗУ ДАНИХ НА БАЗІ FPGA

В роботі досліджено використання методу створення SVM-класифікатора для аналізу даних на базі FPGA, що незважаючи на свою ефективність, може стикатися з викликами, пов'язаними з обмеженими ресурсами та швидкодією обробки даних. У цьому контексті виникає необхідність розвитку нових методів інтеграції SVM-класифікаторів з високопродуктивним обчислювальним обладнанням. Зростаюча потреба у швидкодії та енергоефективності вимагає нових підходів до реалізації методів машинного навчання. Одним з ключових інструментів для класифікації та аналізу даних є метод опорних векторів (SVM - Support Vector Machine), який широко використовується у бізнесі, науці, медицині та багатьох інших галузях. Розробка ефективного та оптимізованого методу створення SVM-класифікатора для FPGA вимагає додаткових досліджень та розробки, оскільки існуючі методи можуть бути неоптимальними з точки зору швидкодії та використання ресурсів FPGA. У статті представлено огляд відомих апаратних рішень даної задачі, запропонованих у сучасній науковій літературі. Крім того, обговорюється ефективність поєднання апаратних та програмних компонентів для досягнення значного прискорення процесу аналізу даних. Стаття підкреслює необхідність подальших досліджень та вдосконалення для розкриття трансформаційного потенціалу методу класифікації машинного навчання.

Результатом роботи стала розробка нового, спеціалізованого та оптимізованого апаратного прискорювача на базі ПЛІС для методу опорно-векторних обчислень з використанням опуклої оптимізації на вбудованих платформах. Запропоновані вбудовані архітектури є універсальними, параметризованими та масштабованими. Це означає, що ці вбудовані рішення можуть працювати з різними наборами даних різного розміру і можуть бути реалізовані на різних вбудованих платформах, включаючи ті, що оснащені новітніми ПЛІС. Вони також здатні обробляти як лінійну, так і нелінійну дискримінацію багатовимірних наборів даних.

Ключові слова: SVM-класифікатор, машинне навчання, опукла оптимізація, FPGA, методи аналізу даних, класифікація даних, метод опорних векторів.

Introduction

In today's world, the volume and complexity of data is growing, which requires effective methods of their analysis and processing. Machine learning, particularly classification methods, is becoming a key tool for automated data processing and analysis in fields ranging from medicine to finance [1-3]. However, the growing need for speed and energy efficiency requires new approaches to the implementation of these methods. One of the key tools for data classification and analysis is the method of support vectors (SVM - Support Vector Machine), which is widely used in business, science, medicine and many other fields [4-6].

The use of the SVM-classifier creation method for FPGA-based data analysis is an urgent problem, since FPGAs provide high speed and energy efficiency, which makes them attractive for application in the field of machine learning [2,4]. Despite its effectiveness, the implementation of the SVM classifier can face challenges related to limited resources and data processing speed. In this context, there is a need to develop new methods of integrating SVM classifiers with high-performance computing equipment [7, 8].

The study is devoted to the development of a method for creating an SVM classifier for data analysis based on FPGAs, known for their high speed and the possibility of parallel data processing, which makes them an ideal candidate for the effective implementation of machine learning classifiers.

Known methods of implementing FPGA classifiers

Methods for implementing classifiers on FPGAs involve various techniques and approaches to embedding classification algorithms on programmable logic gate arrays (FPGAs).

The [9] paper describes the implementation of an optimized SVM classifier on a modern FPGA platform, employing state-of-the-art design techniques to integrate it into a proposed device for real-time melanoma detection. The hardware implementation achieves impressive results, boasting a classification accuracy of 97.9% and a substantial speedup factor of 26 compared to equivalent software implementations on embedded processors. Moreover, it utilizes only 34% of available resources and consumes a mere 2 watts of power. These outcomes signify that the system fulfills crucial requirements for embedded systems, including high performance, cost-effectiveness, minimal resource usage, and energy efficiency, all while maintaining exceptional classification accuracy.

In [10] a specialized embedded SVM classifier tailored for an affordable handheld device intended for early melanoma detection in primary healthcare settings was performed. Authors introduced a hardware/software co-design approach aimed at deploying the SVM classifier onto an FPGA, enabling melanoma detection directly on the chip. Leveraging the latest UltraFast High-Level Synthesis design methodology on a modern hybrid FPGA platform (Zynq), our implementation achieves efficient melanoma classification on the chip.

In [11] authors proposed method employed in an SVM-based isolated digit recognition system, which undergoes examination using both speaker-dependent and multispeaker-dependent TI46 databases of isolated digits. Feature extraction is conducted using both LPC and MFCC methods. These extracted features are then subjected to dimensionality reduction via self-organized feature maps (SOFM), with the resulting mapped features utilized by the SVM classifier to assess recognition accuracy employing various kernels.

A method of creating an SVM classifier for data analysis based on FPGA

An enhanced method for developing an SVM classifier for data analysis is proposed. This method leverages the hardware capabilities of FPGAs, leading to reduced data analysis time, increased accuracy, and decreased software and hardware resource usage compared to traditional methods. All embedded hardware and software experiments were conducted on the DE1-SoC Board development platform, utilizing the Altera Cyclone V SoC 5CSEMA5F31C6 device.

The appearance of the Terasic DE1-SoC development board is shown in Figure 1 [12]. Figure 2 illustrates the system architecture for embedded hardware and software projects.

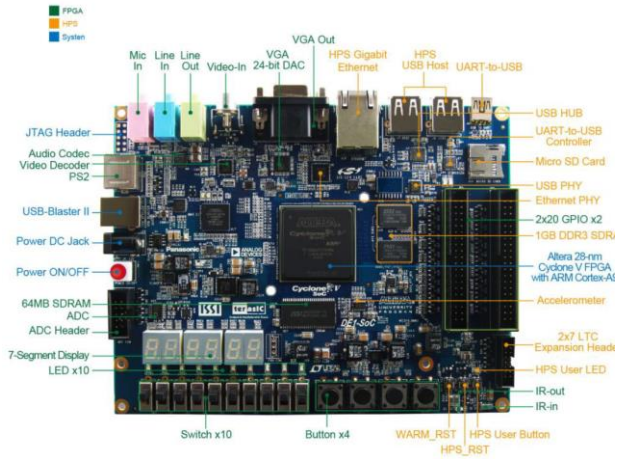

Fig. 1. Top view of the DE1-SoC development board

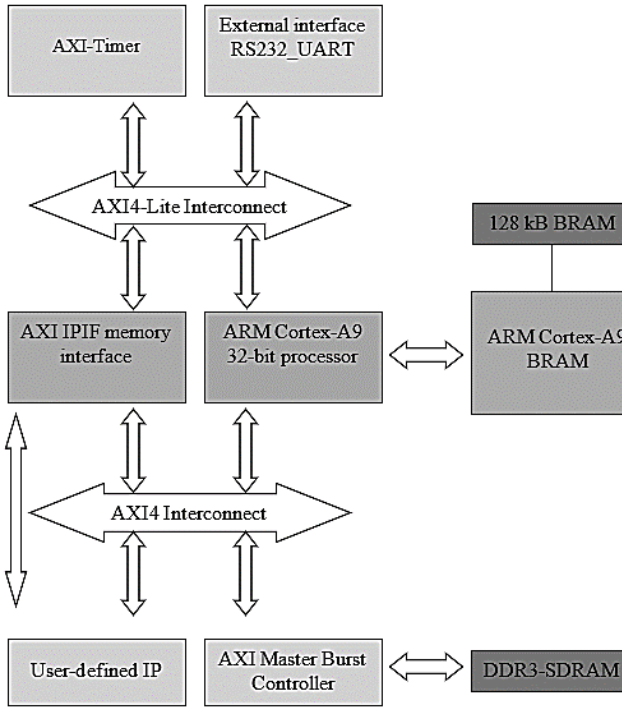

Fig. 2.System-level architecture

One of the objectives of this design was to develop a system-level architecture capable of training and classifying a continuous data stream using the AXI4 burst/stream interface. In practical applications, such as

autonomous vehicles, this feature enables direct connection between custom IP hardware and a camera to process input data in real-time.

This allows the hardware IP to dynamically perform the procedure of training and testing to adapt to changing environments. Figure 2 demonstrates the presented AXI4-lite and AXI4 lines serve as the "glue" logic for entire system. The AXI4-lite interface is a one-time transactional bidirectional interface, with memory mapped to a specific memory region.

During preload mode (Figure 3), the custom logic module calculates the total number of bytes to load based on the SVM specifications provided to the ARM Cortex-A9 via work registers.

The next step is training. After that the weight vectors and offset values to DDR3-SDRAM for further computation and analysis are ready to use.

For developing the SVM algorithm based on convex optimization, C++ was utilized alongside Microsoft Visual Studio development tools. The outcomes were compared with the Python open-source results from [14] for validation of functionality and accuracy.

Upon analyzing the operational flow of the convex optimization based SVM algorithm, it was determined to segment the complex process into three stages (illustrated in the SVM Module in Fig. 3) to enhance design efficiency. These stages are orchestrated non-sequentially to leverage the parallel processing capabilities of FPGA-based hardware.

In the initial phase of the hardware project, a suitable mathematical core was selected and integrated. This entails transforming linear-inseparable vectors into linear-separable vectors. Various kernels can facilitate this transformation.

$$K(A_x, A_y) = \Phi(A_x) \times \Phi(A_y)$$
$$\int\int g(A_x)K(A_x, A_y)g(A_y)dA_x dA_y \geq 0 \quad (1)$$

Within the convex function (CO) optimization based SVM algorithm, the optimization stage emerges as the most intricate phase among the three. To simplify its complexity, stage 2 is divided into three distinct phases: parameter initialization, convex optimization, and displacement value computation.

During the next step a set of parameters is calculated, spanning feature function parameters (N, u), constraint parameters (G, h, D, z_{const}), and supplementary parameters (α, Gr), including the admissible point.

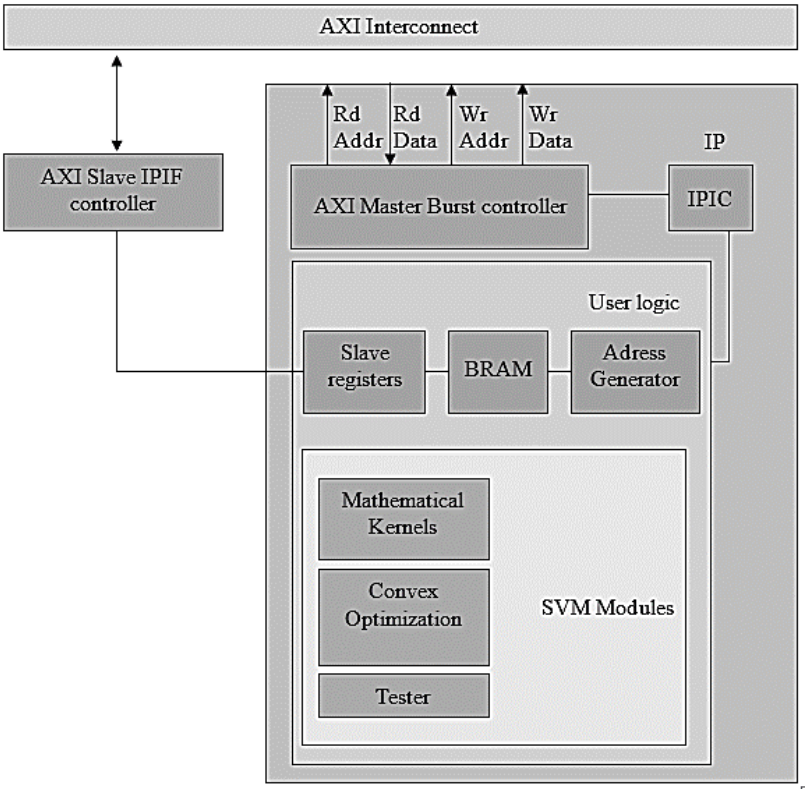

Fig. 3. Preload method and upper-level architecture

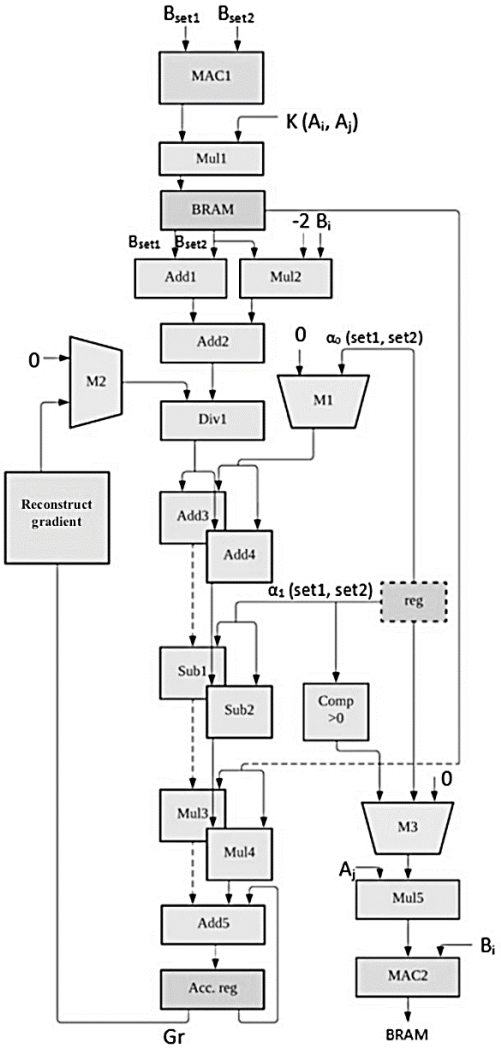

Fig. 4. Data path

The data processing scheme for optimization, presented in Figure 4, comprises various components (registers, multiplexers etc). The optimization process aims to determine the value α_{min}, computed using modules presented in Fig 4.

The final step of stage 2 involves the calculation of the displacement value. Also known as the z-shift value, it signifies the intersection of the hyperplane from the datum. The displacement value is derived using the following equation (2):

$$z = -\frac{1}{2}\left[\max_{\{x|B_y=-1\}}\left(\sum_{y=1}^{m}\alpha_x BK(A_x, A_y)\right) + \min_{\{x|B_y=+1\}}\left(\sum_{y=1}^{m}\left(\alpha_x B_x K(A_x, A_y)\right)\right)\right] \quad (2)$$

$$z = \frac{\left(\sum_{x=1}^{m} B_x - \sum_{x=1}^{m}\alpha_x B_{sv} K(A_x, A_y)\right)}{\text{number of support vectors}, n_{sv}} \quad (3)$$

In Stage 3, the testing process unfolds. Typically, in classification tasks, the input dataset is split into two subsets: training and testing. Data path for the test process is presented in Figure 5.

Experiments

Testing was carried out to assess the viability and efficacy of the proposed technique. Execution time analysis was conducted to evaluate the performance of the proposed embedded hardware. The firmware operates on an ARM Cortex-A9 processor integrated into the same DE1 Soc development platform.

Figures 6 and 7 presents the experimental results.

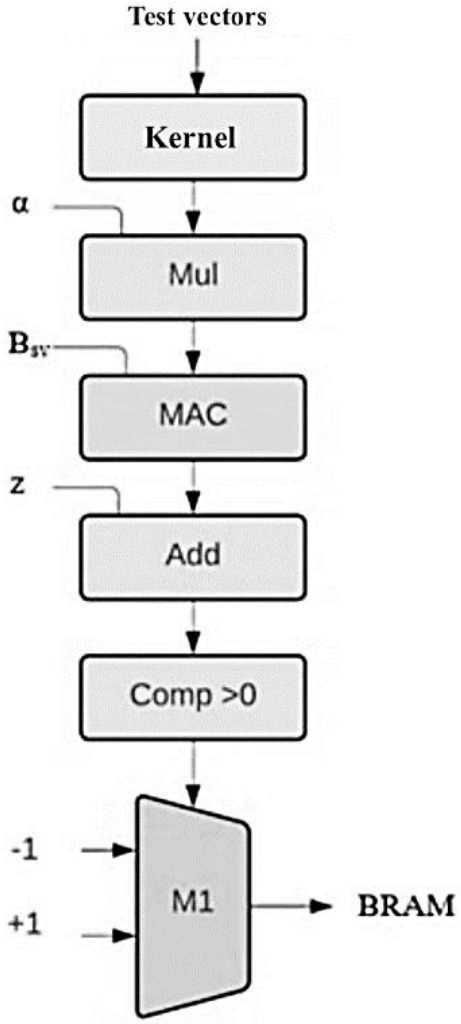

Fig. 5. Data path for the test process

An analysis of classification accuracy reveals that the accuracy varies across different datasets and percentages of training sets. Notably, the Pulsar test dataset attains the highest classification accuracy of 96% when utilizing the polynomial kernel with a training set percentage of 90% of the dataset.

In addition, experiments were conducted to assess classification accuracy using Python code. Plots illustrating the results of the Python code design, employing different kernels, are depicted in Figures 8 and 9 for the Pulsar and Iris Flowers experimental datasets, respectively.

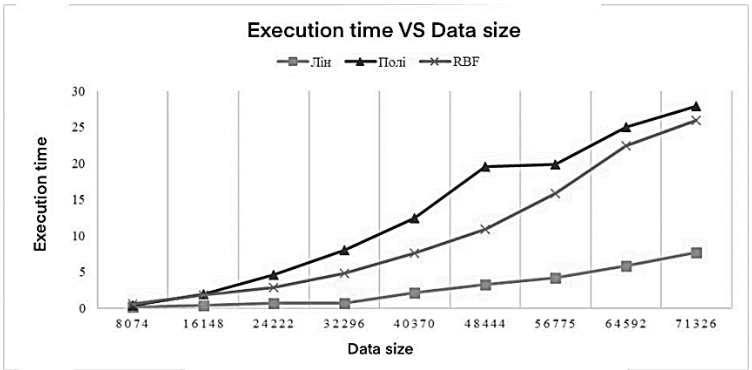

Fig. 6. Embedded software for CO-based SVM algorithm: Execution time and data size for the "Pulsar" experimental data set

Fig. 7. Embedded hardware for the CO-based SVM algorithm: Execution time and data size for the "Pulsar" experimental data set

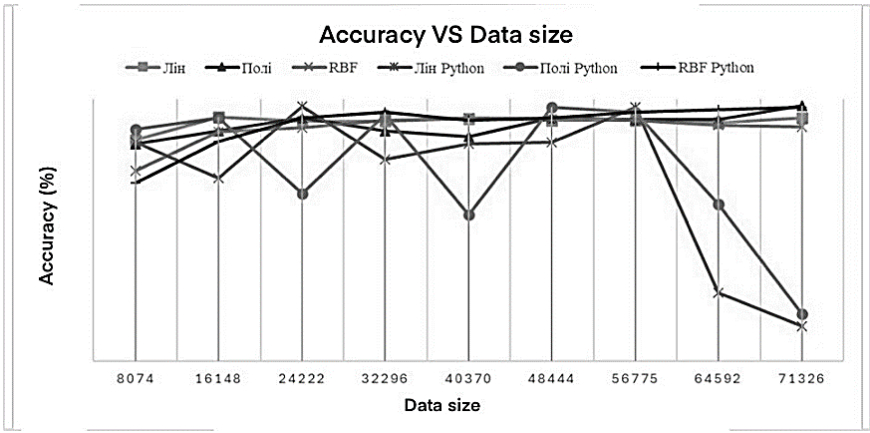

Fig. 8. Classification accuracy and data size for the "Pulsar" experimental dataset

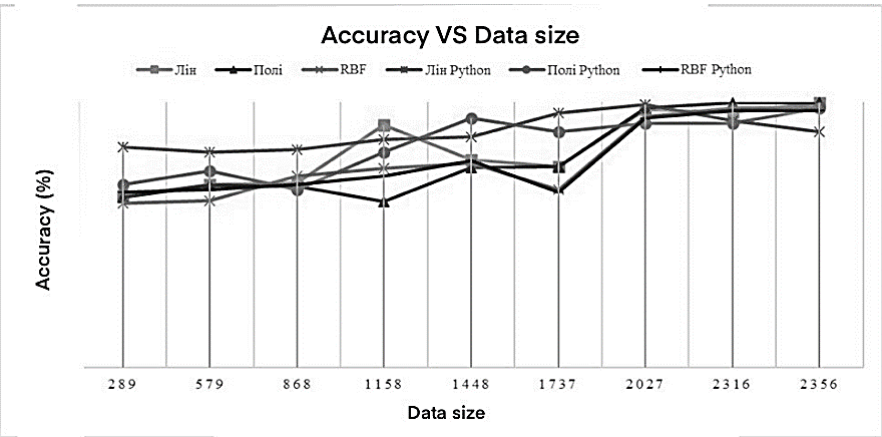

Fig. 9. Classification accuracy and data size for the "Iris Flowers" experimental dataset

Conclusions

The work resulted in the development of a new, specialized, and optimized hardware accelerator based on FPGA for the Support Vector Machine (SVM) method using convex optimization (CO) on embedded platforms. The proposed embedded architectures are designed to be universal, parameterized, and scalable. This means that these embedded solutions can accommodate different datasets of varying sizes and can be implemented on various embedded platforms, including those equipped with the latest FPGAs. They are also capable of handling both linear and nonlinear discrimination across multidimensional datasets.

References

1. Afifi, S., GholamHosseini, H. & Sinha, R. FPGA Implementations of SVM Classifiers: A Review. SN COMPUT. SCI. 1, 133 (2020).
2. Xiaohui Song, Hong Wang, Lingfeng Wang. FPGA Implementation of a Support Vector Machine Based Classification System and Its Potential Application in Smart Grid. 11th International Conference on Information Technology: New Generations. 2014
3. Amezzane, I., Fakhri, Y., El Aroussi, M., Bakhouya, M. (2020). Hardware Acceleration of SVM Training for Real-Time Embedded Systems: Overview. In: Dos Santos, S., Maslouhi, M., Okoudjou, K. (eds) Recent Advances in Mathematics and Technology. Applied and Numerical Harmonic Analysis. Birkhäuser, Cham.
4. Frank Bruno, Guy Eschemann. The FPGA Programming Handbook: An Essential Guide to FPGA Design for Transforming Your Ideas into Hardware Using SystemVerilog and VHDL, Second Edition. 2024. 392.

5. Daniel H. Noronha, Matheus F. Torquato, Marcelo A.C. Fernandes. A parallel implementation of sequential minimal optimization on FPGA, Microprocessors and Microsystems, Volume 69, 2019, Pages 138-151.
6. De Caires, S. A., Keshavarzi, A., Bottega, E. L., & Kaya, F. (2024). Towards site-specific management of soil organic carbon: Comparing support vector machine and ordinary kriging approaches based on pedo-geomorphometric factors. Computers and Electronics in Agriculture, 216, 108545.
7. Sunday O. Oladejo, Stephen O. Ekwe, Adedotun T. Ajibare, Lateef A. Akinyemi, Seyedali Mirjalili, Chapter 36 - Tuning SVMs' hyperparameters using the whale optimization algorithm, Editor(s): Seyedali Mirjalili, Handbook of Whale Optimization Algorithm, Academic Press, 2024.
8. Lucas Amilton Martins, Felipe Viel, Laio Oriel Seman, Eduardo Augusto Bezerra, Cesar Albenes Zeferino, A real-time SVM-based hardware accelerator for hyperspectral images classification in FPGA, Microprocessors and Microsystems, Volume 104, 2024.
9. S. Afifi, H. GholamHosseini and R. Sinha, "SVM classifier on chip for melanoma detection," 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Jeju, Korea (South), 2017, pp. 270-274, doi: 10.1109/EMBC.2017.8036814. keywords: {Support vector machines;IP networks;Malignant tumors;Field programmable gate arrays;Hardware;Tools;Embedded systems},
10. Shereen Afifi, Hamid GholamHosseini, Roopak Sinha, A system on chip for melanoma detection using FPGA-based SVM classifier, Microprocessors and Microsystems, Vol. 65, 2019, Pages 57-68, ISSN 0141-9331, https://doi.org/10.1016/j.micpro.2018.12.005.
11. J. Manikandan, B. Venkataramani, Study and evaluation of a multi-class SVM classifier using diminishing learning technique, Neurocomputing, Volume 73, Issues 10–12, 2010, Pages 1676-1685, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2009.11.042.
12. DE1-SoC User Manual (rev.F/rev.G Board) Terasic Technology Inc. URL: https://www.terasic.com.tw/cgi-bin/page/archive.pl.

Sergii LYSENKO **Сергій ЛИСЕНКО**	DrSc (Engineering), Professor of Computer Engineering & Information Systems Department, Khmenlnytskyi National University https://orcid.org/0000-0001-7243-8747 e-mail: sirogyk@ukr.net	Доктор технічних наук, професор кафедри комп'ютерної інженерії та інформаційних систем, Хмельницький національний університет
Yevheniia SHPULIAR **Євгенія ШПУЛЯР**	Education seeker of Computer Engineering & Information Systems Department, Khmelnytskyi National University, Khmelnytskyi, Ukraine, e-mail: eugenia.shpuliar@gmail.com https://orcid.org/0009-0002-0233-2909	Здобувач освіти, кафедра комп'ютерної інженерії та інформаційних систем, Хмельницький національний університет, Хмельницький, Україна.