МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

ВІСНИК
ЧЕРКАСЬКОГО ДЕРЖАВНОГО ТЕХНОЛОГІЧНОГО УНІВЕРСИТЕТУ

Науковий збірник

Том 29,
№ 2. 2024

ЧЕРКАСИ
2024

ISSN: 2306-4412
E-ISSN: 2708-6070

Засновник:
Черкаський державний технологічний університет

Рік заснування: 1996

Рекомендовано до друку та поширення
через мережу Інтернет Вченою радою
Черкаського державного технологічного університету
(протокол № 10 від 27 травня, 2024 р.)

Державна реєстрація: Ідентифікатор медіа R30-04613.
Рішення Національної ради України з питань телебачення і радіомовлення
№ 1916, протокол № 17 (30.05.2024 р.).

Науковий збірник входить до переліку наукових фахових видань України
Категорія «Б». Спеціальність:
122 – Комп'ютерні науки, 123 – Комп'ютерна інженерія, 125 – Кібербезпека,
126 – Інформаційні системи та технології, 151 – Автоматизація та комп'ютерно-інтегровані технології, 172 – Телекомунікації та радіотехніка
(наказ МОН України від 28 грудня 2019 року № 1643)

Науковий збірник представлено у міжнародних наукометричних базах даних, репозитаріях та пошукових системах: Bielefeld Academic Search Engine (BASE), Crossref, CiteFactor, Index Copernicus, Ulrich's Periodicals Directory, WorldCat, Eurazian Scientific Journal Index (ESJI), Directory of Open Access Journals (DOAJ), Open Ukrainian Citation Index (OUCI), Наукова періодика України, Національна бібліотека України імені В. І. Вернадського (НБУВ), Dimensions, UCSB Library, Google Академія, German Union Catalogue of Serials (ZDB), Бібліотека Університету Осло, Бібліотека Університету Халла, SOLO - Search Oxford Libraries Online, Інститут Європейського університету, Бібліотека Лейпцизького університету (UBL), Бібліотека Кембриджського університету, Zendy Research Library, Asian Science Citation Index

Адреса редакції:
Черкаський державний технологічний університет
18006, бульв. Шевченка, 460, м. Черкаси, Україна
E-mail: info@bulletin-chstu.com.ua
https://bulletin-chstu.com.ua/uk

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
CHERKASY STATE TECHNOLOGICAL UNIVERSITY

BULLETIN
OF CHERKASY STATE TECHNOLOGICAL UNIVERSITY

Scientific Collection

Volume 29,
No. 2. 2024

CHERKASY
2024

ISSN: 2306-4412
E-ISSN: 2708-6070

Founder:
Cherkasy State Technological University

Year of foundation: 1996

Recommended for printing and distribution
via the Internet by the Academic Council
of Cherkasy State Technological University
(Minutes No. 10 of May 27, 2024)

State Registration: Media identifier R30-04613.
Decision of the National Council of Television and Radio Broadcasting of Ukraine
No. 1916, Minutes No. 17, dated 30.05.2024.

The scientific collection is included in the list of Professional Scientific Publications of Ukraine
Category "B". Specialties: 0613 – Software and applications development and analysis,
0611 – Computer use, 0612 –Database and network design and administration,
0714 – Electronics and automation
(order of the Ministry of Education and Science of Ukraine No. 1643, dated December 28, 2019)

The scientific collection is presented international scientometric databases, repositories and scientific systems: Bielefeld Academic Search Engine (BASE), Crossref, CiteFactor, Index Copernicus, Ulrich's Periodicals Directory, WorldCat, Eurazian Scientific Journal Index (ESJI), Directory of Open Access Journals (DOAJ), Open Ukrainian Citation Index (OUCI), Scientific Periodicals of Ukraine, Vernadsky National Library of Ukraine (VNLU), Dimensions, UCSB Library, Google Scholar, German Union Catalogue of Serials (ZDB), University of Oslo Library, University of Hull Library, SOLO - Search Oxford Libraries Online, European University Institute, Leipzig University Library (UBL), Cambridge University Library, Zendy Research Library, Asian Science Citation Index

Editors office address:
Cherkasy State Technological University
18006, 460 Shevchenko Blvd., Cherkasy, Ukraine
E-mail: info@bulletin-chstu.com.ua
https://bulletin-chstu.com.ua/en

Редакційна колегія

Editorial Board

Зміст / Contents

BULLETIN of Cherkasy State Technological University

Journal homepage: https://bulletin-chstu.com.ua/en

Vol. 29 No. 2. 2024

UDC 004.43
DOI: 10.62660/bcstu/2.2024.10

A software model to generate permutation keys through a square matrix

Emil Faure*

Doctor of Technical Sciences, Professor
Cherkasy State Technological University
18006, 460 Shevchenko Blvd., Cherkasy, Ukraine
https://orcid.org/0000-0002-2046-481X

Anatoly Shcherba

PhD in Physical and Mathematical Sciences, Associate Professor
Cherkasy State Technological University
18006, 460 Shevchenko Blvd., Cherkasy, Ukraine
https://orcid.org/0000-0002-3049-3497

Artem Skutskyi

Postgraduate Student
Cherkasy State Technological University
18006, 460 Shevchenko Blvd., Cherkasy, Ukraine
https://orcid.org/0000-0002-8632-1176

Artem Lavdanskyi

PhD in Technical Sciences, Associate Professor
Cherkasy State Technological University
18006, 460 Shevchenko Blvd., Cherkasy, Ukraine
https://orcid.org/0000-0002-1596-4123

Abstract. Information security and data protection are among the key aspects, which should be intensively developing in the 21st century. A conventional approach to cryptographic algorithms offers to apply matrices to represent information. However, more recent approaches deploy other data structures, including permutations, thus necessitating accordance between differing data structures to integrate different methods into a wholistic system of processing and transmitting information. This study aims to generate permutations, which serve as a key for factorial data coding according to a known key matrix. The paper presents two algorithms for transforming a square matrix into a permutation. An example of matrix transformation following each of the proposed algorithms is given. A software model was created and described to investigate the transformation of square matrices into permutations with the Matlab software product. The authors have considered the built-in methods of statistical information processing in the Matlab program and their graphical representation by built-in functions, which are applied in the process of the software model. A matrix transformation has been performed according to the proposed algorithms. The paper investigates all possible combinations of a square matrix of order 2 with elements referring to the finite integer field modulo $p = 17$ and $p = 23$. According to each transforming algorithm, the results of a square matrix transforming into a permutation number are obtained in the lexicographic order. The statistical properties of the obtained results have been studied, and the most

Article's History: Received: 17.02.2024; Revised: 01.05.2024; Accepted: 27.05.2024.

Suggested Citation:
Faure, E., Shcherba, A., Skutskyi, A., & Lavdanskyi, A. (2024). A software model to generate permutation keys through a square matrix. *Bulletin of Cherkasy State Technological University*, 29(2), 10-23. doi: 10.62660/bcstu/2.2024.10.

*Corresponding author

efficient algorithm for transforming matrices into permutations has been determined based on the distribution uniformity criterion for the generated permutation numbers. The study demonstrates that this algorithm can potentially be deployed in information exchange systems based on factorial data coding

Keywords: factorial data coding; data security; key agreement; statistics; uniform distribution

INTRODUCTION

Software is widely used in everyday life. Every second, millions of operations are performed globally, from ordinary requests to requests that require a high level of security and reliability of information transmission. These operations include managing bank accounts, electronic digital signatures for documents, automated technological processes, controlling unmanned aerial vehicles, providing secure communication channels for the Internet of Things (IoT), etc.

Rapid development of information technologies involves information security as well. In particular, Ye. Kaptiol & I. Horbenko (2020) in their work used cloud computing with IBM quantum computers to investigate Grover's algorithm, which is a necessary method to solve cryptology problems. S. Joshi *et al.* (2023) discussed research on the ways quantum computing can break the RSA encryption algorithm in polynomial time, an operation that conventional computers are not able to perform because of processing power limitations. P. Li *et al.* (2024b) propose a scalable parallel circuit for ultrafast random bit generation at 100 terabits per second based on a single micro-ring resonator. In an investigation into Nvidia's development, A. Lavdanskyi *et al.* (2023) demonstrates how software and hardware technology of compute unified device architecture (CUDA) of graphics processors increases the performance of operations on permutations.

Based on the above, users have access to hardware with potent computing capabilities, which allows them to test the reliability of conventional encryption algorithms. However, modern technology may be used to achieve malicious goals. When determining the cryptographic stability of encryption algorithms, the breaking time is an important parameter. Passwords with a total length of 8 characters (including numbers, uppercase letters, lowercase letters, and special characters) and encrypted with MD5 hash function require approximately one hour to be computed when using an RTX4090 graphics card. The cloud computing detailed in S. Rani *et al.* (2023) is able to reduce this time by a factor of 4. Factorial data coding according to the definition offered in V. Shvydkyi *et al.* (2021) is the process of transforming information for its protection based on the factorial numbering system. Non-separable factorial data coding is the coding that uses π permutations of length M as signal carriers. The elements of the permutation are integer numbers limited by the range [0; M – 1]. This transformation enables implementation of integrated information protection against unauthorised access and communication channel errors. An advantage to non-separable factorial coding is that the cardinality of the key space is a function of $M!$, which, as M increases, grows faster than all polynomial and exponential functions of M.

On the other hand, a considerable number of cryptographic transformations are implemented by matrix information representations. In particular, H. Huang *et al.* (2022) analysed the imminent cryptographic systems based on tropical circular matrices. The results offer a new encryption algorithm with an open key based on the tropical circular matrices. This cryptosystem is able to resist KU and RM attacks. A.R. Naseri *et al.* (2023) applied the class of square Fibonacci matrices $(q+1)\times(q+1)$ in cryptographic transformations, where q – an integer number. This type of cryptographic permutations uses matrices as a key in their encryption algorithm. The approach simplified and enhanced the key exchange procedure. M. Maxrizal (2022) proposes matrices with zero determinant as a key to implement three-pass data transmission protocols. A study by F. Al-Shaarani & A. Gutub (2022) demonstrates a cryptographic key transmission algorithm through a matrix. Along with that, a matrix is employed to obtain stereograms, which are additionally encrypted and transmitted through an unsecured communication channel.

Compatible implementation of matrix data representation, for cryptographic key agreement in particular, as well as for factorial coding of data to be transmitted through secure channels, requires transforming the received matrix key into a permutation key. Consequently, a need arises to build and investigate algorithms of transforming matrices into permutations. Considering the fact that key lengths and information block lengths for matrix-based permutations and factorial protocols do not coincide in the majority of cases, reflecting matrices into permutations is not bijective, being simultaneously surjective and injective instead (Maturin *et al.*, 2023).

The aim of this research was to agree permutation keys for factorial data coding based on the informational interaction of the key matrix known to the conversers. The main goal includes several objectives:

- to develop algorithms to transform a matrix into a permutation;
- to design and to analyse bar charts reflecting permutation frequency after enumerating every square matrix of the given order for each algorithm;
- to compare the bar charts obtained under various transformation parameters.

MATERIALS AND METHODS

According to the definition offered by W. Greub (1975), a square matrix is a matrix that has a number of rows equal to the number ofcolumns: $i = j$. In this case, $A = (a_{ij})_{n \times n}$ is a square matrix of order n. The elements of a_{ij} matrix are distributed within the $0 \leqslant a_{ij} \leqslant p - 1$ range, where p – an integer number; i, j – the row and the column of the matrix, correspondently. In this research, it is proposed two algorithms to transform the A matrix into a permutation: 1) by representing A matrix in decimal and factorial notations; 2) by using auxiliary square matrices.

Permutation of the matrix A according to the first algorithm includes three steps.

1) Transforming matrix A into a decimal A_{10} value by the calculation noted as:

$$A_{10} = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} \cdot p^{(i-1)n+j-1} =$$

$$= a_{11} \cdot p^0 + a_{12} \cdot p^1 + \ldots + a_{nn} \cdot p^{n^2-1} \quad (1)$$

2) Limiting the decimal equivalent A_{10} of matrix A to the range $[0; M! - 1]$:

$$A_{10}^{norm} = |A_{10}|_{M!}. \quad (2)$$

3) Transforming the decimal representation of the number A_{10}^{norm} selected from the range $[0; M! - 1]$ into the factorial number A_F and permutation π of length M:

$$A_{10}^{norm} \to A_F \to \pi. \quad (3)$$

Transformation (3) is simple and shall not be studied in this paper. The second algorithm to transform a matrix into a permutation is to be implemented in four steps.

1) Building auxiliary matrices $H_k = (h_{ij}(k))_{(n \times n)}$, where $(h_{ij}(k)) = |(k-1) \cdot n^2 + (i-1) \cdot n + j - 1|_M$, while $1 \le k \le \left\lceil \frac{M}{n^2} \right\rceil$.

Thus, when M value exceeds n^2, there are several auxiliary matrices H_k. For example, if $M = 8$, $n = 2$, two auxiliary matrices will be necessary: $H_1 = \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix}$, $H_2 = \begin{pmatrix} 4 & 5 \\ 6 & 7 \end{pmatrix}$.

If $M = 9$, $n = 2$, three auxiliary matrices are to be built: $H_1 = \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix}$, $H_2 = \begin{pmatrix} 4 & 5 \\ 6 & 7 \end{pmatrix}$, $H_3 = \begin{pmatrix} 8 & 0 \\ 1 & 2 \end{pmatrix}$.

2) Successive multiplication from the left of the matrix A by all auxiliary matrices H_k:

$$B_k = A \cdot H_k = (b_{ij}(k))_{(n \times n)}. \quad (4)$$

3) Finding factorial coefficient a_l, $0 \leqslant l \leqslant M - 1$, of number A_F by calculation:

$$a_l = |b_{ij}(k)|_{(l+1)}, \quad (5)$$

where values i, j, k, correspondent to the coefficient a_l, are selected so that $l = h_{ij}(k)$.

4) Converting factorial number A_F into permutation π of length M:

$$A_F \to \pi. \quad (6)$$

To automate operations converting a decimal number into a permutation, the authors of this study have developed a transformation function *int2perm(number, base_permut)* in Matlab programming environment. The function accepts *number*, a number in decimal representation to be transformed into a permutation, as input parameter. Parameter *base_permut* is a base permutation presented as a vector (row). Figure 1 features the block diagram showing the function algorithm converting a decimal number into a permutation.

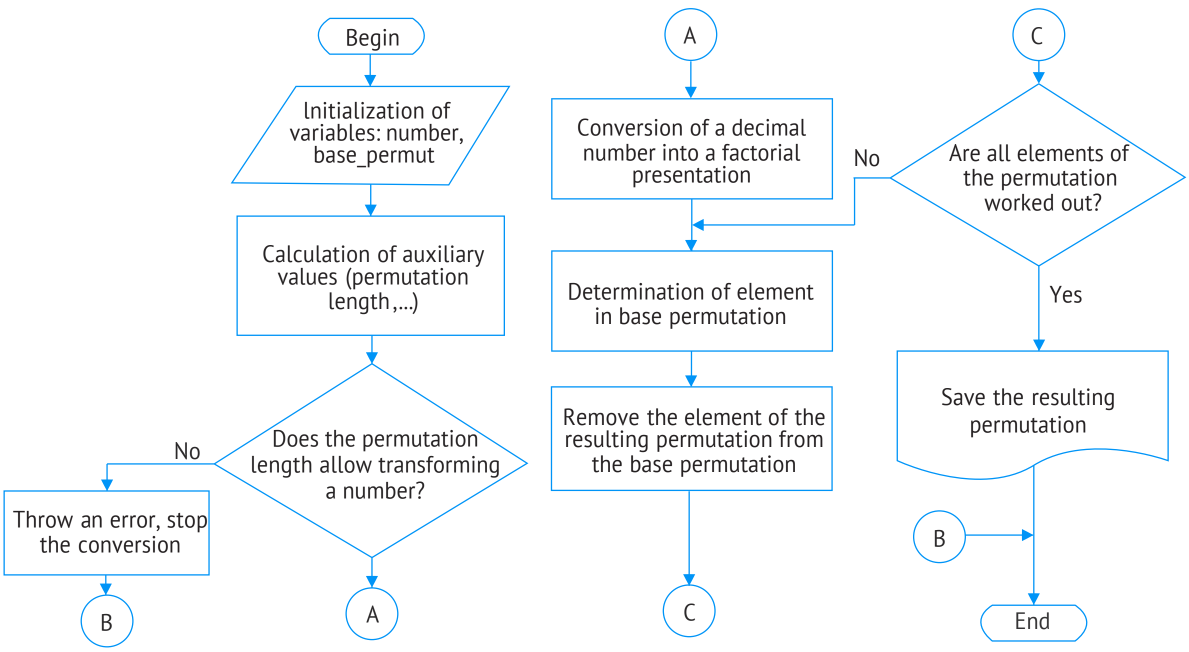


Figure 1. Algorithm for transforming a number into a permutation (*int2perm* function)
Source: developed by the authors

To conduct the research, Matlab software environment has been involved. The study applied programming principles and product capabilities developed by D.J. Higham & N.J. Higham (2005) and D.T. Valentine & B.H. Hahn (2023). Two additional methods have been developed to represent the number in the factorial and decimal notations: *int2fact()* – which is a method of transforming a number from decimal into factorial notation, and *factint()* – which is a method for reverse transformation of a number from factorial to decimal notation.

To construct the software model, the following basic variables were used:

- *A* – a square matrix;
- *n* – the order of matrix *A*;
- *M* –the permutation length;
- *pMax* denotes the maximum values for the elements of the matrix *A* (a variable that allows verifying the $a_{ij} \leqslant p - 1$ condition);
- *Adec* – the current value of the decimal number obtained from matrix *A* by the first transformation algorithm (a variable containing the result of transformation according to Equation (1) and presented in decimal notation);
- *Bdec* – the current value of the number obtained from matrix *A* according to the second transformation algorithm (a variable containing the transformed result according to Equation (5) and reduced to decimal notation);
- *AdecCollection* – the list of matrix values converted according to the first transformation algorithm (the variable is a one-dimensional array (vector), where each element is the result of stepwise transformations of the matrix *A* according to the first algorithm *Adec*);
- *BdecCollection* – the list of the matrix values converted according to the second transformation algorithm (the variable is a one-dimensional array (vector), where each element is the result of stepwise transformations of the matrix *A* according to the second algorithm *Bdec*);
- *aCollection* – the list of obtained matrix *A* values (the variable is represented as a three-dimensional array. Two dimensions define rows and columns, while the third dimension corresponds to the sequence number of each matrix generated within the range $[1; p^{n^2}]$).

The software model algorithm. The procedure for launching the software model includes the following steps:

1. Entering parameters: *M*, *pMax*, *n*.

2. Calculating values: number *M*!, the number of possible matrix *A* variants, number of auxiliary matrices to be processed according to the second transformation algorithm.

3. Enumerating all matrix *A* variants under the given conditions, transforming them according to the first and second algorithms.

4. Saving the obtained values.

As a result of the software model, statistical data were obtained to be further evaluated as an expansion of the *.mat workspace.

The block diagram of the algorithm described above can be represented as shown in Figure 2.

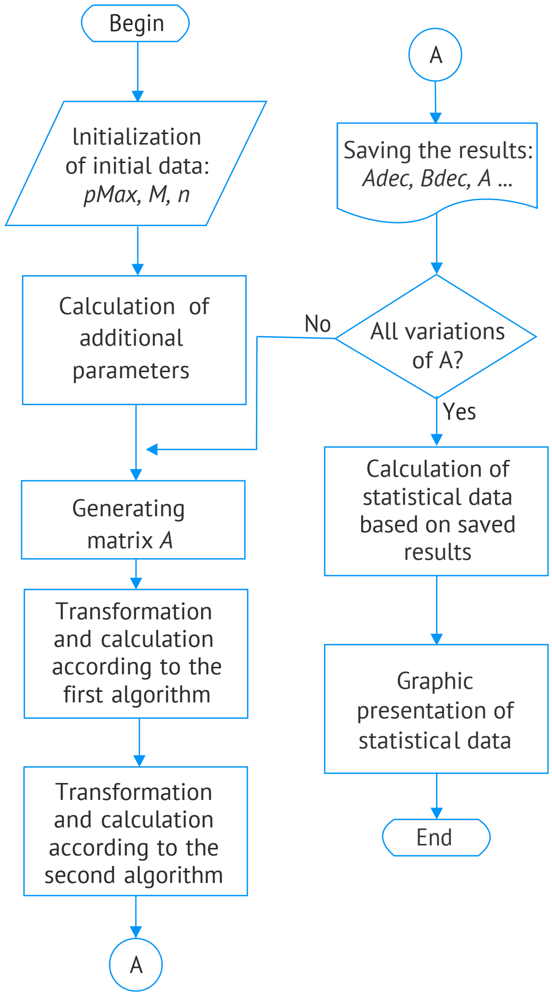


Figure 2. Software model algorithm
Source: developed by the authors

Collecting statistical data and their validation. The Matlab software product stores the results of the experiments as a workspace extension *.mat. This format saves all the variables produced by the software. By saving the program's working space, it is possible to analyse the simulation results and access the arrays of variable values separately from the main algorithm of the model, i.e., to simulate the experiments and save the results separately for each of the experiments. To save the workspace, Matlab program employs *save(name)* function where *name* – the file name. Thus, after starting the simulation program in two experiments, the workspace was saved for each simulation. Two files containing simulation variables and their values have thus been obtained. Applying this procedure to save the workspace of the program variables, one can separate the software part of the simulation from additional calculations related to obtaining statistical parameters and to separate the experimental part from the results analysis (analytical part). In addition, this procedure optimises analysing the results by creating a graphical interface, adds the possibility of extending the study by adding or diversifying the methods of statistical analysis applied to analyse the results. The software implementation of the model performs solely data generation, the analytical part is responsible for calculating statistical parameters, graphs, etc. Based on the defined list of variables present in the **.mat* file, an additional program has been developed that calculates the statistical parameters of the obtained results. Figure 3 demonstrates the program algorithm.

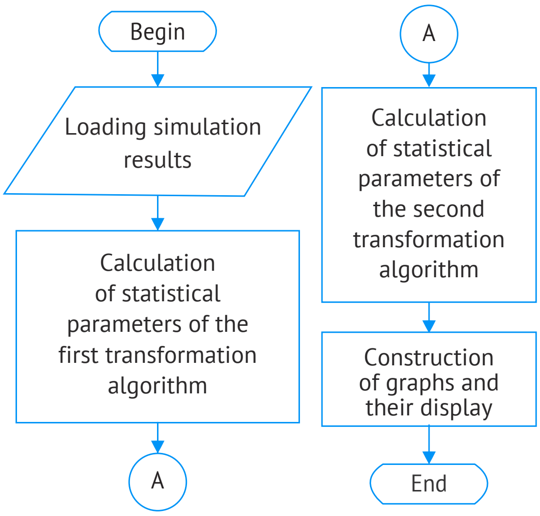


Figure 3. Calculating statistical parameters
Source: developed by the authors

While implementing the algorithm, the built-in functions of Matlab were deployed: *min* – the minimum value in a numerical sample; *max* – the maximum value in a numerical sample; *range* – the range of values in a numerical sample; *num* – sample size; *mean* – sample mean; *median* – the median of the sample values; *standart deviation* – standard deviation.

Matlab function *datastats()* can retrieve the listed values. The *var()* function will calculate the variance. The *bar(x,y,*)* method was to display bar graphs for absolute frequency, where *x* – the vector of values along the *Ox* axis; *y* – vector of values along the *Oy* axis; * denotes additional settings (line thickness, colour, etc.). The *plot(x,y,*)* method was employed to display graphs of relative frequency.

RESULTS

Simulation of the described algorithms and methods for matrix transformation into permutation was performed by conducting two experiments with the following input parameters:

1) $p = 17, M = 8, n = 2$;
2) $p = 23, M = 8, n = 2$.

A fragment of setting up the software implementation by entering the initial data for the first experiment is shown in Figure 4.

```
pMax = 17; %maximum p
pMin = 0; %minimum p
M = 8; % permutation lenght
```

Figure 4. A code fragment to enter primary parameters
Source: developed by the authors

The set cardinality of matrix *A* equals to $17^4 = 83521$ for the first experiment and to $23^4 = 279841$ for the second experiment. The data arrays resulting from experimental data are represented by sets of permutation numbers A_{10}^{norm} for the first algorithm and the obtained combinations of the matrix *A*. The array of results saved for the second transformation algorithm contains permutation numbers obtained by transforming the A_F values from the factorial into decimal representation. Figures 5 and 6 contain fragments of the results for the first and second experiments.

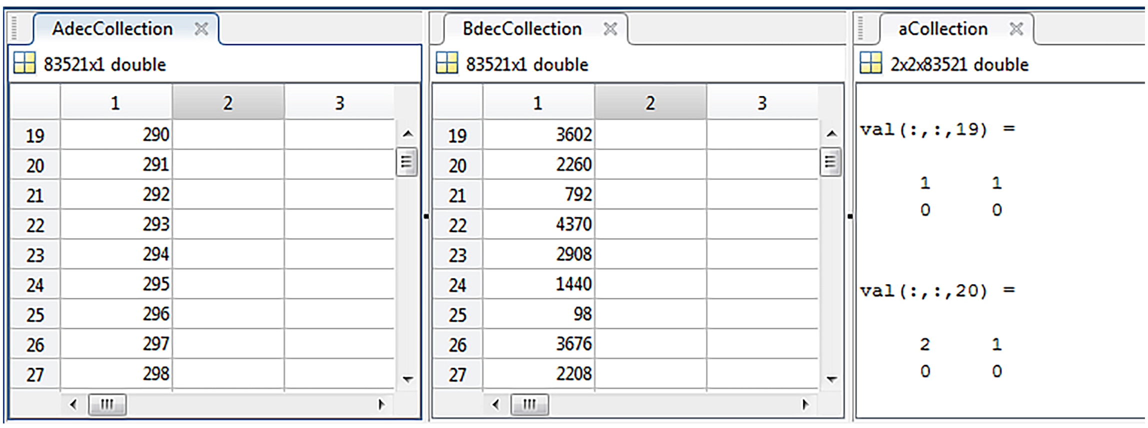


Figure 5. Generated and saved transformation results according to the first and second algorithms for $p = 17, M = 8, n = 2$
Source: developed by the authors

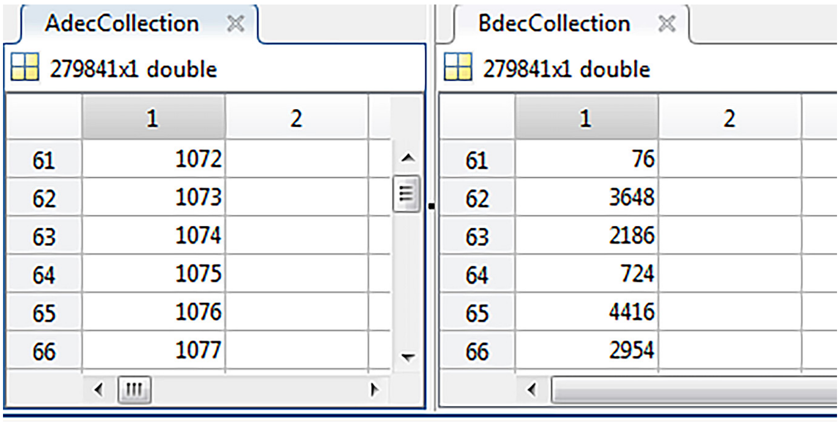


Figure 6. Generated and saved transformation results according to the first and second algorithms for $p = 23$, $M = 8$, $n = 2$

Source: developed by the authors

Variable *AdecCollection* contains the transformation results according to the first algorithm (A_{10}^{norm} value). The variable *BdecCollection* contains the transformation results according to the second algorithm (decimal representation of the received permutation number A_F). The variable *aCollection* contains all the obtained *A* matrix values.

In the first experiment with input parameters $p = 17$, $M = 8$, $n = 2$, the following statistical results have been obtained for the transformation executed according to the first algorithm:

- the number of generated matrix *A* combinations, attribute *num* in the *datastat()* function: $N = 83521$;
- the maximum value of the number obtained in the result of transforming the matrix *A*: $Nmax = M! - 1 = 40319$;
- the minimum value of the number obtained in the result of transforming the matrix *A*: $Nmin = 0$;
- the mean value in the generated sample: $mean = 19514$;
- the median values in the generated sample: $median = 19439$;
- the maximum number of permutation repetitions: $rep_{max}^{A1} = 3$;
- the minimum number of permutation repetitions: $rep_{min}^{A1} = 2$;
- A_{10}^{norm} values from the range [0; 2880] repeat rep_{max}^{A1} times;
- A_{10}^{norm} values from the range [2881; 40320] repeat rep_{min}^{A1} times;
- variance: $D_{A1} = 142498178.841$;
- the maximum value of the relative frequency for a permutation to occur is: $W_{max}^{A1} = 3.5919 \cdot 10^{-5}$;
- the minimum value of the relative frequency for a permutation to occur is: $W_{min}^{A1} = 2.3946 \cdot 10^{-5}$.

Distribution of absolute frequencies N_j, $0 \leqslant j \leqslant M! - 1$ of permutations with the number $x_j = j$ in the lexicographic order following the first transformation algorithm is shown in Figure 7 as a bar graph.

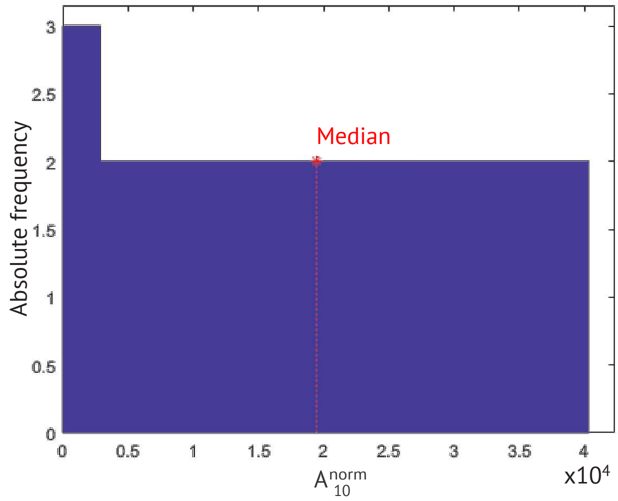


Figure 7. Distribution of absolute frequencies for a permutation to occur according to the first transformation algorithm based on the results of the first experiment

Source: developed by the authors

Figure 8 shows the graph of the relative frequency of occurrence of permutations according to the first transformation algorithm is shown in. *Ox* axis contains the generated values of the permutation sequence numbers when matrix *A* is transformed by formula A_{10}^{norm}, and *Oy* axis is the relative frequency of the generated number.

In the experiment with input parameters $p = 17$, $M = 8$, $n = 2$, the following statistical results were obtained for the transformation according to the second algorithm:

- the number of generated combinations of the matrix *A*, attribute *num* in the *datastat()* function: $N = 83521$;
- the maximum value of the number resulting from transforming the matrix *A*: $Nmax = 40198$;
- the minimum value of the number resulting from transforming the matrix *A*: $Nmin = 0$;

- the mean value in the generated sample: $mean = 19995$;
- the median values in the generated sample: $median = 20012$;
- the maximum repetition number of the generated numbers (sequence number of the permutation), $rep^{B1}_{max} = 52$;
- the minimum repetition number of the generated numbers (sequence number of the permutation), $rep^{B1}_{min} = 0$;
- the number of ordinal number values of the permutations that have the maximum number of repetitions rep^{B1}_{max} is 408;
- the number of ordinal number values of the permutations that have the minimum number of repetitions rep^{B1}_{min} is 38159;
- variance: $D_{B1} = 136109307.2089$;
- the maximum value of the relative frequency for a permutation to occur: $W^{B1}_{max} = 62.2597 \cdot 10^{-5}$;
- the minimum value of the relative frequency for a permutation to occur: $W^{B1}_{min} = 35.9191 \cdot 10^{-5}$.

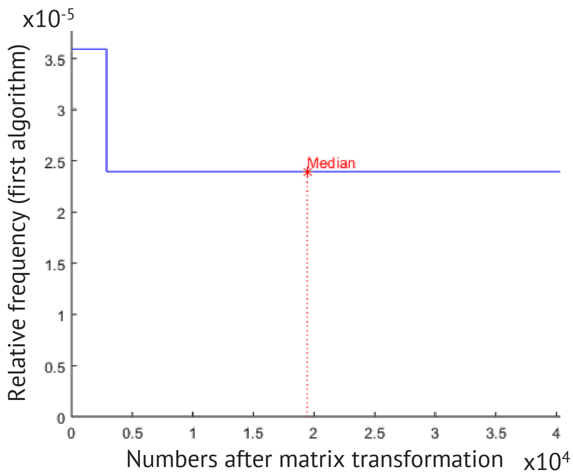


Figure 8. Relative frequency for a permutation to occur according to the first transformation algorithm based on the results of the first experiment
Source: developed by the authors

Figure 9 shows a bar graph for distribution of the relative frequency N_j, $0 \leqslant j \leqslant M! - 1$ of permutations with number $x_j = j$ in the lexicographic order according to the first transformation algorithm. The graph describing the relative frequency of permutation occurrence according to the first transformation algorithm is shown in Figure 10. *Ox* axis contains the generated values of the permutation sequence numbers when the matrix *A* is transformed by formula A^{norm}_{10}, and *Oy* axis is the relative frequency of the generated number.

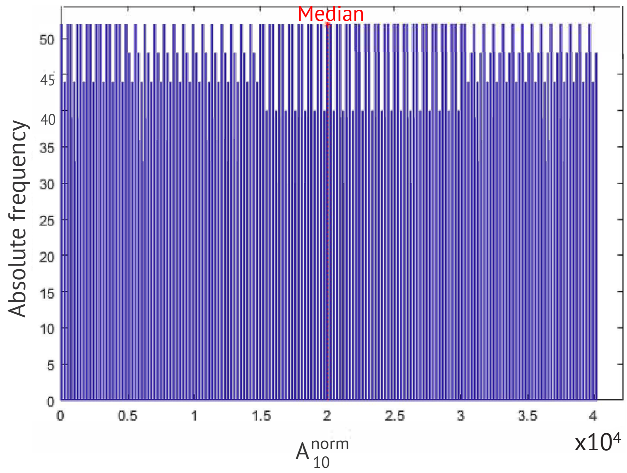


Figure 9. Distribution of the absolute frequencies for a permutation to occur according to the second transformation algorithm, based on the results of the first experiment
Source: developed by the authors

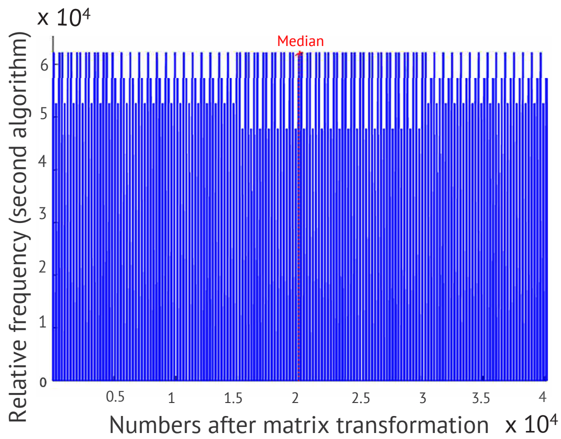


Figure 10. Relative frequency for a permutation to occur according to the second transformation algorithm, based on the results of the first experiment
Source: developed by the authors

In the experiment with input parameters $p = 23$, $M = 8$, $n = 2$, the following statistical results were obtained for the transformation executed according to the first algorithm:

- the number of generated combinations of the matrix *A*, attribute *num* in the *datastat()* function: $num = 279841$;
- the maximum value of the number resulting from transforming the matrix *A*: $max = M! - 1 = 40319$;
- the minimum value of the number resulting from transforming the matrix *A*: $min = 0$;
- the mean value in the generated sample: $mean = 19996.957$;

• the median values in the generated sample: $median = 19988$;

• the maximum number of permutation repetitions: $rep^{A1}_{max} = 7$;

• the minimum number of permutation repetitions: $rep^{A1}_{min} = 6$;

• A^{norm}_{10} values from the range [0; 37920] repeat rep^{A1}_{max} times;

• A^{norm}_{10} values from the range [37921; 40320] repeat rep^{A1}_{min} times;

• variance: $D_{A1} = 133524637.447$;

• the maximum value of the relative frequency for a permutation to occur is: $W^{A1}_{max} = 2.5014 \cdot 10^{-5}$;

• the minimum value of the relative frequency for a permutation to occur is: $W^{A1}_{min} = 2.1441 \cdot 10^{-5}$.

Figure 11 contains a bar graph for distribution of the absolute frequencies N_j, $0 \leqslant j \leqslant M! - 1$ of permutations with number $x_j = j$ in the lexicographic order according to the first transformation algorithm. The graph illustrating the relative frequency for permutations to occur according to the first transformation algorithm is demonstrated in Figure 12. *Ox* axis contains the generated values of the permutation sequence numbers when the matrix A is transformed by formula A^{norm}_{10}, and *Oy* axis is the relative frequency of the generated number. In the experiment with input parameters $p = 23$, $M = 8$, $n = 2$, the following statistical results were obtained for the transformation according to the second algorithm:

• the number of generated combinations of the matrix A, attribute *num* in the *datastat()* function: $num = 279841$;

• the maximum value of the number resulting from transforming the matrix A: $max = 40198$;

• the minimum value of the number resulting from transforming the matrix A: $min = 0$;

• the mean value in the generated sample: $mean = 20072.2193$;

• the median values in the generated sample: $median = 20162$;

• the maximum repetition number of the generated numbers (sequence number of the permutation), $rep^{B1}_{max} = 230$;

• the minimum repetition number of the generated numbers (sequence number of the permutation), $rep^{B1}_{min} = 0$;

• the number of ordinal number values of the permutations that have the maximum number of repetitions rep^{B1}_{max} is 64;

• the number of ordinal number values of the permutations that have the minimum number of repetitions rep^{B1}_{min} is 37679;

• variance: $D_{B1} = 1335421877.799$;

• the maximum value of the relative frequency for a permutation to occur: $W^{B1}_{max} = 82.1895 \cdot 10^{-5}$;

• the minimum value of the relative frequency for a permutation to occur: $W^{B1}_{min} = 28.5877 \cdot 10^{-5}$.

Figure 13 contains the bar graph showing the distribution of the absolute frequency N_j, $0 \leqslant j \leqslant M! - 1$ of permutations with number $x_j = j$ in the lexicographic order following the first transformation algorithm.

The graph illustrating the relative frequency for permutations to occur according to the second transformation algorithm is shown in Figure 14. *Ox* axis contains the generated values of the permutation sequence numbers when the matrix A is transformed by formula A^{norm}_{10}, and *Oy* axis is the relative frequency of the generated number.

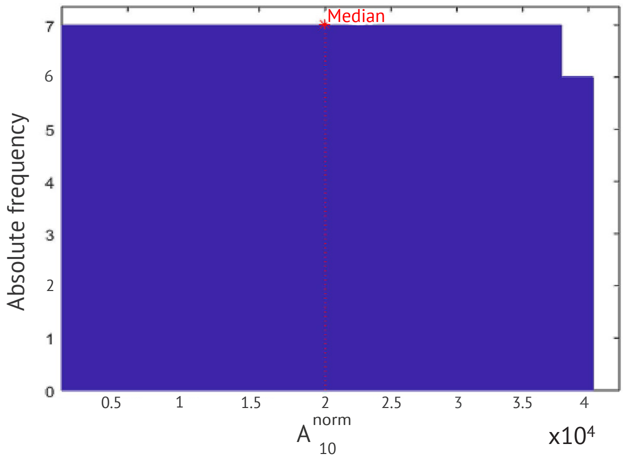


Figure 11. Absolute frequency for a permutation to occur according to the first transformation algorithm, based on the results of the second experiment
Source: developed by the authors

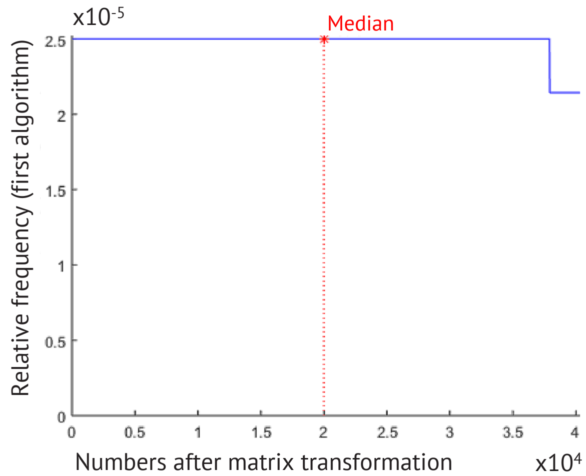


Figure 12. Relative frequency for a permutation to occur according to the first transformation algorithm, based on the results of the second experiment
Source: developed by the authors

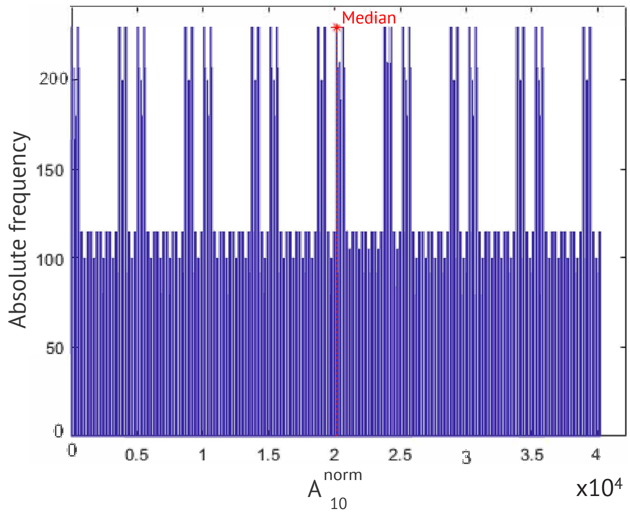


Figure 13. Absolute frequency of permutation occurrence according to the second transformation algorithm, based the results of the second experiment
Source: developed by the authors

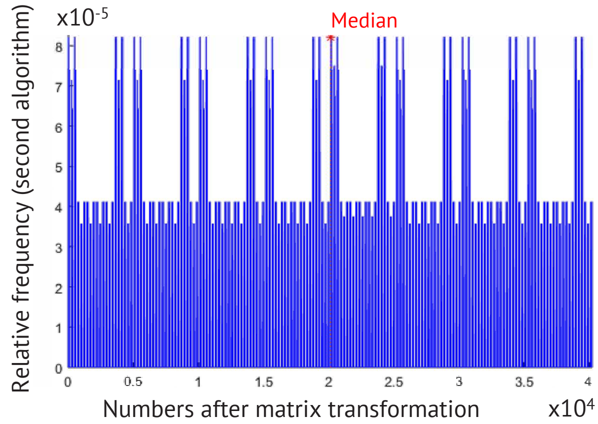


Figure 14. Relative frequency of permutation occurrence according to the second transformation (results of the second experiment)
Source: developed by the authors

The statistical results for the first and second experiments are shown in Table 1.

Table 1. Experimental statistical results

Parameter	Experiment 1		Experiment 2	
Input conditions for the experiment	$p = 17, n = 2, M = 8$		$p = 23, n = 2, M = 8$	
Matrix transformation algorithm	I	II	I	II
The volume of the generated sample,	83521	83521	279841	279841
N_{max}	40319	40198	40319	40198
N_{min}	0	0	0	0
Variance, D	142498178.84	136109307.21	133524637.45	13542187.80
Relative frequency, W_{max}	$3.5919 \cdot 10^{-5}$	$62.2597 \cdot 10^{-5}$	$2.50142 \cdot 10^{-5}$	$82.1895 \cdot 10^{-5}$
Relative frequency, W_{min}	$2.3946 \cdot 10^{-5}$	$35.9191 \cdot 10^{-5}$	$2.1440 \cdot 10^{-5}$	$28.5877 \cdot 10^{-5}$
Number of repetitions rep_{max}	3	52	7	230
Number of repetitions rep_{min}	2	0	6	0
The sample median	19439	20012	19988	20162
The sample mean	19514	19995	19996.96	20072.22

Source: developed by the authors

Transformation of the matrix A by the first algorithm generated the maximum repetitions of the permutation numbers in the lexicographic order $rep^{A1}_{max} = 3$ in the first experiment and $rep^{A2}_{max} = 7$ in the second experiment. The minimum number of the generated permutation repetitions is: $rep^{A1}_{min} = 2$ in the first experiment and $rep^{A2}_{min} = 6$ in the second experiment.

The result of transformations with auxiliary matrices (the second algorithm) is the maximum number of repeated permutation numbers $rep^{B1}_{max} = 52$ in the first experiment and $rep^{B2}_{max} = 230$ in the second experiment. The minimum number of repeated permutation numbers is $rep^{B1}_{min} = 0$ in the first experiment and $rep^{B2}_{min} = 0$ in the second experiment.

The variance value of the obtained permutation numbers in the first experiment and matrix processing according to the first algorithm is $D_{A1} = 142498178.841$. The variance value in the first experiment and the processing algorithm with auxiliary matrices is $D_{B1} = 136109307.2089$. The obtained relations $D_{A1} > D_{B1}$ indicate that random variable values in the first algorithm have larger deviations from the distribution centre, which is a prerequisite for more uniform distribution over a set of possible values. The bar graph featuring the absolute distribution of the matrix transformation according to the second algorithm (Fig. 13) shows zeros, which means that the corresponding permutation number has not been generated in the sample.

To test the null hypothesis H_0 that the received samples of permutation numbers in the lexicographic order correspond to a uniform distribution over the segment $[0; M! - 1]$, the χ^2 Pearson test was applied.

H. Chernoff & E.L. Lehmann (1954) argue that the resulting sample must be grouped into classes to correctly apply the χ^2 criterion. H.B. Mann & A. Wald (1942) found that the optimal $r = r(N, \alpha)$ number of grouping classes (data grouping intervals) based on the sample size N and the significance level α for the χ^2 criterion is defined by the equation:

$$r(N,\alpha) = 2 \cdot 2^{1.2} \cdot \left(\frac{N-1}{Norm_{0;1;\alpha}}\right)^{0.4}, \tag{7}$$

where $Norm_{0;1;\alpha}$ – the upper limit of the standard normal distribution $Norm_{0;1}$.

Later, it was established that without significant loss of the criterion's length, the $r(N, \alpha)$ value can be halved, and for large N select the following (Turchyn, 2014):

$$r(N,\alpha) = 2 \cdot 2^{0.2} \cdot \left(\frac{N-1}{Norm_{0;1;\alpha}}\right)^{0.4}. \tag{8}$$

For $\alpha = 0.05$, $Norm_{0;1;0.05} = 1.645$. The following is an analysis of the results of experiment 1 for the first matrix transformation algorithm. For N=83521 and α=0.05, the value $r(83521,0.05) \approx 175.18$. Therefore, the number of grouping intervals is $r = 175$. The length of the grouping interval is $H = \frac{40319}{175} \approx 230.3943$. The value of the interval width is taken as $H = 230.395$. Then $x_i^* = H \cdot i$, $0 \leqslant i \leqslant r$. v_i denotes absolute frequencies of grouping intervals: $v_i = \sum_{x_{i-1}^* < x_j \le x_i^*} N_j$, $0 \leqslant i \leqslant r$, and $v_1 = \sum_{x_0^* \le x_j \le x_1^*} N_j$, N_j is absolute frequency of x_j value in the i-th grouping interval. If the degree of freedom m of the χ^2_m distribution acquires large values, then by virtue of the central limit theorem, this distribution approaches the normal one with parameters $a = m$ and $\sigma^2 = 2m$:

$$\chi^2_m(x) \approx Norm_{m;2m}(x) \text{ for } m > 30. \tag{9}$$

Since $Norm_{m;2m}(x) = Norm_{0;1}\left(\frac{x-m}{\sqrt{2m}}\right)$, the equation for the upper α-limit of the distribution is $Norm_{m;2m}(x)$: $1-\alpha = Norm_{m;2m}(x_\alpha) = Norm_{0;1}\left(\frac{x_\alpha - m}{\sqrt{2m}}\right) = Norm_{0;1}(t_\alpha)$.

Then $\frac{x_\alpha - m}{\sqrt{2m}} = t_\alpha$ or:

$$x_\alpha = m + t_\alpha\sqrt{2m}\,. \tag{10}$$

The Pearson test statistics for a uniform distribution has degrees of freedom $m = r - 1$. If $\chi^2_{\alpha;m}$ is the upper α-limit of χ^2 distribution with m degrees of freedom, the H_0 hypotheses regarding the continuous uniform distribution of the sample is accepted at the level of significance α if:

$$\chi^2_{observ.} = \sum_{i=1}^{r}\frac{(v_i - Np_i)^2}{Np_i} < \chi^2_{\alpha;m}\,, \tag{11}$$

where p_i – theoretical relative occurrence frequency of the investigated random variable in the i-th grouping interval.

Worth noting that

1) $p_1 = P(0 \le X \le x_1^*) = P(x_{i-1}^* < X \le x_{i1}^*) = p_i = \frac{1}{r}$, $2 \leqslant i \leqslant r$. For $N = 83521$ and $\alpha = 0.05$, the value $p_i = \frac{1}{175}$, $1 \leqslant i \leqslant 175$;

2) for each i, $1 \leqslant i \leqslant 175$ value, $Np_i = 83521 \cdot \frac{1}{175} \gg 10$ has been performed;

3) by virtue of relations (9), (10) and (11), $\chi^2_{observ.} = \sum_{i=1}^{r}\frac{\left(v_i - \frac{N}{r}\right)^2}{\frac{N}{r}} = \frac{r}{N}\sum_{i=1}^{r}\left(v_i - \frac{N}{r}\right)^2 < \chi^2_{\alpha;r-1} \approx r - 1 + t_\alpha\sqrt{2}\sqrt{r-1}$ holds. Thus, the verification rule is:

$$\sum_{i=1}^{r}\left(v_i - \frac{N}{r}\right)^2 < \frac{N}{r}\left(r - 1 + t_\alpha\sqrt{2}\sqrt{r-1}\right). \tag{12}$$

For $N = 83521$, $\alpha = 0.05$, $t_\alpha = 1.645$, $r = 175$ hypothesis H_0 about uniform distribution of the permutation numbers in the segment [0; 40319] is conditionally accepted:

$$\sum_{i=1}^{175}(v_i - 477.26)^2 < 97689.54. \tag{13}$$

Substituting the absolute frequencies v_i obtained from experiment 1 for the first matrix transformation algorithm into (13) results in the value $\sum_{i=1}^{175}(v_i - 477.26)^2 = 603773.91$, which significantly exceeds the permissible value 97689.54. Table 2 demonstrates the results of applying the Pearson test as Equation (12) for the results of experiments from Table 1.

Table 2. Results of testing the hypothesis about the uniform distribution of permutations

Parameter	Experiment 1		Experiment 2	
Matrix transformation algorithm	I	II	I	II
Number of intervals r	175		284	
Interval width H	230.395		141.969	
Value $\sum_{i=1}^{r}\left(v_i - \frac{N}{r}\right)^2$	603773.91	1973127.91	317923.08	350321441
Critical value $\frac{N}{r}\left(r - 1 + t_\alpha\sqrt{2}\sqrt{r-1}\right)$	97689.54		317418.32	

Source: developed by the authors

As has been proved here, none of the obtained permutation distributions corresponds to the uniform distribution according to χ^2 Pearson's test. However, with increasing $\frac{p^{n^2}}{M!}$ ratio, such discrepancy disappears for the first matrix transformation algorithm. On the contrary, the second matrix transformation algorithm requires additional research to improve its statistical properties.

DISCUSSION

If the computing power of the computer system is sufficient, additional hardware is inexpedient. However, storing and transmitting the matrix for key generation becomes an issue. In this case, it is possible to select a complex approach to the organization of a secure system. Transmitting a matrix through an open, unprotected communication channel between two conversers is possible by taking the following steps:

- a square matrix is generated in accordance with the requirements of the cryptosystem;
- the transformation of the matrix into a permutation is performed through successive steps described in "materials and methods";
- an exchange protocol using permutations is implemented.

Factorial coding ensures implementing protective functions against unauthorised access, as well as detecting and correcting errors. However, the issue of matching the common key permutations persists. To solve this issue, this study proposes an approach based on matrices. The advantages of matrix calculations when solving cryptography problems include:

- complexity of analysis: matrix operations (especially matrix inversions) complicate analysing and breaking cryptographic systems, increasing their security;
- flexibility and efficiency: matrix methods allow efficient encoding and decoding of large data arrays;
- increased resistance to quantum attacks: some matrix cryptographic schemes (for example, lattice-based) are resistant to quantum computer attacks, which enhances their prospects for future use.

Matrix operations, such as multiplication or inversion of large matrices, are computationally complex. This may lead to a significant increase in processing time and resource consumption, especially in cases with large matrix sizes. Note that transforming a matrix according to the first algorithm through representing matrix *A* in decimal and factorial notations requires significant computer system resources. In particular, a way to achieve a uniform distribution is to increase the $\frac{p^{n^2}}{M!}$ ratio, which, as a result, is able to increase the amount of time required for calculations. However, this flaw can be insignificant if part of the calculations is distributed and transferred to the hardware component of the information exchange system. One of the examples of realising this approach is offered in a study by M. Issad *et al.* (2020). The authors proposed to transfer encrypting and decrypting server's algorithms to a programmable gate array (FPGA). Another work by S. Sheikhpour *et al.* (2021) proposes two schemes based on a combination of hardware and time reservation: a novel hardware reservation for the cyclic AES function and time reservation for the AES key extension module hardware.

Another limitation of matrices may be related to the high memory volume requirements by information processing tools, since processing matrices requires significant amounts of memory, especially large dimensional matrices or a significant quantity of matrix operations. This may be problematic for resource-constrained devices, such as embedded devices and especially mobile devices. Thus, applications of the proposed approaches in low-resource cryptography should be investigated in more detail. Key management procedures can also be complicated. In case of matrix cryptosystems, matrices are the keys. Managing these keys (generation, storage, transmission) is more complex compared to traditional encryption algorithms, such as Advanced Encryption Standard (AES) (National Institute for Standards and Technology, 2001), where keys are scalar values. In addition, encryption and decryption algorithm offered by Z.Y. Karatas *et al.* (2019) involves finding an inverse matrix, which is a complicated computational task. Cryptosystems using lattice-based encryption have a similar property (Li *et al.,* 2024a).

Regarding the issue of cryptographic stability, matrix cryptosystems can be stable. For example, a study by H. Huang *et al.* (2022) emphasises that, based on the results of the research, an algorithm based on ring tropical matrices can be considered as a new post-quantum cryptosystem. However, the specific properties of matrices used for decryption can increase efficiency of certain cryptographic analysis methods, for example, attacks based on the matrix structure. An important stage in investigating cryptographic stability of matrix cryptosystems is hence analysing vulnerabilities to attacks performed through specialised methods, for example, lattice-based analysis or other algebraic methods.

In addition to the above, matrix cryptosystems may be vulnerable to linear dependencies between matrix elements. This can degrade the system stability and increase its chances of being hacked. To combat linear dependencies between elements, various approaches are used, which can be attributed to: nonlinear transformations (in AES and DES cryptosystems, s-blocks are deployed that perform bit transformations), which as a result reduce linear dependencies between input and output elements; another approach to reduce linear dependencies is to pad with redundant data or use longer keys to increase the key space; use of algebraic transformations using matrices with high entropy; adding randomness during key generation. An example is a study by M. Gafsi *et al.* (2020).

Note that matrix cryptosystems work with data blocks of fixed size, which can impede processing messages that are not a multiple of the block size and require additional mechanisms to overcome the barrier (such as adding placeholders). The existing methods of adding placeholders include PKCS No. 5/7, ANSI X.923, ISO/IEC 9797-1, bit padding, filling the message content to the required block size with bytes of zeros. Cryptosystems applying these techniques are AES, which often use PKCS #7, or DES, which may use PKCS #5 or ANSI X.923. However, implementing the mentioned mechanism in the cryptosystem reduces its stability because of possible attacks targeted at this aspect, for instance, a boomerang type attack proposed back in 1998 (Bleichenbacher Attack Explained, 2019) and its improved algorithm (Kelesidis, 2022). In the second algorithm for converting a matrix into a permutation proposed in this paper, a mechanism for solving the backfill problem is introduced for the value of the permutation length M, which is greater than the number of elements of the matrix. Nonetheless, considering the possible identified risks, it is necessary to provide for a flexible encryption mechanism that would not depend on the length of the incoming message in the future.

Matrix fields are a powerful tool in cryptography, offering efficient and robust methods for protecting information. Involving matrices in cryptographic algorithms assists in constructing complex and reliable encryption systems that satisfy modern security requirements under the condition of choosing the appropriate algorithms, matrix sizes and protection methods. Nonetheless, high computational complexity can be attributed to the potential disadvantages of using matrices in cryptographic information exchange systems; high requirements for the volume of memory used; complexity of key management; vulnerability to certain types of attacks; message length and block size.

CONCLUSIONS

This study attempted to integrate methods of matching matrix key elements with factorial data coding, which relies on permutations to transfer information. For this purpose, a software model for transforming the square matrix A into the permutation π was developed and implemented. Two matrix transformation algorithms have been considered and tested: by representing the matrix A in decimal and factorial notations and by using auxiliary square matrices. The first transformation algorithm involves step-by-step transformation of matrix elements into coefficients of positional (decimal and factorial) notations. The second transformation algorithm involves successive multiplication from the left of the matrix A by all auxiliary matrices followed by generating factorial coefficients from the elements of the product matrices.

Analysis of existing cryptosystems built on matrices has revealed both advantages and disadvantages in applying matrix transformations to generate key elements of factorial coding. The study has also determined new vectors of developing and improving matrix transformations in systems with factorial data coding as a tool for key agreement. A potential weak point to implement an attack on a cryptographic system is encryption, which takes place in blocks of a fixed size. Despite the above-mentioned shortcomings, combinations of matrix cryptographic systems and factorial data coding can be effective and safe, provided they are used correctly, and possible risks are taken into account. Selecting appropriate algorithms, matrix sizes, and protection methods, as well as algorithms for converting matrices into permutations, can help minimise these flaws and ensure reliable information security.

Two experiments have been conducted for different dimensions of Z_p finite field of matrix elements: $p = 17$ and $p = 23$. Histograms of permutation distributions at the output of the simulated model have been constructed. The results of the study indicate that the transformation algorithm that involves representing the matrix A in decimal and factorial notations for large $\frac{p^{n^2}}{M!}$ has proved that this method can achieve a uniform distribution of possible permutations and can be used as a means of generating keys in systems with factorial coding. The second transformation algorithm with auxiliary matrices does not ensure a uniform distribution of the generated permutations and hence requires further research.

ACKNOWLEDGEMENTS

This research was funded by the Ministry of Education and Science of Ukraine under grant 0123U100270.

CONFLICT OF INTEREST

None.

REFERENCES

[1] Al-Shaarani, F., & Gutub, A. (2022). Securing matrix counting-based secret-sharing involving crypto steganography. *Journal of King Saud University - Computer and Information Sciences*, 34(9), 6909-6924. doi: 10.1016/j.jksuci.2021.09.009.

[2] Bleichenbacher Attack Explained. (2019). Retrieved from https://medium.com/@c0D3M/bleichenbacher-attack-explained-bc630f88ff25.

[3] Chernoff, H., & Lehmann, E.L. (1954). The use of maximum likelihood estimates in x^2 tests for goodness of fit. *The Annals of Mathematical Statistics*, 25(3), 579-586. doi: 10.1214/aoms/1177728726.

[4] Gafsi, M., Abbassi, N., Hajjaji, M.A., Malek, J., & Mtibaa, A. (2020). Improved chaos-based cryptosystem for medical image encryption and decryption. *Scientific Programming*, 2020, article number 6612390. doi: 10.1155/2020/6612390.

[5] Greub, W. (1975). *Linear algebra*. New York: Springer. doi: 10.1007/978-1-4684-9446-4.

[6] Higham, D.J., & Higham, N.J. (2005). *MATLAB guide*. Philadelphia: Society for Industrial and Applied Mathematics.

[7] Huang, H., Li, C., & Deng, L. (2022). Public-key cryptography based on tropical circular matrices. *Applied Sciences*, 12(15), article number 7401. doi: 10.3390/app12157401.

[8] Issad, M., Anane, N., Bellemou, A.M., & Boudraa, B. (2020). Secure hybrid crypto-system AES/RSA on FPGA for data communication. *Malaysian Journal of Computing and Applied Mathematics*, 3(1), 1-10. doi: 10.37231/myjcam.2020.3.1.38.
[9] Joshi, S., Bairwa, A.K., Pljonkin, A.P., Garg, P., & Agrawal, K. (2023). From pre-quantum to post-quantum RSA. In *Proceedings of the 6th international conference on networking, intelligent systems & security* (pp. 1-8). doi: 10.1145/3607720.3607721.
[10] Kaptiol, Ye., & Horbenko, I. (2020). Analysis of the possibilities and peculiarities of programming cryptology problems on a quantum computer. *Radiotekhnika*, 3(202), 37-48. doi: 10.30837/rt.2020.3.202.03.
[11] Karatas, Z.Y., Luy, E., & Gonen, B. (2019). A public key cryptosystem based on matrices. *International Journal of Computer Applications*, 182(42), 47-50. doi: 10.5120/ijca2019918432.
[12] Kelesidis, E.-A. (2022). An optimization of Bleichenbacher's oracle padding attack. In P.Y. Ryan & C. Toma (Eds.), *Innovative security solutions for information technology and communications* (pp. 145-155). Cham: Springer. doi: 10.1007/978-3-031-17510-7_10.
[13] Lavdanskyi, A., Faure, E., Skutskyi, A., & Bazilo, C. (2023). Accelerating operations on permutations using graphics processing units. In E. Faure, O. Danchenko, M. Bondarenko, Y. Tryus, C. Bazilo & G. Zaspa (Eds.), *Information technology for education, science, and technics* (pp. 3-12). Cham: Springer. doi: 10.1007/978-3-031-35467-0_1.
[14] Li, J., Yan, M., Peng, J., Huang, H., & El-Latif, A.A.A. (2024a). A lattice-based efficient certificateless public key encryption for big data security in clouds. *Future Generation Computer Systems*, 158, 255-266. doi: 10.1016/j.future.2024.04.039.
[15] Li, P., *et al.* (2024b). Scalable parallel ultrafast optical random bit generation based on a single chaotic microcomb. *Light: Science & Applications*, 13(1), article number 66. doi: 10.1038/s41377-024-01411-7.
[16] Mann, H.B., & Wald, A. (1942). On the choice of the number of class intervals in the application of the Chi square test. *The Annals of Mathematical Statistics*, 13(3), 306-317. doi: 10.1214/aoms/1177731569.
[17] Maturin, Yu., Komarnytska, L., & Hordiienko, I. (2023). *Discrete math*. Drohobych: Drohobych Ivan Franko State Pedagogical University.
[18] Maxrizal, M. (2022). Public key cryptosystem based on singular matrix. *Trends in Sciences*, 19(3), article number 2147. doi: 10.48048/tis.2022.2147.
[19] Naseri, A.R., Abbasi, A., & Atani, R.E. (2023). A new public key cryptography using M_q matrix. *Journal of Mathematical Modeling*, 11(4), 681-693. doi: 10.22124/jmm.2023.23982.2142.
[20] National Institute for Standards and Technology. (2001). *Specification for the Advanced Encryption Standard (AES)*. Retrieved from https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.197.pdf.
[21] Rani, S., Bhambri, P., Kataria, A., Khang, A., & Sivaraman, A.K. (2023). *Big data, cloud computing and IoT: Tools and applications*. Boca Raton: Chapman and Hall/CRC. doi: 10.1201/9781003298335.
[22] Sheikhpour, S., Mahani, A., & Bagheri, N. (2021). Reliable advanced encryption standard hardware implementation: 32- bit and 64-bit data-paths. *Microprocessors and Microsystems*, 81, article number 103740. doi: 10.1016/j.micpro.2020.103740.
[23] Shvydkyi, V., Shcherba, A., Kharin, O., Lavdanskyi, A., & Faure, E. (2021). *Basics theory of inseparable factorial data coding*. Kharkiv: Novyi Kurs.
[24] Turchyn, V. (2014). *Probability theory and mathematical statistics*. Dnipro: IMA-Press.
[25] Valentine, D.T., & Hahn, B.H. (2023). *Essential MATLAB for engineers and scientists*. London: Academic Press.

Програмна модель формування ключів-перестановок через квадратну матрицю

Еміль Фауре

Доктор технічних наук, професор
Черкаський державний технологічний університет
18006, бульв. Шевченка, 460, м. Черкаси, Україна
https://orcid.org/0000-0002-2046-481X

Анатолій Щерба

Кандидат фізико-математичних наук, доцент
Черкаський державний технологічний університет
18006, бульв. Шевченка, 460, м. Черкаси, Україна
https://orcid.org/0000-0002-3049-3497

Артем Скуцький

Аспірант
Черкаський державний технологічний університет
18006, бульв. Шевченка, 460, м. Черкаси, Україна
https://orcid.org/0000-0002-8632-1176

Артем Лавданський

Кандидат технічних наук, доцент
Черкаський державний технологічний університет
18006, бульв. Шевченка, 460, м. Черкаси, Україна
https://orcid.org/0000-0002-1596-4123

Анотація. Одним із важливих аспектів, які необхідно розвивати у 21 столітті, є безпека та захист інформації. Для криптографічних алгоритмів розповсюдженим підходом є використання матричного представлення інформації. Разом з тим, нові підходи можуть використовувати інші структури даних, в тому числі перестановки. Це викликає необхідність узгодження структур даних для інтеграції різних методів у одній системі обробки та передавання інформації. Метою роботи є формування перестановок, що є ключем для факторіального кодування даних, за відомою ключовою матрицею. У роботі представлено два алгоритми перетворення квадратної матриці в перестановку. Наведено приклад перетворення матриці за кожним з запропонованих алгоритмів. Створено та описано програмну модель для дослідження перетворення квадратних матриць у перестановки з використанням програмного продукту Matlab. Розглянуто вбудовані методи обробки статистичної інформації в програмі Matlab та графічного їх відображення за допомогою вбудованих функцій, які використані в процесі роботи програмної моделі. Виконано перетворення матриці за запропонованими алгоритмами. Досліджено всі можливі комбінації квадратної матриці порядку 2 з елементами, що належать скінченному полю цілих чисел за модулем $p = 17$ та $p = 23$. За кожним алгоритмом перетворення отримано результати перетворення квадратної матриці у номер перестановки в лексикографічному порядку їх слідування. Досліджено статистичні властивості отриманих результатів, визначено найбільш ефективний алгоритм перетворення матриць у перестановки за критерієм рівномірності розподілу отриманих номерів перестановок. Показано, що цей алгоритм має перспективу використання в системах обміну інформації з використанням факторіального кодування даних

Ключові слова: факторіальне кодування даних; захист інформації; узгодження ключів; статистика; рівномірний розподіл

BULLETIN of Cherkasy State Technological University

Journal homepage: https://bulletin-chstu.com.ua/en

Vol. 29 No. 2. 2024

UDC 681.7, 535.8
DOI: 10.62660/bcstu/2.2024.24

Characterisation of optical waveguides for photonic integrated circuits

Iryna Avdieionok*

Postgraduate Student
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
03056, 37 Berestetskyi Ave., Kyiv, Ukraine
https://orcid.org/0009-0007-4709-0030

Volodymyr Borovytsky

Doctor of Technical Sciences, Professor
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
03056, 37 Berestetskyi Ave., Kyiv, Ukraine
https://orcid.org/0000-0001-6816-0391

Abstract. Fast signal processing at the speed of light is the main advantage of photonic integrated circuits. Therefore, these circuits have good prospects for the implementation of mathematical calculations, including matrix to vector multiplication. The purpose of the research was to create and investigate a technique for automatic measurement of brightness distribution along optical waveguides of analogue photonic integrated circuits. Empirical methods (observation, measurement, comparison, experiment) and a complex method (analysis and synthesis) have been used during the research. The proposed technique uses a digital camera that captures images of optical waveguide illuminated by light emitting diodes and image processing software to calculate brightness distribution. This technique determines the best approximation of this distribution, calculates parameters of brightness non-uniformity and losses of optical radiation. Measurements of a set of optical waveguides help to identify the best candidates for photonic integrated circuits. It has been found that optical waveguides with grinded surfaces acting as diffusive scattering have good combination of smooth brightness distribution and small losses of optical radiation. Due to multiple diffuse reflection and scattering within waveguide material, these waveguides are promising candidates for analogue photonic integrated circuits. All other waveguides with non-processed surface, with grooves or grinded with a large grain have sufficient losses of optical radiation. These losses are usually caused by the exit of optical radiation from waveguide surface. The obtained results are necessary for accurate design of circuits that takes into account scattering and losses in optical waveguides. The proposed technique can be applied in automatic technological process of manufacturing a fast and economical photonic matrix to vector multiplication, which does not require expensive electron-beam, optical or laser lithographic equipment

Keywords: brightness distribution; automatic measurement; image processing; uniformity evaluation; losses evaluation; photonic matrix to vector multiplication; optoelectronics

INTRODUCTION

As global data consumption increases and demand for faster networks continues to grow, the world needs to find more sustainable solutions to the energy crisis. Expanding of data networks and data centres, safer autonomous vehicle driving, and more efficient manufacturing cannot be sustained by electronic microchip

Article's History: Received: 06.01.2024; Revised: 21.04.2024; Accepted: 27.05.2024.

Suggested Citation:
Avdieionok, I., & Borovytsky, V. (2024). Characterisation of optical waveguides for photonic integrated circuits. *Bulletin of Cherkasy State Technological University*, 29(2), 24-31. doi: 10.62660/bcstu/2.2024.24.

*Corresponding author

technology alone. However, combining of electrical devices with embedded photonics through heterogeneous integration provides a more energy-efficient way to increase the speed and capacity of data networks, reduce costs, and meet an increasingly diverse range of needs in industries.

Photonic integrated circuits (PICs) are considered as promising devices for telecommunication, signal and image processing, mathematical calculation, hardware neural network implementation for deep learning, and others at the speed of light (Luo *et al.*, 2023). Many PIC applications including signal filtration, integral transform calculations, and neural network implementation require high-speed matrix to vector multiplication (MVM). According to the classification by architecture proposed in the publication of V. Borovytsky *et al.* (2022), they can be divided into three main groups: PICs with Mach-Zehnder interferometres (MZIs), PICs with microring resonator arrays (MRRAs), a combination of two above mentioned groups, and PICs based on the application of non-linear optical effects.

Another approach is proposed by J. Cheng *et al.* (2021) – MVM using two-dimensional MRRA. The entire architecture is based on wavelength division multiplexing and a reconfigurable MRRA, which forms a full 4 × 4 transfer matrix network. Each microring resonator can be adjusted using a miniature electrical heater. Heating changes the transmission spectrum of the microring resonator. It allows to change matrix coefficients for MVM. Researchers C. Huang *et al.* (2020) have demonstrated an experimental MRRA PIC that is able to perform MVM with good accuracy. However, the biggest and scalable MRRA PIC for hardware implementation of large neural networks is described by S. Ohno *et al.* (2022). It is worth noting that MRRA PICs are considered as one of the promising PICs for MVM. This is confirmed by a large number of publications in prestigious scientific journals. Nevertheless, no such MVM PICs are manufactured serially due to high sensitivity of MRRAs to any variations of technological regimes and operating conditions, such as PIC temperature, etc.

PIC technological process that requires electron beam, optical or laser lithographic equipment, equipment for layer deposition and etching, scanning electron microscopes for failure analysis, and specific materials has been described by J. Cheng *et al.* (2022). Economical MVM PIC architecture that can be manufactured without expensive equipment has been proposed by V. Borovytsky & I. Avdieionok (2024). This PIC performs analogue multiplication of input signal to matrix, which is used for signal processing and neural network implementation. However, the accuracy of this multiplication depends on the quality of optical waveguides, mainly on brightness distribution along them.

The purpose of the study was to develop a technique for fully automatic measurement of brightness distribution along optical waveguides for the proposed MVM PIC, which could be used in serial production of such PICs in Ukraine. The list of tasks includes the selection of a technique for measurements of brightness distribution, measurement of brightness distribution in optical waveguides, and analysis of measurement results from the point of view of possible application in MVM PICs.

MATERIALS AND METHODS

The developed PIC for MVM has three layers (Fig. 1). The layer of input optical waveguides is used to read the input optical signal and form a stable brightness distribution. The second layer with a fixed aperture array provides the transmission of optical radiation from the first layer to the third one. As a result, optical flows that have passed through the apertures become proportional to aperture areas and a layer with output optical waveguides. The third layer has optical waveguides that accumulate optical radiation that has passed through the apertures. The outputs of these waveguides are proportional to the MVM results. This PIC performs analogue multiplication. Therefore, the accuracy of this multiplication depends on the quality of optical waveguides, mainly on brightness distribution along them.

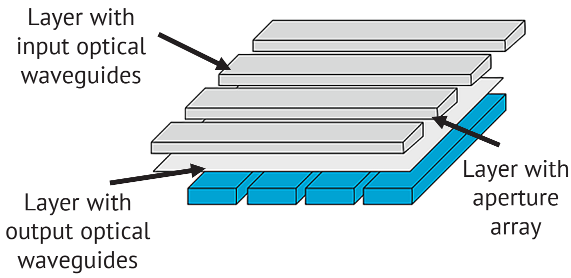


Figure 1. Analogue PIC for MVM
Source: compiled by the authors

There are several approaches to the measurement of brightness distribution. The first one is two-dimensional mechanical scanning by a photodetector on waveguide surface. This method is very slow and expensive because it requires motorised stages with a controller and a signal capture board. The second method is one-dimensional scanning with a linear photodetector, which is used in wide-spread scanners. It is also slow and expensive due to the use of one motorised stage with a controller, a linear photodetector and a signal capture board. The third method is based on the use of digital cameras without mechanical scanning. Researchers K. Bahali *et al.* (2018) have shown that digital cameras can be successfully used to measure brightness distributions. The work of B. Wojcik & M. Żarski (2021) presents a technique for identifying zones with different optical properties in building construction. These studies have become the starting point for the development of an optical setup for waveguide photometric measurements.

The proposed optical setup contains an optical waveguide illuminated by light emitting diodes (LEDs) at both ends and a digital camera that takes digital photographs of this waveguide (Fig. 2). This camera has a lens with a focal length in the range of 18-55 mm and a 24.1 megapixel image sensor of 22.3 × 14.9 mm. The distance from the camera to the waveguide is 300 mm. At this distance, the field of view of the camera covers the entire surface of the waveguide.

Figure 2. Optical setup for measurements of brightness distribution

Source: photo by the authors

Captured images have been processed by software written in Python (Dey, 2020). Software has calculated average brightness value along waveguide axis in waveguide surface zone and presented measurement results in the form of diagrams and tables. To evaluate and compare brightness distributions, the following approximation, which takes into account the effect of scattering and loss of optical radiation in optical waveguide, is proposed (1):

$$L(X) = (1 - b) \cdot exp(-p \cdot X) + b, \qquad (1)$$

where b – a coefficient characterising the constant level due to scattering; p – a coefficient characterising optical absorption losses; $L(X)$ – the distribution of normalised brightness as a function of normalised waveguide length X when it is illuminated by a LED from one end.

Then the distribution of normalised brightness when illuminated by two identical LEDs on both sides has the following form (2):

$$L_2(X) = (1 - b) \cdot \{exp(-p \cdot X) + exp(-p \cdot (1 - X))\} + 2 \cdot b, \qquad (2)$$

where $L_2(X)$ – the distribution of normalised brightness when illuminated on both sides as a function of normalised waveguide length X.

To find the best approximation, it is necessary to find the (p, b) pair that corresponds to the minimum value of the root-mean-square difference between the result of measuring $L_M(X)$ and the expression $L_2(X)$ (2):

$$e(p,b) = \sqrt{\frac{\sum_{k=0}^{N-1}\left(L_M(X_k) - L_2(X_k,p,b)\right)^2}{N}} \xrightarrow{p,b} min, \qquad (3)$$

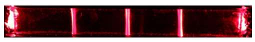
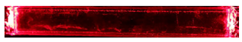
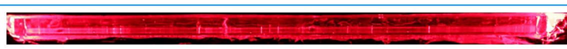
where e – the square root difference as a function of p and b parametres; $L_M(X_k)$ – the measured distribution of normalised brightness when illuminated by two LEDs on both sides as a function of normalised discrete coordinates X_k; N – the number of discrete coordinates where brightness has been measured.

The proposed approximations make it possible to choose optical waveguides applicable in analogue PICs, since the p parameter characterises brightness uniformity, and the b parameter reflects optical losses.

RESULTS AND DISCUSSION

The proposed technique (1)-(3) has been applied for automatic measurements of brightness distribution in nine plastic optical waveguides. These measurements have been performed using the proposed optical setup (Fig. 2) in a dark room. It is necessary to avoid all other illumination sources except optical waveguide. Nine samples have been studied (Table 1).

Table 1. Measured optical waveguides

Sample	Type of surface processing	Photos of the waveguide
1	Grinding with a large grain	
2	Grinding and making of two longitudinal grooves	
3	No surface processing, only three transverse grooves	
4	No surface processing	
5	Grinding with a fine grain	
6	Matted surface, no grinding applied	
7	No surface processing	
8	Grinding of all surfaces	
9	Grinding of only two sides	

Source: compiled by the authors

The first six samples have a matte material without additional surface processing. Other waveguides have a transparent material, but with a different surface processing. Sample 4 has a smooth surface without additional processing, sample 5 has a surface grinded with a fine grain, sample 1 is grinded with a large grain. Sample 2 has a grinded surface with two small longitudinal grooves. Sample 3 has a smooth surface with three transverse grooves. Samples 7-9 are rotated to reduce the radiation zone and improve the output parameters. Sample 7 has both surfaces without additional processing. The other waveguide has an unprocessed surface and the other side is grinded in sample 8. In sample 9, both sides are grinded. Waveguide dimensions are presented in Table 2.

The images of waveguide samples are processed and brightness distribution is calculated. Diagrams of these distributions are presented in Figure 3.

Table 2. Waveguide parameters

Parameter name	Parameter value
Wavelength	700 nm
Optical waveguides of 1-6 dimensions	80 × 10 × 3 mm
Optical waveguide of 7-9 dimensions	80 × 3 × 10 mm

Source: compiled by the authors

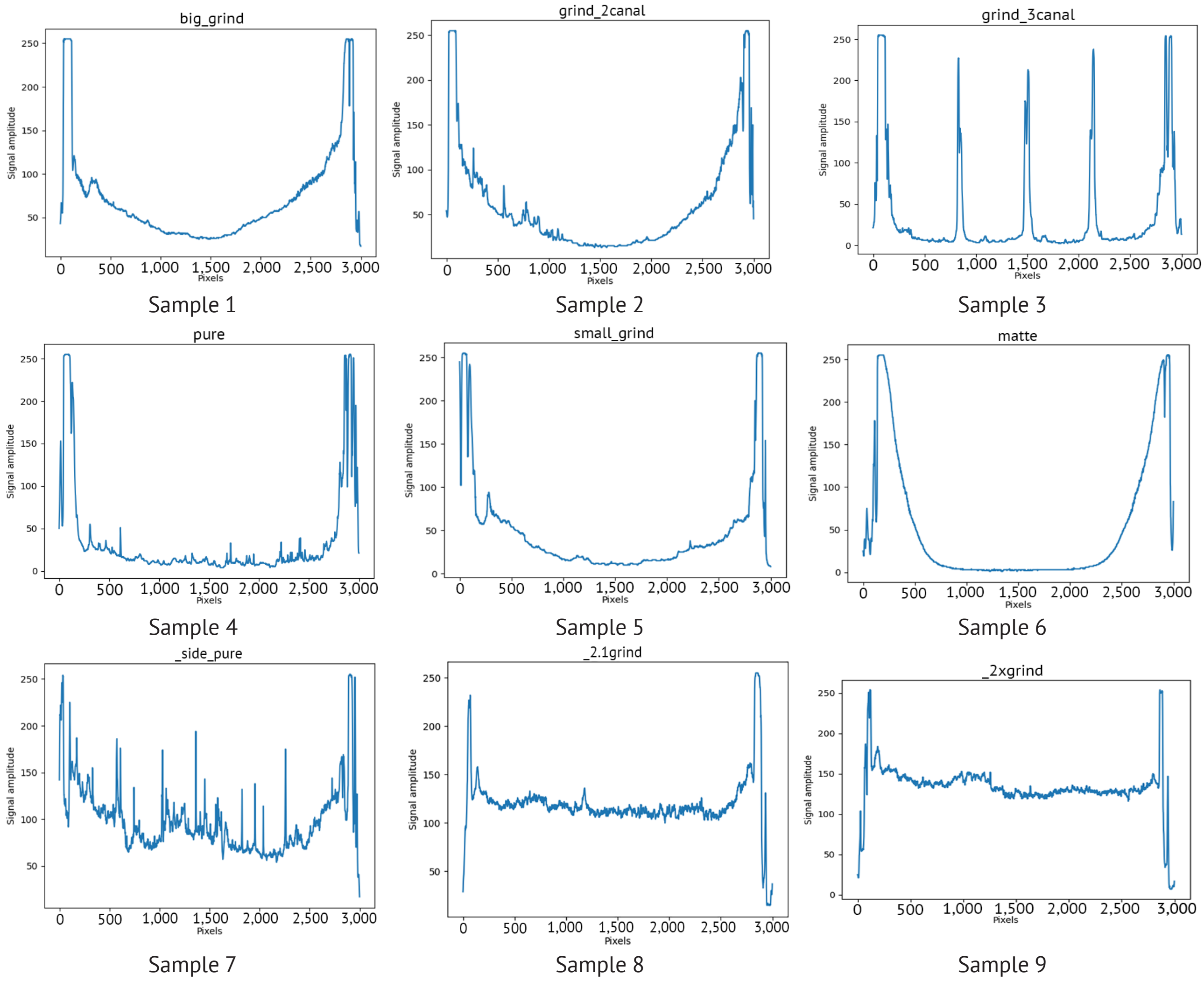


Figure 3. Diagrams of the measured brightness distribution
Source: compiled by the authors

Based on the obtained diagrams, the worst uniformity has been determined in matte waveguide. The best uniformity options are obtained with waveguides 8-9 with variable spatial placement of the waveguide to reduce the area and increase the signal level. But the waveguide 6 turns out to be the worst option due to high density of waveguide material. It can be seen from the constructed graphs that spatial placement to reduce the area of interaction is the main factor for improvement. By creating periodic grooves on the interaction surface, it will be possible to control more complex mathematical functions of signal transmission, which will allow to expand the functionality of the system.

(p, b) parameters have been calculated for all samples according to formulas (1)-(3) and presented in Table 3. It is worth noting that the proposed approximation by exponential functions (1)-(3) gives results close to the measured data for samples with grinded surfaces, such as samples 1, 2, 4, 5, 8, 9 (Fig. 3).

Table 3. Calculations of brightness distribution parameters

Sample	min (e(p,b))	p	b
1	0.073	148.601	0.015
2	0.057	309.240	0.003
3	0.149	501.012	0.015
4	0.086	166.673	0.011
5	0.050	341.368	0.003
6	0.048	40.169	0.001
7	0.111	52.218	0.196
8	0.083	32.138	0.0713
9	0.096	32.138	0.011

Source: compiled by the authors

It has been found that optical waveguides (samples 8 and 9) with grinded surfaces can be considered as good candidates for the proposed PIC (Fig. 4). The processed surfaces act as diffusive scattering. Therefore, a good combination of smooth brightness distribution and small losses of optical radiation is achieved (Tables 3, 4). Due to multiple diffuse reflection and scattering within waveguide material, these waveguides are promising candidates for analogue photonic integrated circuits.

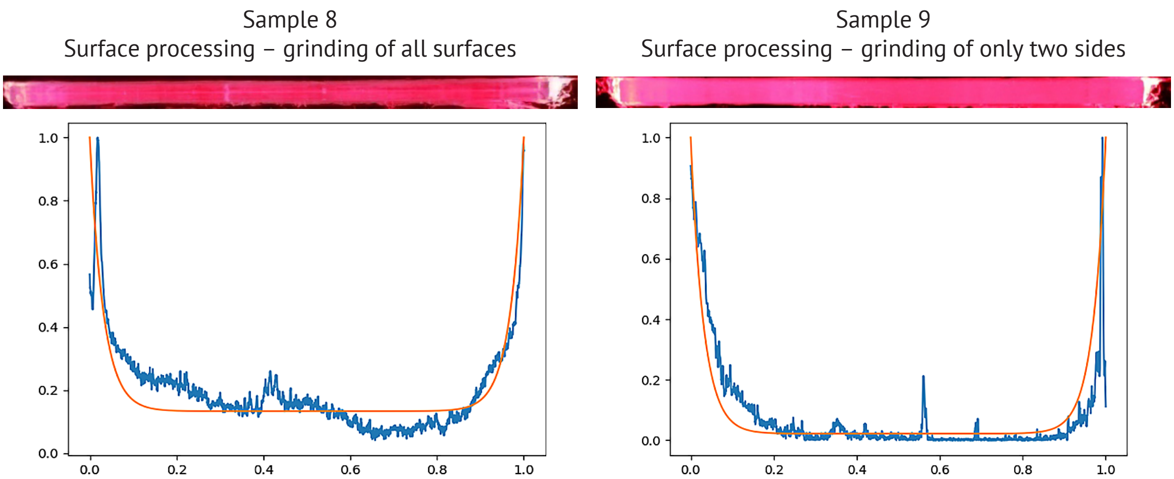


Figure 4. Measured data and their approximation

Source: compiled by the authors

All other waveguides with non-processed surfaces, with grooves or grinded with a large grain have sufficient losses of optical radiation. These losses, generally caused by the exit of optical radiation from waveguide surface, are so sufficient that in central zone of optical waveguides, the brightness decreases to almost 1% of the values (Fig. 3; Table 3).

The proposed method (1)-(3) makes it possible to obtain measurement results in a short time without expensive equipment. One measurement with image processing takes only several minutes. When optical setup is assembled, it takes one hour to perform measurement and compose the report with diagrams and tables (Fig. 3; Table 3). But routine application of the proposed technique, which is necessary for further research, requires the following improvements. First, it has to apply a monochrome digital camera sensitive in visible and infrared ranges of optical radiation. This helps to investigate optical waveguides in a wide range of optical radiation and capture digital images with a high signal-to-noise ratio. Second, it is necessary to design a cover to protect the measurement zone from external optical radiation, a stable fixed holder for the digital camera to avoid repletion of focusing operation, and a manipulator that moves and fixes one or more optical waveguides in the measurement zone. Third, it has to develop software that can process images of several optical waveguides in the field of view with automatic boundary detection and automatic quality control.

The analysis of the measurement results has shown the following: of nine investigated waveguides, sample 8 is the best candidate for PIC, as it has the largest value of $b \approx 7\%$ with good coincidence with the approximating data (Fig. 4). These results are achieved by grinding the entire waveguide surface. This sample must be considered as a starting point for further search for PIC waveguides. There are three factors that reduce waveguide characteristics: non-uniformity of waveguide material, non-uniformity of surface processing, and losses in waveguide interface of LEDs. The correct choice of plastic and its surface processing can reduce negative impact of the first and second factors. The design of waveguide interface of LEDs can be improved by grinding LED emitting surface and pressing it directly to grinded plane of waveguide side. LED measurements also play an important role. It is worth noting that non-uniformity of brightness distribution can be partially compensated by the design of corresponding layer of aperture array (Strąkowska *et al.*, 2024). This means that after manufacturing the layers with optical waveguides, they must undergo quality control according to the proposed technique, and the measurement data must be taken into account when designing the aperture array.

Many articles describe unique PIC approaches. They highlight the advantages of photonics in the context of speed, efficiency and scalability of computing, but also face challenges of implementation and integration. PICs based on the Mach-Zehnder interferometre (MZI) principle have complex implementations, especially when scaling to larger systems. However, MZI PICs provide fast and efficient solutions to various problems. Thus, in the paper by W. Bogaerts *et al.* (2020) programmable photonic PIC uses waveguides meshed with 2 × 2 blocks, or "analogue gates" – crystal-embedded equivalent of free-space optical beam splitters based on MZI. Researchers H. Zhou *et al.* (2020) have developed a nanophotonic processor for MVM that is programmed by setting the voltages on internal and external phase shifters for each MZI. Each MZI performs 2 × 2 unitary transformations over input shape state. The kernel, namely the unitary matrix, can be decomposed into sets of turns implemented by cascaded programmable MZIs. Applying this approach, positive real-value matrix computation, optical routing, and low-loss optical power distribution can be achieved.

A PIC using microring resonators requires adaptation to different architectures, but such a system is compact, efficient and scalable for high-performance computing. Mixed systems using nonlinear optics are widespread because they have high speed and recognition accuracy. However, high-speed data processing can require significant energy resources. Designing and customizing of high-speed chips is a technologically challenging task. The most promising approaches include high-speed and energy-efficient computing solutions that can significantly affect the development of artificial intelligence technologies and other advanced industries. A paper by Y. Zhu *et al.* (2024) discusses an architecture based on silicon photonic integrated circuits for accelerating neural networks. The proposed system uses optical frequency combs to achieve high computing speed and energy efficiency. The architecture provides scalability and can be used in high-performance computing applications. W. Wu *et al.* (2024) have presented an optical parallel computing array chip (OPCA chip) for fast image processing, transmission, and reconstruction. Using high-quality resonator channels, the chip provides high speed and efficiency of calculations without frequent optical-electronic and analogue-digital conversions. The obtained measurement results prove that the proposed MVM PIC can be manufactured using grinded optical waveguides. Compared to complex PICs with a waveguide matrix proposed by W. Bogaerts *et al.* (2020) and analogue PICs with microring arrays investigated by C. Huang *et al.* (2020) and V. Bangari *et al.* (2020), the proposed PIC can be produced from plastic waveguides using mechanical processing and without expensive lithographic equipment. The disadvantages of the MVM PIC proposed by V. Borovytsky & I. Avdieionok (2024) consist in sufficient dimensions of centimetre range, limited sense of input vector, and calculation errors caused by analogue multiplications. Nevertheless, the proposed PIC can potentially implement neural networks with a number of neurons up to 20.

For the production of any analogue MVM PIC, characterisation of optical waveguides plays an important role, since the accuracy of analogue multiplication directly depends on brightness distribution. The proposed technique makes this characterisation fast and economical. It can be used by Ukrainian companies, universities and research groups to create and research their own PICs. It is important that this characterisation can be applied in optical education to attract students to optics and photonics.

CONCLUSIONS

The proposed technique for measuring of brightness distribution helps to characterise optical waveguides quickly, economically and without specialised equipment. It applies only an optical waveguide illuminated by two LEDs on both sides, a digital camera, and a free software package. The application of a monochrome camera sensitive in visible infrared range can expand the types of characterised optical waveguides. This gives good prospects for its implementation into serial PIC production. The proposed approximation parameters (*p, b*) allow a clear and objective evaluation of brightness uniformity and optical losses, which can be used to select the best waveguide surface processing. The analysis of the measurement results shows that only waveguides with grinded surfaces, which work as diffusive scatters, can meet the PIC requirements for MVM. Such waveguides have stable scattering characteristics with acceptable losses of optical radiation. In other words, waveguide surfaces must play the role of elements that carry out multiple scattering of optical radiation to achieve maximum brightness uniformity.

Further research on analogue MVM PICs can be focused in three directions. The identification of optimal grinding parameters in the manufacture of optical waveguides, which guarantees the highest value of the *b* parameter and the stability of the *p* parameter, is the first one. The design of optical interface between input optical waveguides and LEDs and output optical waveguides and photodetector, which minimises optical losses and maximises the uniformity of brightness distribution, is the second one. Technological control of waveguide layers after their assembling is the third one. In all cases, the proposed technique for characterisation of optical waveguides must play an important role.

ACKNOWLEDGEMENTS

None.

CONFLICT OF INTEREST

None.

REFERENCES

[1] Bahali, K., Samian, A.L., Muslim, N., & Hamid, N.S.A. (2018). Measuring luminance of dawn with a DSLR. *International Journal of Civil Engineering and Technology (IJCIET)*, 9(11), 1269-1278.
[2] Bangari, V., Marquez, B.A., Miller, H., Tait, A.N., Nahmias, M.A., de Lima, T.F., Peng, H.-T., Prucnal, P.R., & Shastri, B.J. (2020). Digital electronics and analog photonics for convolutional neural networks (DEAP-CNNs). *IEEE Journal of Selected Topics in Quantum Electronics*, 26(1), article number 7701213. doi: 10.1109/JSTQE.2019.2945540.
[3] Bogaerts, W., Pérez, D., Capmany, J., Miller, D.A.B., Poon, J., Englund, D., Morichetti, F., & Melloni, A. (2020). Programmable photonic circuits. *Nature*, 586, 207-216. doi: 10.1038/s41586-020-2764-0.
[4] Borovytsky, V., & Avdieionok, I. (2024). Economical optical matrix to vector multiplier. *Proceedings SPIE*, 12938, article number 129381F. doi: 10.1117/12.3013064.
[5] Borovytsky, V., Avdieionok, I., Tuzhanskyi, S., & Lysenko, H. (2022). Photonic integrated circuits for optical matrix-vector multiplication. *Optoelectronic Information-Power Technologies*, 43(1), 11-18. doi: 10.31649/1681-7893-2022-43-1-11-18.
[6] Cheng, J., *et al.* (2022). A small microring array that performs large complex-valued matrix-vector multiplication. *Frontiers of Optoelectronics*, 15, article number 15. doi: 10.1007/s12200-022-00009-4.
[7] Cheng, J., Zhou, H., & Dong, J. (2021). Photonic matrix computing: From fundamentals to applications. *Nanomaterials*, 11(7), article number 1683. doi: 10.3390/nano11071683.
[8] Dey, S. (2020). *Python image processing cookbook: Over 60 recipes to help you perform complex image processing and computer vision tasks with ease*. Birmingham: Packt Publishing.
[9] Huang, C., *et al.* (2020). Demonstration of scalable microring weight bank control for large-scale photonic integrated circuits. *APL Photonics*, 5(4), article number 040803. doi: 10.1063/1.5144121.
[10] Luo, W., *et al.* (2023). Recent progress in quantum photonic chips for quantum communication and internet. *Light: Science & Applications*, 12, article number 175. doi: 10.1038/s41377-023-01173-8.
[11] Ohno, S., Tang, R., Toprasertpong, K., Takagi, S., & Takenaka, M. (2022). Si microring resonator crossbar array for on-chip inference and training of the optical neural network. *ACS Photonics*, 9(8), 2614-2622. doi: 10.1021/acsphotonics.1c01777.
[12] Strąkowska, M., Urbaś, S., Felczak, M., Torzyk, B., Shatarah, I.S.M., Kasikowski, R., Tabaka, P., & Więcek, B. (2024). Modelling and thermographic measurements of LED optical power. *Sensors*, 24(5), article number 1471. doi: 10.3390/s24051471.
[13] Wojcik, B., & Żarski, M. (2021). The measurements of surface defect area with an RGB-D camera for a BIM-backed bridge inspection. *Bulletin of the Polish Academy of Sciences. Technical Sciences*, 69(3), article number e137123. doi: 10.24425/bpasts.2021.137123.
[14] Wu, W., Zhou, T., & Fang, L. (2024). Parallel photonic chip for nanosecond end-to-end image processing, transmission, and reconstruction. *Optica*, 11(6), 831-837. doi: 10.1364/OPTICA.516241.
[15] Zhou, H., Zhao, Y., Wang, X., Gao, D., Dong, J., & Zhang, X. (2020). Self-configuring and reconfigurable silicon photonic signal processor. *ACS Photonics*, 7(3), 792-799. doi: 10.1021/acsphotonics.9b01673.
[16] Zhu, Y., *et al.* (2024). Silicon photonic neuromorphic accelerator using integrated coherent transmit-receive optical sub-assemblies. *Optica*, 11(4), 583-594. doi: 10.1364/OPTICA.514341.

Характеристика оптичних хвилеводів для фотонних інтегральних схем

Ірина Авдєйонок
Аспірант
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
03056, просп. Берестейський, 37, м. Київ, Україна
https://orcid.org/0009-0007-4709-0030

Володимир Боровицький
Доктор технічних наук, професор
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
03056, просп. Берестейський, 37, м. Київ, Україна
https://orcid.org/0000-0001-6816-0391

Анотація. Швидка обробка сигналів зі швидкістю світла є основною перевагою фотонних інтегральних схем. Тому ці схеми мають гарні перспективи для реалізації математичних обчислень, в тому числі перемноження матриці на вектор. Метою роботи було створення та дослідження методики автоматичного вимірювання розподілу яскравості вздовж оптичних хвилеводів аналогових фотонних інтегральних схем. Під час дослідження використано емпіричні методи (спостереження, вимірювання, порівняння, експеримент) та комплексний метод (аналіз і синтез). Запропонована методика використовує цифрову камеру, яка фіксує зображення оптичного хвилеводу, освітленого світлодіодами, та програмне забезпечення для обробки зображень для розрахунку розподілу яскравості. Ця методика визначає найкращу апроксимацію цього розподілу, обчислює параметри нерівномірності яскравості та втрати оптичного випромінювання. Вимірювання набору оптичних хвилеводів допомагають визначити найкращих кандидатів для фотонних інтегральних схем. Виявлено, що оптичні хвилеводи з шліфованими поверхнями, які виконують функцію дифузного розсіювання, мають гарне поєднання плавного розподілу яскравості та малих втрат оптичного випромінювання. Завдяки багаторазовому дифузному відбиванню і розсіюванню в матеріалі хвилеводу, ці хвилеводи є перспективними кандидатами для аналогових фотонних інтегральних схем. Всі інші хвилеводи з необробленою поверхнею, з канавками або шліфовані з великим зерном мають достатні втрати оптичного випромінювання. Ці втрати, як правило, спричинені виходом оптичного випромінювання з поверхні хвилеводу. Отримані результати необхідні для точного проектування схем, що враховують розсіювання і втрати в оптичних хвилеводах. Запропонована методика може бути застосована в автоматизованому технологічному процесі виготовлення швидкої та економічної фотонної матриці для векторного множення, яка не потребує дорогого електронно-променевого, оптичного або лазерного літографічного обладнання

Ключові слова: розподіл яскравості; автоматичне вимірювання; обробка зображень; оцінка однорідності; оцінка втрат; множення фотонної матриці на вектор; оптоелектроніка

BULLETIN of Cherkasy State Technological University

Journal homepage: https://bulletin-chstu.com.ua/en

Vol. 29 No. 2. 2024

UDC 004.94: 621.3.049.7:004.4

DOI: 10.62660/bcstu/2.2024.32

A study of the electric circuit modelling and simulation software efficiency and their accuracy, speed and ease of use comparison

Larysa Vasetska*

PhD in Technical Sciences, Associate Professor

Luhansk Taras Shevchenko National University

36000, 3 Ivan Bank Str., Poltava, Ukraine

https://orcid.org/0000-0002-4873-8835

Abstract. The increasing complexity of microchips and the limitations of miniaturisation are making the electronics design process more complex and time-consuming. The problem of Moore's Law losing its force is causing rapid development in the design and manufacture of electronic components. The study aimed to provide structured information on electrical circuit design tools by functionality and availability to help optimise operations, increase productivity and save money. The study provided a theoretical description of the principles of operation of software for modelling and simulation of electrical circuits, an overview of algorithms and methods used in automatic design programs, a classification of programs by key characteristics and a comparative analysis of popular software packages: LTspice, EAGLE, MATLAB/Simulink, Multisim, Proteus, KiCad. The analysis addressed the speed of modelling, accuracy of results, ease of use, functionality and accessibility, as well as user experience. The study recommended selecting software depending on the user's needs, project complexity and budget, accounting for the experience of other users. LTspice, Multisim, and KiCad were recommended for beginners and students: LTspice is powerful, free and easy to use, KiCad offers open source and an active community, and Multisim is suitable for learning. MATLAB/Simulink, Proteus and EAGLE are recommended for professionals: Simulink provides powerful modelling and integration with other MathWorks products; Proteus is optimal for microcontroller system developers; EAGLE is suitable for integration with CAD and automation systems. The practical value of the research results lies in the creation of a rating of EDA software by the criteria of functionality and performance, designed to help users make a choice based on their needs

Keywords: electronics design tools; circuit analysis; component library; supported standards; interface

INTRODUCTION

Electrical circuit simulation (EDA) software is a versatile tool for design, training, analysis and research in electronics that can be used to test models quickly and flexibly without the risk of damaging components, saving time and money and reducing the risk of errors. The graphical and textual interface software visualises electrical quantities that can be used to compare designs and investigate the effects of components, making learning dynamic and interactive. Although EDA software has undeniable advantages, it has limitations due to its cost and the possibility of inaccurate modelling. The development of electrical circuit simulation software began with early computers and is constantly evolving. The first programs for numerical solutions of differential equations appeared in the 1950s. The Simulation Program with Integrated Circuit Emphasis (SPICE) software, developed in 1973, was the first publicly available tool for analysing electrical circuits and quickly gained popularity due to its power and accuracy.

Article's History: Received: 09.01.2024; Revised: 11.04.2024; Accepted: 27.05.2024

Suggested Citation:

Vasetska, L. (2024). A study of the electric circuit modelling and simulation software efficiency and their accuracy, speed and ease of use comparison. *Bulletin of Cherkasy State Technological University*, 29(2), 32-44. doi: 10.62660/bcstu/2.2024.32.

*Corresponding author

As noted in the review of the evolution, current state and trends of EDA software by N. Charig (2021), in the 1980s and 1990s, new versions of SPICE appeared with improved component models and analysis methods. Commercial products based on SPICE offered additional features and technical support. Graphical interfaces have simplified the process of creating and modelling schemes. In the 2000s and 2010s, modelling tools were integrated with Computer-Aided Design (CAD) systems. New features such as high-frequency (HF) circuit analysis, thermal analysis, and electromagnetic compatibility (EMC) analysis have been added. This development allowed engineers to perform complex analysis and design at one workstation.

According to K.I. Gubbi *et al.* (2022), modern modelling tools often employ cloud computing to increase capacity and availability, as well as artificial intelligence to automate and optimise design processes. Software that allows simultaneous modelling of various physical phenomena, such as electrical, thermal, mechanical and electromagnetic processes, is actively emerging. Tools such as Qucs and NGSPICE continue to evolve as open platforms that provide accessibility and flexibility for users. Modern research on EDA tools, by G. Huang *et al.* (2021), covers a wide range of methods and algorithms for modelling, analysis, specification, computer-aided design, testing, and verification of complex electronic systems, making them an indispensable tool for modern electronic design.

A report by G. Marinova & A. Bitri (2021), which assesses the dynamics of EDA software development, indicates that electronic systems design automation is one of the most complex and dynamic industries, requiring companies to invest heavily in research and development, as well as continuous product improvement and adaptation to changing market conditions. As noted by R.I. Bahar *et al.* (2020), changes in electronic design are driving new research: future advances in electronics rely on extreme-scale design automation to achieve breakthroughs in performance, power, and integration for next-generation systems. At the same time, C.-D. Curiac & A. Doboli (2022) argue that their analysis of the EDA industry has revealed a significant gap between basic research and applied development, which leads to a decrease in the industrial influence on the development of electronics. To address this problem, the researchers suggested stimulating cooperation between academic institutions and industry, focusing on current industry issues, and simplifying scientific formulations to make research results more accessible to a wide range of professionals.

In Ukraine, LTSpice, Proteus, Arduino, MATLAB/Simulink, and Multisim tools are used in engineering education and research. D. Arseniuk & Yu. Zinkovsky (2023) used LTSpice to model a bridgeless power factor corrector. V. Shamonia *et al.* (2019) employed the Proteus virtual environment to teach future IT professionals how to design printed circuit boards and develop microcontrollers. V. Andriienko & M. Bondarenko (2018) analysed robotic platforms for prototyping electronic devices and systems. O. Semenikhina *et al.* (2020) compared the most popular Proteus and Multisim programmes among university teachers. A pedagogical experiment has shown that using Multisim by IT students allows them to perform laboratory work more efficiently due to shorter task completion times. At the same time, both programmes promote the development of skills in modelling and analysing electronic circuits, which is confirmed by the reduction in student learning time.

The problem of choosing EDA software is relevant and complex, as it requires a thorough analysis of user needs, consideration of software characteristics, and constant updating of information about available solutions. Thus, the study aimed to compare the efficiency of software for modelling and simulating electrical circuits. The study is designed to help users make a correct choice, helping to optimise operations, increase productivity and save time and money.

MATERIALS AND METHODS

An overview of the mathematical foundations of EDA software was provided to enable a comparative analysis of EDA programs in terms of accuracy, speed, and usability. The flowchart of the electronic circuit modelling process explains the general algorithm for applying data processing methods in EDA applications. To systematise knowledge about design tools and form a holistic view of the process of developing electronic systems, a classification of software by the characteristics under study was created: by the area and industry of application, commercial availability, operating system, functionality, as well as by the method, type and complexity of modelling, which is the basis for a comparative analysis of different tools.

A comparative review of popular software for designing electrical circuits by key characteristics and functionalities was conducted. A detailed analysis of popular software packages was conducted with a focus on mathematical models and modelling capabilities. This provided a list and description of common data processing methods used in popular tools for circuit modelling. Methods for optimising the operation of EDA programs to increase computing speed and reduce memory consumption were analysed.

The quantitative analysis compared the characteristics of the programmes: modelling speed, accuracy of results, ease of use, functionality, and cost. EDA software developer websites were used for the quantitative analysis: LTspice, Simulink, Multisim Live, Proteus, KiCad, Easily Applicable Graphical Layout Editor (EAGLE) and literature review. Information from the manual on designing and analysing circuits using PSpice and MATLAB tools by W.Y. Yang *et al.* (2020), which focuses on how these tools can model electrical circuits, even though their functionality is significantly different was used: MATLAB and PSpice complement each other, allowing an engineer to effectively solve a wide range of tasks, from concept development to debugging a finished device. A book edited by S.L. Tripathi *et al.* (2022), which includes a comprehensive and up-to-date introduction to the basic concepts of digital design and digital design verification using Verilog/Hardware Description Language (HDL), was also used. Mathematical models, data processing methods and possibilities for optimising the operation of programs when modelling digital circuits in LTspice, Simulink, and Multisim tools were analysed separately. Simplified modelling capabilities of KiCad, EAGLE and Proteus tools with a focus on quick verification of circuits and evaluation of their performance, as well as on one of the significant advantages of these programs – openness for development, were highlighted. With support for plug-ins and scripts, these tools are more adaptable and allow users to create personal workflows and integrate applications with other tools.

The information obtained was systematised and presented in the form of summary tables, covering the components of the software functionality: set of models, calculation accuracy, simulation speed, commercial availability, complexity of use, and availability of additional features. The results are intended to provide users with structured information about the capabilities and limitations of EDA software, help optimise their work and increase productivity.

RESULTS

EDA software systems are based on mathematical models that describe the behaviour of various electronic components and systems: linear differential equations, nonlinear differential equations, algebraic equations, and partial differential equations. The use of systems of equations to describe circuits in EDA is a fundamental approach. The choice between linear and nonlinear systems of equations in EDA depends on the complexity of the circuit model, the specifics of the task analysis and the required accuracy of the results.

Linear differential equations are widely used to describe the behaviour of passive elements (resistors, capacitors, inductors), as well as active elements in linear operation (transistors, operational amplifiers). Nonlinear differential equations are used to model nonlinear elements (diodes, transistors in the nonlinear mode), as well as to describe complex physical phenomena such as saturation of a magnetic circuit, and electrical breakdown.

Algebraic equations are used to describe the static characteristics of elements (e.g., the volt-ampere characteristics of diodes). Partial differential equations are used to model distributed parameters of systems, such as transmission lines and chip structures. The modelling process in EDA software is shown in Figure 1.

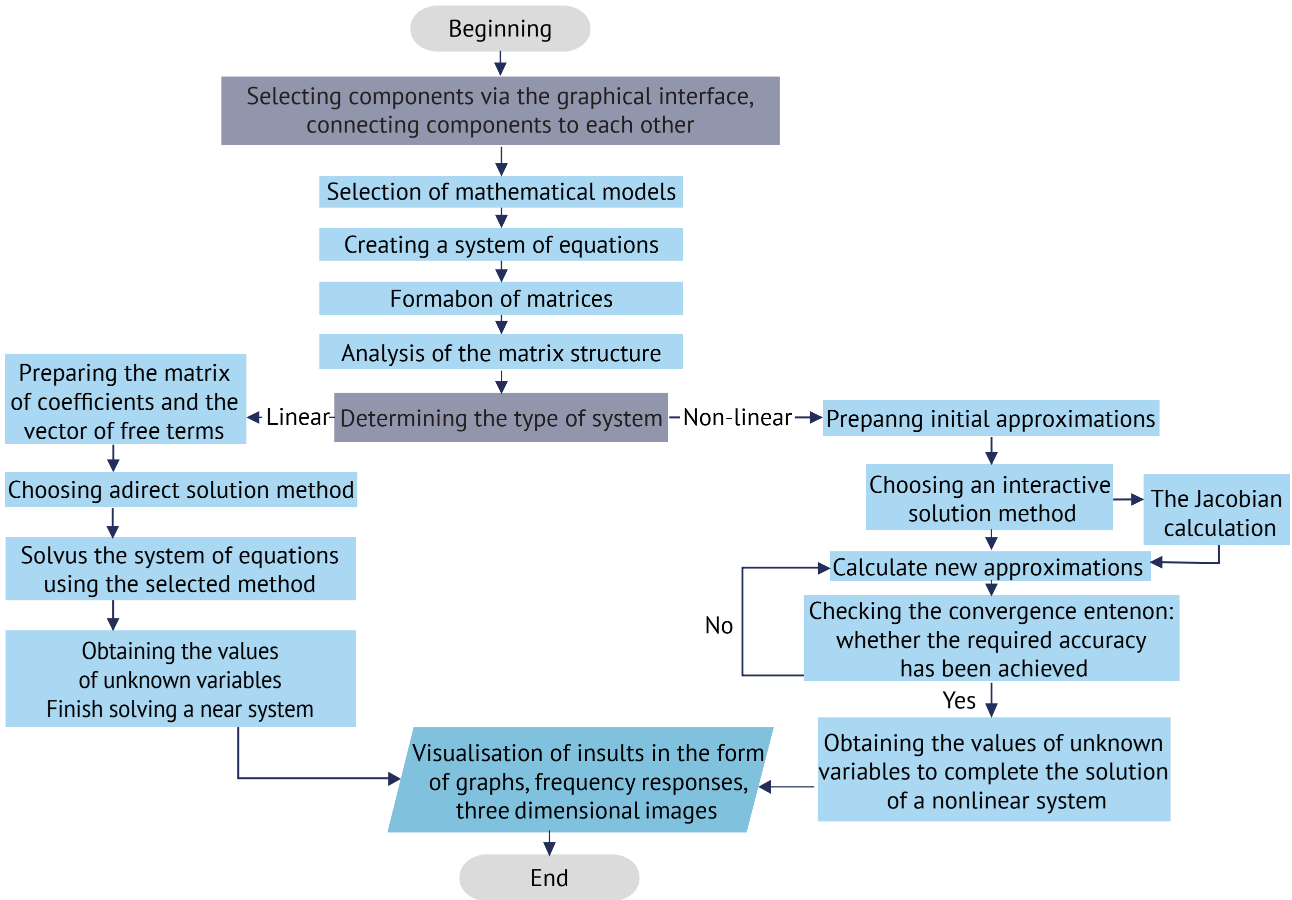


Figure 1. Flowchart of the electronic circuit modelling process

Source: compiled by the author

A distinctive feature of nonlinear differential equations is the absence of an analytical solution in the general case. Therefore, numerical methods are used to solve them. Electronic circuit modelling software uses numerical methods adapted to the specifics of electrical engineering tasks: numerical integration – for solving differential equations (the choice of method depends on the required accuracy and speed of calculations); matrix methods – for solving systems of linear algebraic equations; optimisation methods – for finding optimal model parameters. The methods applied are shown in Table 1.

Table 1. Data processing methods in EDA applications

Method category	Method	Description
Numerical integration	Euler's method	A simpler numerical method for solving ordinary differential equations that uses first derivatives.
	Runge-Kutta method	A more accurate numerical method that uses intermediate estimates of derivatives.
	The trapezoid method	A numerical method that uses the average of the derivatives at each step to integrate.
Matrix methods	Gaussian method	A method for solving systems of linear algebraic equations by successive elimination of variables.
	LU decomposition method	Splitting a matrix into a product of lower and upper triangular matrices to simplify solving systems of equations.
Optimisation methods	Least squares method	A method for minimising the difference between observed and theoretical values.
	Gradient descent method	An iterative method for finding local minima of functions by moving in the direction of the anti-gradient.
Discretisation methods	Finite difference method (FDM)	Converts differential equations into difference equations for numerical solution.
	Finite element method (FEM)	Break down complex geometries into small elements for numerical analysis.
Monte Carlo methods	Monte Carlo method	A statistical method for modelling random processes such as noise or temperature fluctuations.
Event-based modelling	Event-based modelling	Dynamic analysis of digital circuits, where the model is updated in response to events (changes in the state of signals).
Methods for analysing frequency response	Fourier method	Converting signals to the frequency domain for spectral component analysis.
	Laplace transform	Analysis of the behaviour of systems in the frequency domain and finding transfer functions.
Symbolic methods	Symbolic computing	Analytical solution of equations or simplification of mathematical expressions describing electronic circuits.
Methods of stability and sensitivity analysis	Resilience and sensitivity analysis	Evaluation of system stability and its sensitivity to changes in parameters to ensure the reliability and stability of the schemes.
Parallel computing	Parallel computing	Using multiprocessor systems or clusters to speed up computing processes.
Data-driven modelling	Data-driven modelling	Use of machine learning techniques to create models based on empirical data, such as neural networks and regression methods.

Source: compiled by the author

Optimising the performance of EDA is a key factor for effective electronic design. Modern tools use a variety of methods to speed up calculations and reduce memory consumption:

1. Automatic method selection. Most modern EDA tools automatically select the optimal solution method depending on the characteristics of the problem.

2. Hybrid methods combine direct and iterative methods to achieve high efficiency.

3. Simplification of models. To speed up calculations, simplified component models are often used, such as using ideal power supplies instead of real ones or reducing complex circuits to equivalent ones. The choice of model depends on the required accuracy of the results and the range of operating frequencies.

4. Parallel computing for complex models on multi-core processors or GPUs. Typical parallel programming technologies include the Message Passing Interface and the OpenMP standard.

5. Adaptive integration methods. To reduce computational costs, adaptive integration methods, such as the Runge-Kutta method, are used, which automatically select the integration step depending on the rate of change of the variables.

6. Compilation of models. To increase the speed of execution, models can be compiled into machine code, which avoids interpretation overheads.

7. Algorithm optimisation increases the speed of computation through more efficient data structures, reduced operations, and parallelisation.

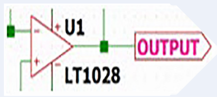
8. The results are verified by comparing them with other simulators, analytical solutions and experimental data. Formal verification confirms the correctness of the scheme mathematically.

To create accurate models, EDA programs use HDL. In particular, the analysed tools use SPICE, Analysis of Basic Circuits and Devices (AB-CD), Verilog and Very-High-Speed Hardware Description Language (VHDL), each of which has its characteristics and applications. SPICE and AB-CD have traditionally been used for modelling analogue circuits, while Verilog and VHDL have become the standard for describing digital circuits. EDA tools can be used to effectively combine these languages to model mixed analogue and digital systems (Table 2). Figure 2 provides a classification of EDA software based on their functionality and modelling methods. For further analysis, packages widely used in the industry from such companies as LTspice with the software of the same name, MathWorks (MATLAB/Simulink tool), National Instruments (Multisim tool), Labcenter Electronics (Proteus tool), KiCad Developers (KiCad tool), Autodesk/CadSoft Computer (EAGLE tool) were used.

Table 2. Comparing HDL for modelling analogue and digital circuits

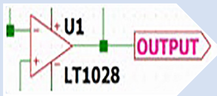
Method	Description	Usage	Connection with other methods
SPICE	A powerful tool for simulating the behaviour of analogue circuits.	Analyse voltage, current, power consumption, frequency response and other aspects of analogue circuits.	It can be used to model analogue components used in digital circuits described by Verilog or VHDL.
AB-CD	A SPICE extension that focuses on modelling basic electronic components and circuits.	Modelling simple resistors, capacitors, transistors and other basic components.	It can be used as an introduction to SPICE or for quick simulation of simple analogue circuits.
Verilog (Hardware Description Language)	A textual language for describing the behaviour of digital circuits based on the behaviour of components.	Modelling of logic gates, triggers, memory and other digital components.	It is not intended for modelling analogue components but can be used in conjunction with SPICE to model mixed analogue and digital systems.
VHDL		Similar capabilities as Verilog, with some differences in syntax and approach.	It is not intended for analogue modelling but can be used in conjunction with SPICE to simulate mixed analogue and digital systems.

Source: compiled by the author based on W.Y. Yang *et al*. (2020) and S.L. Tripathi *et al*. (2022)



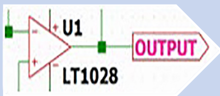
By type of modelling

- Analogue
- Modelling analogue electrical circuits
- Digital
- Modelling digital circuits
- Mixed signal
- Modelling analogue and digital electrical circuits



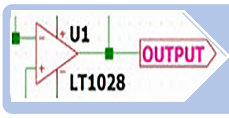
By complexity

- Simple applications for beginners
- Intermediate-level applications for advanced users
- Professional applications for complex analysis



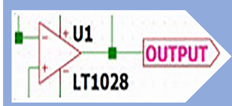
By modelling method

- SPICE
- AB-CD
- Verilog (Hardware Description Language)
- VHDL



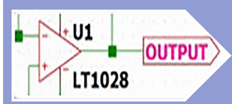
By commercial availability

- Free software for modelling
- Paid modelling software with more features



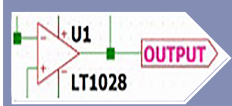
By field of application

- General purpose
- Modelling different schemes
- Specialised
- Modelling a specific type of circuit (e.g. power supply, RF)



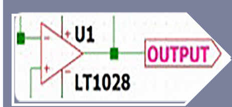
By operating system

- Windows
- macOS
- Linux
- Cloud-based applications accessible via a web browser



Areas of use

- Electronics
- Designing microcircuits
- RF and microwave technology
- Electricity
- Medical equipment



By functionality

- Basic modelling - applications with a limited set of functions
- Applications with an extended range of functions
- Software for analysing the reliability of electrical circuits
- Software for optimising electrical circuit designs
- Programs for generating code from circuit models

Figure 2. Classification of EDA software

Source: compiled by the author

LTspice, Simulink, and Multisim are powerful tools for modelling electronic circuits, including digital circuits. Each has its set of mathematical models, component libraries, and tools for analysis and optimisation. The differences in the use of methods between LTspice, Simulink and Multisim are due to their architecture and are the result of evolution: each programme has evolved to focus on certain types of tasks and adapt to the needs of different audiences. LTspice focuses on analysing analogue and mixed-signal circuits: and often uses numerical integration methods (Euler, Runge-Kutta, trapezoids), matrix methods (Gauss, LU decomposition), discretisation methods (FDM), and frequency response analysis methods (Fourier, Laplace). Simulink is a more versatile tool for modelling systems of various kinds. It supports a wide range of methods, including numerical integration methods, discretisation methods (FDM, FEM), optimisation methods, Monte Carlo methods, event-based modelling, and frequency response analysis methods, and offers advanced tools for optimising model parameters and performing Monte Carlo analysis due to its focus on system modelling. Depending on the configuration, the FEM discretisation method may be more common in Simulink due to its ability to model complex geometries. Multisim is similar in purpose to LTspice but may have some differences in the set of available methods and their implementation. Depending on the configuration, Simulink and Multisim can have more advanced event-based modelling capabilities for discrete and event-driven systems through the availability of special blocks and tools, as well as for data-driven modelling through integration with machine learning tools.

Although all three programs support some elements of symbolic computing, Simulink may have more advanced capabilities in this area due to integration with other MATLAB tools, depending on the configuration. Modern versions of all three programs support parallel computing to speed up simulations. In the following, the modelling of digital circuits in LTspice, Simulink, and Multisim will be discussed:

1. Mathematical models in these software packages for modelling digital circuits include boolean algebra to describe logical operations (AND, OR, NOT, etc.); switching functions that model the behaviour of digital elements (logic gates) depending on input signals; finite state machines to describe sequential logic circuits (registers, counters and controllers); delay models to account for signal propagation delays in real circuits; noise models to simulate the effect of noise on the operation of digital circuits.

2. Data processing during computer modelling of digital circuits in LTspice, Simulink and Multisim is carried out using the following methods: discrete simulation (e.g., Euler, Runge-Kutta), which allows to model the change of signals in time with a given step; Boolean algebraic logic equations that describe the functioning of digital circuits at the level of logical operations; truth tables used to determine the output values of logical elements depending on the input values; and methods of simplifying logical expressions to minimise the number of logical elements in a circuit.

3. The following methods are used to optimise digital circuits in these applications: automatic circuit synthesis, i.e. creating a circuit based on a given function or truth table, speed optimisation to minimise signal propagation delays, chip area optimisation to reduce the number of logic elements used, and power optimisation to reduce circuit power consumption.

Thus, LTspice and Multisim, which specialise in circuit analysis, use discrete simulation and logic equations to model digital components, but their optimisation capabilities are limited. Simulink, as a universal system modelling platform, offers a wider range of tools for working with digital circuits, including block diagrams and powerful optimisation tools.

The main function of KiCad, EAGLE and Proteus software systems is to perform circuit design and PCB development, and they can also include modelling elements. However, unlike specialised simulators (LTspice, Simulink), these programs usually provide more simplified modelling capabilities focused on quick circuit verification and evaluation of their performance. KiCad focuses on circuit design and PCB development. Simulations in KiCad are limited to simple DC and AC analyses. For more complex simulations, it is recommended to use external simulators such as Ngspice. EAGLE is also focused on circuit design. Simulations in EAGLE can be performed using the built-in simulator, but its capabilities are limited. Proteus provides more extensive simulation capabilities, including microcontrollers, analogue electronics, and digital circuits. This tool uses a proprietary simulator based on a SPICE-like kernel. All three programs use simplified models for basic electronic components (resistors, capacitors, transistors, etc.). These models are typically based on linear differential equations and may not account for complex nonlinear effects. Proteus provides detailed models of popular microcontrollers that can be used to simulate the operation of software. These models are based on CPU emulation and can be quite resource intensive.

Data processing in computer modelling of digital circuits in KiCad, EAGLE and Proteus software is carried out using the following methods: numerical integration for solving differential equations (Euler's method, Runge-Kutta method); matrix methods for solving systems of linear algebraic equations; Boolean algebraic logic equations to describe the functioning of digital circuits at the level of logical operations and truth tables to determine the output values of logical elements depending on the input values. The following methods are used to optimise digital circuits in these applications: simplifying models to speed up simulations; caching the results of previous simulations to speed up subsequent calculations; and using efficient algorithms for computing. One of the significant advantages of

KiCad, EAGLE, and Proteus is their openness to development: With support for plug-ins and scripts, these tools are more adaptable and allow users to create their workflows and integrate applications with other tools.

Available electronic design software such as KiCad, EAGLE and Proteus may have limited capabilities for modelling complex systems and lower accuracy compared to specialised simulators such as SPICE (e.g. LTspice, PSpice) or system modelling environments such as MATLAB/Simulink, with simulation speeds in all cases depending on circuit complexity and computer power. For the accurate modelling of complex electronic circuits, it is recommended to use more specialised software, such as LTspice, and Simulink. The summary results of the comparative analysis are presented in Table 3.

Table 3. Summary results of the comparative analysis of EDA software

Program	A set of models	Accuracy of calculations	Simulation speed	The cost of the licence	Difficulty of use	Additional features
LTspice	Extensive library of standard elements, support for non-linear elements, and the ability to create custom SPICE models.	High accuracy (depending on the component model and simulation settings), recognised in the industry for using the SPICE simulator to solve nonlinear differential equations.	High-speed thanks to an optimised simulator core. Depends on the complexity of the scheme and simulation settings.	Free	Intermediate, requires skills at SPICE basics but has a lot of documentation and examples.	Analysis of DC, AC, transients, noise, spectral analysis.
Simulink	A wide library of blocks for modelling physical and control systems. Supports co-simulation with other applications.	High accuracy due to the use of different numerical methods depends on the choice of integrator and integration step.	High, but may vary depending on the complexity of the model and available computing resources.	Paid (there are academic licences for students and teachers)	High, requires knowledge of mathematical modelling and the MATLAB environment. takes time to master, especially for complex models.	Control systems, digital signal processing, embedded systems development.
Multisim	A large library of components from different manufacturers, support for microcontrollers.	The high accuracy of the SPICE algorithm depends on the choice of component model and simulation settings.	High, optimised for fast electronic circuits.	Paid (trial version available)	Medium, intuitive interface, a large number of training materials.	DC, AC, transient and noise analysis, spectral analysis, and microcontroller simulation.
KiCad	Limited compared to other programs, but it is possible to import models. Library of standard elements, the ability to create custom models.	Satisfactory accuracy for most projects, but inferior to commercial applications. May be insufficient for RF or analogue circuits.	Average, depending on the complexity of the circuit and the simulator settings.	Free, open source	Medium requires knowledge of electronics and simulator setup, takes time to learn, but has a lot of documentation and a support community.	Design of printed circuit boards, and analysis of digital circuits.
EAGLE	Library of standard elements, support for external libraries.	High precision for electronic circuits, widely used in industry.	High, optimised for printed circuit boards.	Paid (there is a free version for small projects)	Medium, the interface may seem complicated for beginners.	Design of printed circuit boards, and analysis of digital circuits.
Proteus	A wide library of electronic components and analogue systems, integrated models for microcontrollers.	High accuracy, especially for microcontrollers in integrated systems.	High, optimised for real-time simulation.	Paid (academic licences are available)	Medium, intuitive interface with lots of examples and documentation.	Microcontroller simulation, embedded systems development, analogue modelling.

Source: compiled by the author

According to the review, LTspice is recognised as a popular choice among engineers and students for designing and modelling electronic circuits due to its affordability, ease of use fast, high-fidelity modelling, extensive component library, and extensive support with numerous online tutorials, forums, and video tutorials. LTspice's user interface is noted for its combination of intuitive text and graphical elements, making it convenient for users with different levels of experience. The software offers fast modelling, high accuracy, and the ability to automate Python tasks with external tools, making it useful for researchers and engineers who need fast and accurate results. At the same time, the software's graphics power is rated as average, which can be a limitation for users who need to visualise complex data or models.

EAGLE is another powerful EDA tool, particularly known for its ease of use and integration with Autodesk Fusion 360. EAGLE is recognised as a complete solution for professional electronics designers, offering detailed documentation and an active community to support users at all stages of their work through a wide range of features, regular updates and access to experts. It supports scripting and PCB design, making it a versatile tool for both hobbyists and professionals. The advantages are cross-platform, an extensive library, powerful graphics and a user-friendly graphical interface. MATLAB/Simulink is recognised by users as a powerful commercial product that provides a wide range of functions, including advanced mathematical modelling and system-level modelling, in addition to circuit simulation. It is preferred in professional and academic environments due to its full functionality and cross-platform compatibility. The Multisim software package has proved to be the most popular among users in the sense that engineers first acquire experience with it during their studies. The choice of educational institutions is based on the convenience of the graphical interface, a large library of components with the ability to add custom components, powerful graphics, and signal visualisation. Users praised the free version for its high functionality for educational needs, active forum, and the completeness and relevance of the user manual. Proteus is a commercial software recognised as a powerful tool for the full cycle of electronic device development. This software package is in demand among engineers and electronics developers since it combines tools for circuit design, PCB tracing and microcontroller simulation in a single environment. The advantages of the KiCad tool include open source, minimal system requirements and cross-platform compatibility. The new HyperText Transfer Protocol libraries feature of the latest KiCad version was noted, which creates opportunities for expanding the component library (the basic version has a limited library). The limitations of KiCad include the lack of built-in tools for signal integrity analysis, such as noise and EMC analysis, which may be important for some projects. The support, completeness and relevance of the documentation received no complaints and were considered sufficient.

DISCUSSION

According to the study, commercial tools MatLab/Simulink and Proteus offer a wide range of functionality, a user-friendly interface, and expert technical support from the developers. They can be beneficial for large companies and professional users who need sophisticated features and reliability. On the other hand, open-source software such as LTspice, KiCad, and Eagle are flexible and transparent, allowing them to be modified by users and adapted to specific needs, making them attractive to enthusiasts, researchers, and small teams. This idea is confirmed by the work of S. Grau *et al.* (2024), who used the open-source code of the KiCad tool to create the KiTorquer plug-in to extend the software's capabilities for designing and optimising magnetic activators on printed circuit boards.

Open-source software typically relies on a community of users for support, which can be less reliable. The communities of commercial and open-source software differ: the former tends to have a smaller community, while the latter is characterised by an active community where users share knowledge, collaborate on projects and help solve problems. The difference in the capabilities of proprietary and open EDA solutions was investigated by A.B. Kahng *et al.* (2022). The author emphasised their complementarity, stressing the importance of combining the strengths of both types of programmes to achieve a synergistic effect and solve complex problems in the field of chip design. Thus, this area of EDA development is recognised as promising and requires further in-depth research. In particular, the development of integrated environments that allow for the effective combination of the capabilities of different tools. An example of such innovations is the OpenROAD platform for automated synthesis of digital integrated circuits, described in J. Chen *et al.* (2020). This tool implements the full flow from the description at the register level to the generation of geometric data for production.

Based on the results obtained, LTspice, Multisim and KiCad can be recommended for beginners, students and enthusiasts, not least because of their affordability. The use of powerful commercial software provides a wider range of functions, including analysis, simulation and optimisation, while free software is usually less functional, but with the possibility of extensions through plug-ins and add-ons. P. Dang & H. Arolkar (2019) analysed software packages, including LTspice, noting its use in areas such as electronics, telecommunications, and medical equipment; he also explored its limitations. The author noted that LTspice does not directly support Verilog and VHDL, as it uses its format for describing components and connections and is not designed to simulate complex digital systems.

However, it is possible to create component models in Verilog-A (a subset of Verilog that is compatible with SPICE) and import them into LTspice. It is also possible to use external Verilog/VHDL simulators to simulate the digital part of the design and then import the results into LTspice to simulate the analogue part.

The results obtained in the comparative analysis showed the high accuracy of the LTspice modelling tool. These data are confirmed by O. Osadchuk *et al*. (2021). Using this tool, the research team developed a mathematical model of a microelectronic frequency humidity converter, which demonstrated high accuracy (error does not exceed 1.8%) when compared with the results of computer modelling and experimental studies. MatLab/Simulink provides a wide range of functions beyond circuit simulation, including advanced mathematical modelling and system-level modelling. It is preferred in professional and academic environments due to its full functionality. O. Tolochko (2020) considered the modelling of electrical and electromagnetic circuits, and electronic devices using Simulink by using virtual blocks of SimPowerSystems libraries. It is noted that users do not need to know the mathematical details of the elements, and the capabilities of the program provide analysis of models in the time, frequency and Laplace space for a complete understanding of the system.

Multisim offers advanced modelling capabilities and a user-friendly graphical interface. It is widely used in education and industry due to its detailed analysis tools, ease of use, and the availability of a free version. Ye. Dosymov *et al*. (2023) noted that the virtual circuit model verification function deepens students' physical perception of electromagnetic processes and provides a better understanding of the relationship between the mathematical model and its real-world analogue. A similar conclusion was reached by L. Li *et al*. (2021) and V. Nerubatskyi *et al*. (2023), who noted the speed, efficiency, intuitive presentation of the characteristics of circuit elements and signal simulation in Multisim when developing even, odd, and fractional frequency dividers for different modes of division.

Proteus has been praised for its ability to simulate microcontroller operation and integrate embedded software, making it useful for embedded system developers. The results highlighted its user-friendly interface and widespread use in both educational and professional contexts. M.A.A. Mareai (2024) focused on its application to electronic circuit modelling, noting how Proteus Simulation uses real-time technology to render the simulation. This allows users to see how the circuit performs in dynamics, which can be useful for debugging and optimising the design. The capabilities of the Proteus tool in teaching and developing skills in designing and modelling electronic circuits are demonstrated in the work of Z. Hashaam (2024), in which a RAM interface was modelled and tested using the Proteus Design Suite to understand the process of data transfer in embedded systems.

A valuable feature of the EAGLE tool in microcontroller PCB manufacturing is the synchronisation of changes in the circuit and on the board, as well as the ability to roll back changes. This makes it easier to work and saves time, provides flexibility and ensures project reliability. These features make EAGLE a versatile tool for both hobbyists and professionals. Similarly to other commercial tools, it offers a complete set of features tailored to the needs of electronics designers, as evidenced by the result of the development of a multilayer PCB by M.N. Islam *et al*. (2022). To design the board using the EAGLE layout editor module, he used an auto-router to generate a trace on the board, and the CAM Job tool to generate production data. After manufacturing, the board was successfully tested for functionality.

The introduction of artificial intelligence and big data analytics into EDA systems is leading to significant progress in the development of electronic devices. Automatic routing and other intelligent tools simplify and speed up the design process, making it more efficient. This is evidenced by the results of the work of M. Venkateswara *et al*. (2020). The paper describes the design and testing of an algorithm for automatically recognising hand-drawn electrical schematics and converting them into LTspice format using machine learning techniques. The implementation of this method is intended to automate the process of checking the circuit and save time and effort for engineers. As noted by M. Cirstea *et al*. (2024), modern EDA tools, by integrating different levels of abstraction, allow engineers to efficiently design complex electronic systems from the top down. By combining functional description and electronic design details, they provide a comprehensive optimisation approach, reducing development time and improving the quality of the final product. G. Cauwenberghs *et al*. (2023) pointed out that the automation of micro/nano circuits and system design is key to advances in information processing, but traditional methods face limitations in size, power consumption, and complexity. To overcome these challenges, new solutions such as machine learning, artificial intelligence, specific accelerators, and nanotechnology are needed, requiring interdisciplinary research and collaboration. Confirmation of expectations regarding AI and big data analytics was provided by the founder of the educational platform techexplorations.com (Generative AI..., 2024). The author noted that artificial intelligence has become a key force in enhancing the functionality and aesthetics of electronic products while minimising production costs and energy consumption. AI development applications and simulation technologies allow engineers to visualise and test products in virtual environments, significantly reducing material waste and supporting sustainability practices. They optimise the placement of components on printed circuit boards, making designs more compact and energy efficient. Sensor data analysis and fault prediction help identify potential problems before they occur, minimising rejects and repair costs.

In the electronics industry, artificial intelligence and EDA systems are also becoming key factors in development. Practical applications of generative AI in electronics include circuit design optimisation, power management, and fault detection. AI-driven CAD applications are leading to significant improvements in design development, verification, and implementation.

CONCLUSIONS

The study systematises and analyses information on available solutions for selecting software for modelling and simulating electrical circuits, addressing user experience. The study contains a general classification of EDA software, a theoretical overview of the physical and mathematical principles underlying the software, and a brief description of their algorithms, methods and key characteristics. The result obtained in the form of a comparative analysis of the key characteristics and functionalities of LTspice, MATLAB/Simulink, Multisim, Proteus, KiCad, and EAGLE software packages is intended to inform users in their choice, helping to optimise work, increase productivity, and save time and money.

The limitations of the study were: quantitative analysis data is limited to EDA developer websites and literature reviews, which may lead to insufficient completeness of data on modelling speed, accuracy of results, and ease of use; qualitative analysis is based on subjective aspects, which may affect the objectivity of the conclusions; qualitative analysis used a randomly available sample of respondents who agreed to participate, which may not represent the entire user community and affect the generalisability of the conclusions. Some programmes may have been excluded from the analysis due to their unavailability at the time of the study. The review of the commercial availability of software may not be complete due to constant changes in the software market and the emergence of new products. The use of EDA developers' websites to collect information may lead to bias, as these sources may not provide objective information about the shortcomings of their products.

Future research should focus on expanding and more detailed selection of programme evaluation criteria to cover all aspects of the use of EDA tools, monitoring their updates to take into account current technological capabilities. To confirm the results of the analysis of EDA programmes, it is useful to study the impact on engineering practice.

ACKNOWLEDGEMENTS

None.

CONFLICT OF INTEREST

None.

REFERENCES

[1] Andriienko V., & Bondarenko, M. (2018). Robotechnics systems in technical education. In *Abstracts of the 4th international scientific and practical conference "Information Technologies in Education, Science and Technology"* (pp. 44-45). Cherkasy: ChSTU.

[2] Arseniuk, D., & Zinkovsky, Yu. (2023). Model of bridgeless totem pole power factor corrector using wideband semiconductor devices. *Scientific Notes of Tavriya National University Vernadsky*, 34(6), 317-322. doi: 10.32782/2663-5941/2023.6/48.

[3] Bahar, R.I., Jones, A.K., Katkoori, S., Madden, P.H., Marculescu, D., & Markov, I.L. (2020). *Workshops on extreme scale design automation (ESDA) challenges and opportunities for 2025 and beyond*. Washington, DC: Computing Community Consortium.

[4] Cauwenberghs, G., Cong, J., Hu, X.Sh., Joshi, S., Mitra, S., Porod, W., & Wong, H.-S.P. (2023). Micro/nano circuits and systems design and design automation: Challenges and opportunities. *Proceedings of the IEEE*, 111(6), 561-574. doi: 10.1109/JPROC.2023.3276941.

[5] Charig, N. (2021). *The evolution of electronic design automation technology*. Retrieved from https://www.power-and-beyond.com/the-evolution-of-electronic-design-automation-technology-a-95ff88f226bb89c338b6d09047fc6d27/.

[6] Chen, J., Jiang, I.H.-R., Jung, J., Kahng, A.B., Kravets, V.N., Li, Y.-L., Lin, S.-T., & Woo, M. (2020). DATC RDF-2020: Strengthening the foundation for academic research in IC physical design. In *ICCAD '20: Proceedings of the 39th international conference on computer-aided design* (article number 71). New York: Association for Computing Machinery. doi: 10.1145/3400302.3415742.

[7] Cirstea, M., Benkrid, K., Dinu, A., Ghiriti, R., & Petreus, D. (2024). Digital electronic system-on-chip design: Methodologies, tools, evolution, and trends. *Micromachines*, 15(2), article number 247. doi: 10.3390/mi15020247.

[8] Curiac, C.-D., & Doboli, A. (2022). Combining informetrics and trend analysis to understand past and current directions in electronic design automation. *Scientometrics*, 127(10), 5661-5689. doi: 10.1007/s11192-022-04481-9.

[9] Dang, P., & Arolkar, H. (2019). Electronic design automation tool: A comparative study. *International Journal for Research in Applied Science & Engineering Technology*, 7(3), 2138-2144. doi: 10.22214/ijraset.2019.3395.

[10] Dosymov, Ye., Usembayeva, I., Polatuly, S., Ramankulov, S., Kurbanbekov, B., Mintassova, A., & Mussakhan, N. (2023). Effectiveness of computer modeling in the study of electrical circuits: Application and evaluation. *International Journal of Engineering Pedagogy*, 13(4), 93-112. doi: 10.3991/ijep.v13i4.34921.

[11] Generative AI for electronic circuit design – an exploration. (2024). Retrieved from https://techexplorations.com/blog/artificial-intelligence/ai-in-circuit-design/.

[12] Grau, S., Lopez, J.M.D., Roychowdhury, D., Chachowski, J., & Stoll, E. (2024). *Design automation of embedded air coils for CubeSat attitude control*. Retrieved from https://www.researchgate.net/publication/381708730_Design_Automation_of_Embedded_Air_Coils_for_CubeSat_Attitude_Control.

[13] Gubbi, K.I., Beheshti-Shirazi, S.A., Sheaves, T., Salehi, S., Manoj, S., Rafatirad, S., Sasan, A., & Homayoun, H. (2022). Survey of machine learning for electronic design automation. In *GLSVLSI '22: Proceedings of the Great Lakes symposium on VLSI 2022* (pp. 513-518). New York: Association for Computing Machinery. doi: 10.1145/3526241.3530834.

[14] Hashaam, Z. (2024). *Interface of RAM in proteus demonstration*. doi: 10.13140/RG.2.2.14168.06406.

[15] Huang, G., *et al*. (2021). *Machine learning for electronic design automation: A survey*. Retrieved from https://arxiv.org/pdf/2102.03357v2.

[16] Islam, M.N., Alam, M.S., & Haque, M.A.S. (2022). Development of eagle multi-layer printed circuit board with CAD and CAM. *International Journal of Systems Signal Control and Engineering Application*, 14(6), 77-80.

[17] Kahng, A.B. (2022). A mixed open-source and proprietary EDA commons for education and prototyping. In *ICCAD '22: Proceedings of the 41st IEEE/ACM international conference on computer-aided design* (article number 17). New York: Association for Computing Machinery. doi: 10.1145/3508352.3561378.

[18] Li, L., Meng, L., & Wang, F. (2021). Design and simulation of frequency divider circuit based on multisim. *E3S Web of Conferences*, 268, article number 01058. doi: 10.1051/e3sconf/202126801058.

[19] Mareai, M.A.A. (2024). *Basic electronic circuits by proteus simulating software*. Retrieved from https://www.researchgate.net/publication/381260714_Basic_Electronic_Circuits_by_Proteus_Simulating_Software.

[20] Marinova, G., & Bitri, A. (2021). Data analysis environment to study the dynamics in electronic design automation industry. *IFAC-PapersOnLine*, 54(13), 528-532. doi: 10.1016/j.ifacol.2021.10.503.

[21] Nerubatskyi, V., Plakhtii, O., Hordiienko, D., Philipjeva, M., & Bagach, R. (2023). Improving of simulation accuracy of transient processes and calculation of power losses of semiconductor converters in the NI Multisim program. *Information and Control Systems at Railway Transport*, 28(2), 22-35. doi: 10.18664/ikszt.v28i2.283312.

[22] Osadchuk, O., Krylyk, L., Zviahin, O., & Osadchuk, Ya. (2021). Mathematical model of a microelectronic humidity transducer with a humidity-sensitive resistive element. *Scientific Notes of Tavriya National University Vernadsky*, 32(1), 175-182. doi: 10.32838/2663-5941/2021.1-2/28.

[23] Semenikhina, O., Drushlyak, M., Lynnyk, S., Kharchenko, I., Kyryliuk, H., & Honcharenko, O. (2020). On computer support of the course "fundamentals of microelectronics" by specialized software: The results of the pedagogical experiment. *TEM Journal*, 9(1), 309-316. doi: 10.18421/TEM91-43.

[24] Shamonia, V., Semenikhina, O., Proshkin, V., Lebid, O., Kharchenko, S., & Lytvyn, O. (2019). Using the Proteus virtual environment to train future IT professionals. *Educational Dimension*, 1, 181-198. doi: 10.31812/educdim.v53i1.3842.

[25] Tolochko, O. (2020). *Application software packages for PC MATLAB, SIMULINK, SimPowerSystems*. Kyiv: National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute".

[26] Tripathi, S.L., Saxena, S., Sinha, S.K., & Patel, G.S. (2022). *Digital VLSI design and simulation with Verilog*. London: John Wiley & Sons. doi: 10.1002/9781119778097.

[27] Venkateswara, M., Rao, Hemanth, K., Razia, S., & Kumar, K.R. (2020). Synthesizing of hand-drawn electrical circuits using machine learning techniques. *International Journal of Emerging Trends in Engineering Research*, 8(9), 5523-5529. doi: 10.30534/ijeter/2020/100892020.

[28] Yang, W.Y., *et al*. (2020). *Electronic circuits with MATLAB, PSpice, and Smith chart*. London: John Wiley & Sons.

Дослідження ефективності програм для моделювання та симуляції електричних схем та їх порівняння за точністю, швидкістю та зручністю використання

Лариса Васецька

Кандидат технічних наук, доцент
Луганський національний університет імені Тараса Шевченка
36000, вул. Івана Банка, 3, м. Полтава, Україна
https://orcid.org/0000-0002-4873-8835

Анотація. Зростаюча складність мікросхем та обмеження мініатюризації роблять процес проектування електроніки складнішим та трудомісткішим. Проблема втрати сили законів Мура спричиняє стрімкий розвиток у сфері проектування та виробництва електронних компонентів. Метою дослідження було надати структуровану інформацію про інструменти проектування електричних схем за функціональними можливостями та доступністю, щоб сприяти оптимізації роботи, підвищенню продуктивності та економії. Виконано теоретичний опис принципів роботи програм для моделювання та симуляції електричних схем, огляд алгоритмів та методів, що використовуються в програмах автоматичного проектування, класифікацію програм за ключовими характеристиками та порівняльний аналіз популярних програмних пакетів: LTspice, EAGLE, MATLAB/Simulink, Multisim, Proteus, KiCad. Аналіз враховував швидкість моделювання, точність результатів, легкість використання, функціональні можливості та доступність, а також досвід користувачів. Рекомендовано вибирати програмне забезпечення залежно від потреб користувача, складності проекту та бюджету, з урахуванням досвіду інших користувачів. Для початківців та студентів рекомендуються LTspice, Multisim та KiCad: LTspice – потужний, безкоштовний і простий у використанні, KiCad пропонує відкритий код та активну спільноту, а Multisim підходить для навчання. MATLAB/Simulink, Proteus та EAGLE рекомендовані для професіоналів: Simulink забезпечує потужне моделювання та інтеграцію з іншими продуктами MathWorks; Proteus оптимальний для розробників мікроконтролерних систем; EAGLE підходить для інтеграції з системами автоматизованого проектування та автоматизації. Практична цінність результатів дослідження полягає у створенні рейтингу EDA-програм за критеріями функціональності та продуктивності, покликаного надати користувачам допомогу у виборі на основі їхніх потреб

Ключові слова: інструменти для проектування електроніки; аналіз схем; бібліотека компонентів; підтримувані стандарти; інтерфейс

BULLETIN of Cherkasy State Technological University

Journal homepage: https://bulletin-chstu.com.ua/en Vol. 29 No. 2. 2024

UDC 004.91
DOI: 10.62660/bcstu/2.2024.45

Cross-platform adaptation of algorithmic editing techniques

Xiang Li*
Postgraduate Student
Chiang Mai University
50200, 239 Huay Kaew Rd., Chiang Mai, Thailand
https://orcid.org/0009-0000-7073-3646

Tamprasirt Anukul
PhD
Chiang Mai University
50200, 239 Huay Kaew Rd., Chiang Mai, Thailand
https://orcid.org/0009-0005-5424-1170

Fangli Ying
PhD
Chiang Mai University
50200, 239 Huay Kaew Rd., Chiang Mai, Thailand
https://orcid.org/0000-0001-8390-3229

Abstract. The research relevance is determined by the rapid development of technology and the growing need for efficient data processing on various platforms. The study aimed to address methods that would enable the use of data editing algorithms on various operating systems and hardware platforms. A methodology was developed for studying cross-platform adaptation technologies, including cross-compilation, virtualisation, the use of universal libraries and Application Programming Interface, as well as methods for testing and optimising algorithm performance. The study addressed various approaches to implementing cross-platform compatibility, including the use of cross-compilation, virtualisation and containerisation. The main technical challenges are managing resources, optimising performance and ensuring compatibility with hardware from different platforms. The principles included selecting the most appropriate technology for the task at hand, considering performance and security requirements, and ensuring effective integration of existing systems and infrastructure. The workflows are focused on creating modular and extensible solutions that can easily adapt to changes in the technological environment and user requirements. In the context of this study, artificial intelligence software plays a key role in improving the efficiency and accuracy of data processing across different platforms. The results showed that artificial intelligence software can automate various stages of the video and audio editing process. Artificial intelligence is used to analyse large amounts of content data, such as video files, images and audio recordings. The study determined that artificial intelligence is increasingly relevant in various aspects of movie production. Artificial intelligence can analyse scripts, predict their potential success and suggest improvements using data from previous films and their commercial success

Keywords: optimisation; software; efficiency; compatibility; interface

Article's History: Received: 14.12.2023; Revised: 26.03.2024; Accepted: 27.05.2024.

INTRODUCTION

Modern multimedia production and cinematography require the efficient use of algorithmic editing technologies to create high-quality content. However, the diversity of operating systems and hardware platforms poses a challenge for developers to ensure that algorithms are compatible and efficient across devices. In

Suggested Citation:
Li, X., Anukul, T., & Ying, F. (2024). Cross-platform adaptation of algorithmic editing techniques. *Bulletin of Cherkasy State Technological University*, 29(2), 45-56. doi: 10.62660/bcstu/2.2024.45.

*Corresponding author

this context, the cross-platform adaptation of algorithmic editing methods becomes a necessary component to ensure the versatility and reliability of media content processing and editing solutions. The modern requirements to produce media content such as films, videos and animation imply the use of highly efficient algorithms for processing and editing video, audio and graphics. However, the diversity of devices and technology platforms creates challenges in ensuring the same quality and performance across all target devices. The issue includes several key aspects, each of which is important and requires special attention when developing and implementing multimedia technologies.

Firstly, differences in the hardware characteristics of different platforms are a critical factor. Each operating system and device have unique features in terms of processor, memory, graphics and input/output. This requires in-depth optimisation of algorithms for the specific technical conditions of each platform to ensure optimal performance and quality of media content processing. Secondly, the compatibility and interoperability of algorithms between different operating systems and devices are equally relevant. From Windows and mac Operation System (OS) to Linux and mobile platforms such as iOS and Android, it is worth addressing the differences in software and ensuring that algorithms work without losing functionality and performance on each platform. The third aspect concerns the efficient use of resources. Algorithms for algorithmic editing should be designed to efficiently use the available computing and memory resources of devices. This is especially relevant for large data volume processing, where it is necessary to ensure fast and stable information processing without significant resource consumption. Ensuring data security and protection is a critical aspect of adapting algorithmic editing methods. When dealing with confidential information or personal data, security standards must be strictly adhered to prevent data leaks or unauthorised access to sensitive information.

The problem in this area is the need for effective integration and compatibility of algorithms on different platforms. J.Z. Blanco & D. Lucrédio (2021) proposed a holistic approach to cross-platform software development. Y. Ma *et al.* (2022) addressed support for cross-platform real-time collaborative programming, describing the architecture, methods, and prototyping system. Areas requiring further research include effective methods for integrating algorithmic editing across platforms, optimising performance and ensuring interoperability. The need to optimise algorithms to ensure uniform performance across different hardware and software platforms is also an important task. I. Bieda & T. Panchenko (2022) analysed various artificial intelligence (AI) tools, their functionalities, as well as advantages and limitations when used on different platforms. R. Eugeni & P. Pisters (2020) explored how modern algorithms are changing approaches to video creation and editing, and how these changes affect the perception and use of video content in different environments. Ensuring the security and protection of data when using algorithms on different platforms requires further study. Another pressing issue in this area is the need to create efficient and compatible algorithms that can run seamlessly on different platforms. M. Sajbidor *et al.* (2023) considered the creation of a cross-platform application in Java and C++, which demonstrates the opportunities and challenges associated with the implementation of algorithms on different systems. H. Nguyen (2022) studied the compatibility of programming languages in the context of cross-platform development, emphasising the difficulties that arise when trying to ensure the smooth operation of software on different platforms. The development of universal standards for cross-platform interaction of algorithms requires further study. Another issue that requires further study is the need to develop effective strategies for testing and debugging algorithms on different platforms. Y. Zhang (2021) examined cross-platform methods in computer graphics that facilitate the creation of experimental films. N. Dey (2021) has studied cross-platform development using Qt 6 and modern C++, which allows implementing algorithms on different systems. The development of methods for automatic detection and elimination of errors in the cross-platform operation of algorithms requires additional consideration. One of the key issues is the need to develop effective strategies for managing resources and optimising performance when working with algorithms on different platforms. C. Rieger & T.A. Majchrzak (2019) explored ways to create a definitive framework for evaluating approaches to cross-platform application development, which focuses on the complexities and challenges of this process. A.A. Menegassi & A.T. Endo (2020) addressed automated tests of cross-platform mobile applications in various configurations, emphasising the importance of testing algorithms on different devices and environments. Another issue that requires further study is the need to develop effective methods for adapting the user interface and visualising algorithmic editing for different devices and screens.

The study aimed to analyse and create approaches that would allow data editing algorithms to be effectively applied on different operating systems and hardware platforms. The following tasks were set as part of this goal: analysing existing technologies and tools that support cross-platform data editing and studying the technical features and challenges associated with adapting algorithms for different platforms.

MATERIALS AND METHODS

The study is conducted on an analysis of the advantages and limitations of each technology, such as performance losses, and difficulty in integrating or maintaining functionality across different platforms.

The study included a detailed analysis of existing approaches to cross-compilation, virtualisation, the use of universal libraries and Application Programming Interface (APIs), as well as methods for testing and optimising algorithm performance on different platforms. The analysis of technologies included an assessment of their applicability and effectiveness for ensuring that data editing algorithms work on different operating systems and devices. Particular attention is devoted to an analysis of the limitations of technologies, such as performance losses, difficulty in maintaining functionality across platforms and the need for additional resources for integration and use.

The study examined cross-platform tools such as Qt, Xamarin and Flutter. The capabilities and limitations of these tools for developing cross-platform applications were assessed. Qualitative analysis methods, such as case studies, were used to gain an in-depth understanding of the experiences and perspectives of people involved in the development of algorithmic editors. The following experts took part in the study:

1. Lead developer (software developer), 10 years of experience.

2. Project manager (project manager), 8 years of experience.

3. Test engineer (software tester), 6 years of experience.

4. Chief developer (software developer), 12 years of work experience.

5. Integration specialist (integration engineer), 7 years of work experience.

These interviews were used to gather detailed information about practical experience of using cross-platform adaptation technologies, identify their strengths and weaknesses, and receive recommendations on how to improve and optimise cross-platform application development processes.

The study included a survey that helped to determine the current state of cross-platform adaptation of algorithmic editing methods. The survey involved 50 professionals, including 23 women and 27 men, working in programming and software development, specialising in algorithmic editing. The survey included 4 blocks of questions (Table 1).

Table 1. Blocks of questions in the questionnaire

Block of questions	Questions
The first block	Do you use algorithmic editing methods in your work?
The second block	Have you encountered any difficulties in adapting algorithmic methods for different operating systems?
The third block	Have you ever had to change your algorithms depending on the platform on which they are executed?
The fourth block	Do you think it is important to ensure cross-platform compatibility in your project?

Source: compiled by the authors

In addition to these methods, the study adhered to strict ethical standards. All participants were informed about the purpose of the study and how their answers would be used. Confidentiality and anonymity of respondents were guaranteed. The study was conducted voluntarily, without discrimination based on religion, race or gender. The ethicality of the study was supported by references to relevant standards, including professional codes of ethics and the European Commission's Guidelines on Ethics and Data Protection (2021).

RESULTS

Cross-platform adaptation technologies play a key role in modern software development, enabling the creation of applications that run on different operating systems and hardware platforms. In a rapidly changing technological landscape where users expect applications to work seamlessly across all devices, cross-platform solutions are becoming essential. These technologies not only simplify the development process but also significantly reduce the time to market, providing a competitive advantage. The main technologies for cross-platform adaptation are cross-compilation, virtualisation and containerisation, as well as the use of cross-platform libraries and frameworks.

Cross-compilation is the process of compiling software code on one platform (called the host platform) for execution on another platform (called the target platform). This approach allows developers to create software for a variety of devices and operating systems, even if they do not have direct access to the target hardware. Cross-compilation is especially useful when developing applications for mobile devices or embedded systems. The benefits of cross-compiling include reduced development time, as developers can use powerful host systems to compile code, which significantly speeds up the process compared to compiling on less powerful target devices. The ability to test and debug code on the host platform makes it easy to identify and fix errors without the need for access to the target hardware. In addition, cross-compilation allows developers to create software for many different architectures and operating systems, such as ARM, x86, Windows, macOS and Linux.

The use of cross-compilers such as GNU Compiler Collection (GCC) and Low Level Virtual Machine (LLVM) is a common practice. GCC supports a wide variety of programming languages and platforms, providing flexibility and broad applicability. LLVM, being a modular and flexible compiler, also supports various architectures and is used in Xcode for macOS and iOS development. Cross-compiling reduces development costs and simplifies software support, as the same code can be used for different platforms with minimal changes. This is especially relevant in the modern software development environment, where applications must be available on a variety of devices and operating systems, ensuring the same quality and performance.

Virtualisation and containerisation can be used to create isolated environments for running applications, which simplifies portability and compatibility between different operating systems and hardware platforms (Casalicchio & Iannucci, 2020). Virtualisation is the creation of virtual machines (VMs) that emulate separate hardware, allowing multiple operating systems to run on a single physical server. This provides flexibility and isolation, as each virtual machine operates independently, minimising the impact of one VM on the others. Containerisation, in turn, uses containers that provide lightweight, isolated environments for applications. Unlike VMs, containers do not emulate an entire operating system but use the kernel of the host OS, which makes them more efficient in terms of resource usage. Containers contain all the necessary components to run an application, including code, libraries, and dependencies, which ensures consistent behaviour in any environment.

Examples of virtualisation tools include VMware and VirtualBox, which can be used to run VMs with different operating systems on the same physical server. Examples of containerisation tools include Docker and Kubernetes. Docker provides a platform for creating, distributing, and running containers, ensuring their isolation and ease of deployment. Kubernetes, in turn, automates the deployment, scaling and management of containerised applications, providing high availability and scalability.

The main benefits of virtualisation and containerisation include the isolation of environments, which avoids conflicts between dependencies of different applications and simplifies the deployment and scaling of applications. These technologies improve safety, as isolated environments minimise the risk of impacting other parts of the system. They also provide high resource efficiency, as containers consume less memory and central processing unit (CPU) resources than VMs. The use of virtualisation and containerisation is key to modern Information Technology (IT) infrastructures, especially in the context of cloud computing and DevOps practices. These technologies enable developers and system administrators to effectively manage complex, multi-component applications, ensuring their reliability, security and performance across multiple platforms and environments.

Cross-platform libraries and frameworks allow developers to create applications that run on different operating systems with minimal changes to the code (Ameen & Mohammed, 2022). This is achieved using common APIs and development tools, which greatly simplifies the process of developing and maintaining applications for different platforms. Such libraries and frameworks provide unified interfaces and abstractions that hide the differences between operating systems, allowing developers to focus on the logic of the application rather than the specifics of each platform.

Examples of cross-platform tools include Qt, Xamarin, and Flutter. Qt is a powerful framework for creating graphical user interface applications that support multiple platforms, including Windows, macOS, Linux, iOS, and Android. It provides a set of tools and libraries for developing graphical interfaces, working with files, networking and much more, creating complex applications that look and work the same on all supported platforms.

Xamarin is a C# mobile application development platform that can be used to create applications for iOS and Android using a common code (Toasa *et al.*, 2023). Xamarin provides access to the native APIs of each platform, which provides all the features and functions of the operating systems, providing high performance and a native user experience.

Flutter is a framework from Google for creating native applications for mobile, web and desktop platforms using a single code in the Dart language (Kishore *et al.*, 2022). Flutter includes a substantial set of tools for creating animations and user interfaces, thus providing tools to create beautiful and responsive applications. One of the key advantages of Flutter is its ability to compile code into native machine code, which ensures high application performance.

The main benefits of using cross-platform libraries and frameworks include reduced development time and costs due to code reuse, a consistent user experience across platforms, and ease of maintenance and updating of applications. These tools allow developers to create high-quality applications that run on a variety of devices and operating systems while ensuring high performance and reliability. The study created a diagram showing the importance of different technologies in the process of cross-platform adaptation of algorithmic editing methods (Fig. 1).

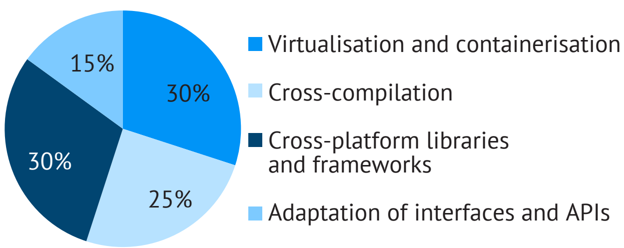


Figure 1. The importance and contribution of various technologies to the process of cross-platform adaptation of algorithmic editing methods

Source: compiled by the authors based on K. Vassallo *et al.* (2019)

The technical features and challenges of cross-platform adaptation of algorithmic editing methods are complex and multifaceted tasks that require careful analysis and optimisation at different levels of development. Each platform has unique features in terms of processor, memory, graphics and data input and output. These differences require the optimisation of algorithms for the specific technical conditions of each platform. For instance, applications designed for desktop computers with powerful processors and large amounts of Random Access Memory may experience performance issues on mobile devices with more limited resources. Optimisation includes adapting algorithms to make efficient use of multithreading, minimising memory usage and improving graphics performance.

Ensuring that algorithms work on different operating systems, such as Windows, macOS, Linux, and mobile platforms (iOS, Android), requires implementation of compatibility and interoperability between different systems. This includes support for various system calls, file systems, network protocols and interfaces. Interoperability can be achieved using standard libraries and frameworks that abstract low-level details and provide unified APIs. Algorithms must efficiently use the available computing and memory resources of devices to ensure fast and stable operation when processing large amounts of data. This includes optimising code to reduce memory and CPU time, using hardware acceleration (e.g. GPUs) and ensuring low latency for interactive applications. Efficiencies can be achieved by profiling and analysing application performance across platforms and making changes accordingly. Data protection and security standards are essential when adapting algorithmic editing techniques, especially when dealing with sensitive information or personal data. This includes protecting data in transit and storage, managing access and authenticating users, and protecting against vulnerabilities and attacks. Security can be enhanced using encryption, multi-factor authentication and regular updates and patches to address vulnerabilities.

The results of the first block of the survey on the use of algorithmic editing methods show that out of 50 survey participants, 46% (23 respondents) regularly use algorithmic editing methods in their work. This confirms the high level of prevalence and importance of such approaches among professionals in the field of programming and software development. The remaining 54% (27 respondents) do not use algorithmic editing methods. This observation is relevant as it indicates the diversity of approaches and preferences in software development. Some professionals may prefer other methods or see no need to use algorithmic editing in their current work. These data indicate the need for further study and optimisation of algorithmic editing methods.

The results of the second block of the survey show that 62% (31 respondents) reported that they had encountered difficulties in adapting algorithmic methods for different operating systems. This shows the prevalence of cross-platform compatibility issues in software development. Such challenges may include API incompatibilities, differences in programming language standards, or hardware platforms. The remaining 38% of participants (19 respondents) did not encounter such difficulties, indicating fewer cross-platform compatibility issues.

In the third block of the survey, 50% (25 respondents) noted that they had to change algorithms depending on the platform on which they were executed. This highlights the importance of adapting algorithms to the specific technical features and requirements of different platforms, which can affect their efficiency and performance. The remaining 50% (25 respondents) did not express this need, which indicates that they were working on projects where the algorithms remained unchanged regardless of the platform. The results of the fourth block of the survey show that 78% (39 respondents) consider cross-platform compatibility to be an important or very important aspect of their project. The remaining 22% (11 respondents) stated that this is a less important aspect of software development for them. Based on the survey, a histogram was developed to visualise the key results and reflect the distribution of respondent opinions and experiences (Fig. 2).

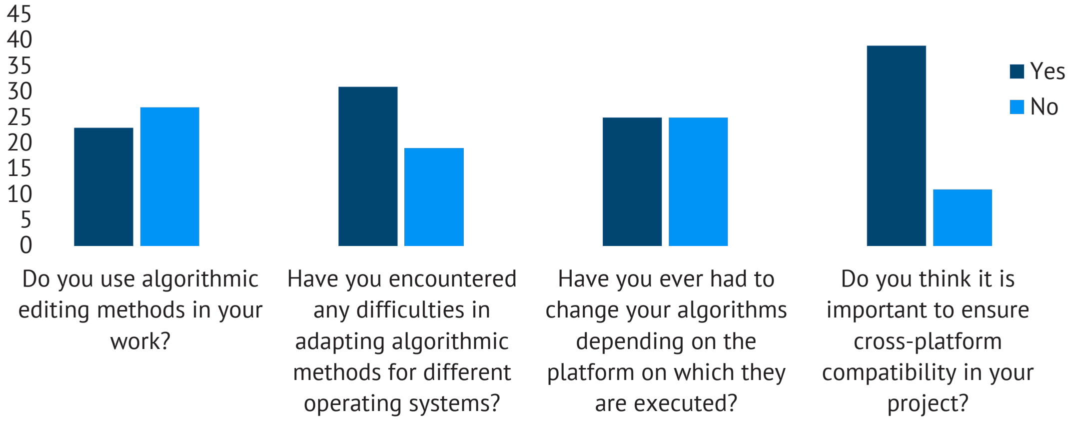


Figure 2. Results of answers to questions regarding the use of algorithmic editing methods
Source: compiled by the authors

Cross-platform adaptation of algorithmic editing methods requires addressing several technical features and challenges that need to be considered for the effective operation of applications on different operating systems and hardware platforms. One of the key aspects is the modularity and extensibility of solutions. Creating modular systems allows developers to update or replace individual software components without having to rewrite the entire system. This is especially relevant for cross-platform applications, where the need to adapt to different technical specifications and changes in the technology stack can arise at any time.

Another aspect is unification and standardisation. The use of unified standards and protocols, such as POSIX for operating systems, OpenGL for graphics, or JSON for data exchange, simplifies development and ensures application compatibility across platforms. This reduces development and support costs by reducing the number of adaptations and tests required on different devices. Another important aspect is performance optimisation. Cross-platform applications often require the adaptation of algorithms to different computing resources and hardware accelerators. For example, the use of Compute Unified Device Architecture for GPU-accelerated computing or specialised hardware accelerators for performing complex operations in real-time. This ensures high application performance across devices while maintaining functionality and efficiency. All these aspects combined ensure successful cross-platform adaptation of algorithmic editing methods, addressing a wide array of technical platforms and user requirements. This is necessary to create universal and stable solutions for processing media content and other applications running on different devices and operating systems.

Content analysis and optimisation using AI is a process that involves the automatic study and processing of large amounts of content data to improve its quality, style and relevance to the target audience across multiple platforms. Key examples of algorithms used for content optimisation include:

- Automatic colour correction – this algorithm automatically adjusts the colour palette of images or videos to achieve the desired visual effect. It can be used to eliminate colour deviations, balance hues, and adjust brightness and contrast, which is important for ensuring consistency and aesthetic perception of content across devices.
- Noise reduction – this algorithm aims to reduce noise in images or videos. Noise can occur due to insufficient lighting, poor sensor quality, or other technical reasons. The use of noise reduction techniques improves the clarity and quality of images, which is particularly important for maintaining the detail and clarity of images in different lighting conditions.
- Image quality enhancement – this algorithm includes various techniques aimed at increasing the clarity, contrast and detail of images (Li *et al.*, 2022). It can use resampling, filtering and edge enhancement techniques to enhance the overall visual experience of content, which also contributes to its aesthetic quality and appeal to viewers.

These algorithms work by analysing large amounts of content data to automatically optimise content for various purposes, from improving visuals to making it more relevant to audience requirements and platform specifications.

AI in the media is used in various aspects of the production, processing, distribution and consumption of multimedia content. The main areas of AI use in media include process automation, improving content quality, personalising user experience, creating new content products and optimising content management. The first area is content analysis. AI is used to recognise and classify media data such as images, video and audio files. This includes recognising objects and faces in images, analysing video content and detecting audio files, making it easier to automate the indexing and organisation of large volumes of content. The second important area is algorithmic editing. AI is used to automatically process and improve the quality of media content. Examples include automatic colour correction, noise reduction, sharpening and image detail, which help to achieve the desired visual effects and deliver high-quality content across multiple platforms and devices. The third area is content personalisation. AI is used to adapt media content to individual user needs and preferences. This includes the development of recommender systems that offer personalised content recommendations based on the analysis of user behaviour and preferences, which increases audience engagement and satisfaction with media services. The fourth area is content creation. AI is used to automatically create new media content, such as generating texts, images, music and videos. This includes the creation of various forms of multimedia content by algorithms, which promotes innovation in the field of creativity and expands the possibilities for creating unique content products. The fifth aspect is content management and distribution. AI helps to improve digital library management, copyright control, audience analysis, and optimisation of content distribution across various platforms and channels, which increases the efficiency of media resources and increases their commercial value.

The use of AI in the media significantly increases the efficiency of production processes, improves user experience and contributes to the creation of innovative solutions aimed at meeting the growing needs and expectations of modern users of multimedia technologies (de-Lima-Santos & Ceron, 2022). Synchronisation problems on TV can occur for several reasons related to the technical characteristics of different platforms and devices. These problems can affect the quality of content perception and require special measures to prevent them. One of the main reasons is the difference in hardware characteristics of the devices. Each device has unique technical features, such as processor, memory, graphics capabilities and audio systems. These differences may result in inconsistencies in the playback of audio and video content. Network delays also play an important role in synchronisation problems. Differences in Internet connection speed and network stability can cause delays in data transmission, which is especially critical when streaming video and audio. Differences in the codecs and compression formats used to compress and decompress media files can lead to incompatibilities and additional delays when playing content on different platforms.

Another reason is software errors and bugs that occur in media content processing and playback systems. These errors can lead to incorrect synchronisation of audio and video signals, which can degrade the viewing experience. Adapting content for different devices and screens can also pose challenges. For example, content optimised for one type of screen may not display or play properly on another device. Various measures are taken to prevent these problems. One approach is to use unified standards and protocols for encoding and transmitting media data. This helps to ensure compatibility between different platforms and devices. Optimising algorithms for processing media content is also important. The use of hardware accelerators such as GPUs and FPGAs can significantly increase data processing speeds and reduce latency. Testing and debugging software on different platforms helps to identify and fix synchronisation issues before the product is released. This includes the use of automated tests and emulators of various devices and operating systems. It is also necessary to implement adaptive streaming mechanisms that can automatically adjust video and audio quality depending on the speed of the Internet connection and device characteristics.

Deep Fake is a technology that allows the creation of realistic fakes of videos and images using AI (Kwok & Koh, 2021). The process of creating a deep fake usually involves the following steps. The first stage is data collection. Deep fake creation requires large amounts of data, such as images or videos of faces to be replaced or synthesised. This data serves as the basis for model training, so its quality and diversity play a key role in the final result. This is followed by the data pre-processing stage. At this stage, the images and video are aligned and normalised to improve the quality of the model training. This can include cropping faces, adjusting lighting and colour, and removing noise. The purpose of pre-processing is to prepare the data so that the neural network can train efficiently, minimising errors. After that, the model training phase begins. Generative adversarial networks (GANs) are used in this stage, which consists of two neural networks – a generator and a discriminator. The generator creates fake images, and the discriminator tries to distinguish them from the real ones. These two networks are trained in a competitive mode: the generator tries to deceive the discriminator by creating more and more realistic images, and the discriminator improves its skills in recognising fakes. The training process continues until the generator learns to create sufficiently realistic images that the discriminator cannot distinguish them from real ones.

The next step is to create fake images or videos. The trained generator is used to create realistic images or videos by replacing the face in the original with the face of another person. In the case of video, this also requires lip movement and facial expressions to be synchronised to achieve maximum realism. At this stage, additional algorithms are often used to improve the quality, such as anti-aliasing filters and artefact correction. The final stage is post-processing. Includes final adjustments and enhancements to the video or image quality to eliminate artefacts and improve realism. This can include manual editing, the use of specialised tools to enhance images and audio, and additional checks for realism. Post-processing brings a fake video or image to a level where it will be difficult to distinguish it from the real material, even with careful analysis. Thus, the process of creating a deep fake involves several key stages, each of which plays an important role in ensuring the high realism of the final product. Data collection and pre-processing, training of GANs, creation of fake images and videos, and post-processing all combine to create high-quality fakes that can deceive even the most careful observers.

Creating Deep Fakes requires the use of various technologies, each of which is required in the process of generating realistic fakes. GANs are the main technology used to create Deep Fake (Revi *et al.*, 2021). GANs consist of two neural networks: generator and discriminator. Autoencoders also play an important role in the creation of Deep Fake. They are used to reduce data dimensionality and extract key facial features. Autoencoders consist of two parts: an encoder that converts the input data into a compact representation, and a decoder that recovers the original data from this representation. This simplifies the face replacement process, as autoencoders can effectively learn from face characteristics and then reproduce them with high accuracy.

Open-Source Computer Vision Library (OpenCV) is a computer vision library that is widely used in the process of creating Deep Fake. OpenCV provides a variety of tools for image and video processing, such as face detection, object tracking, lighting and colour correction, and noise filtering. These tools are necessary for pre-processing data to prepare it for training neural networks. TensorFlow and PyTorch are popular machine-learning frameworks used to create and train neural networks, including GANs and autoencoders (Novac *et al.*, 2022). These frameworks provide powerful tools and libraries that make it easy to develop and customise machine learning (ML) models. They support parallel computing and the use of graphics processing units (GPUs), which significantly speeds up the learning process and enables large data amount processing. Dlib is a machine-learning library that includes tools for detecting faces and other objects in images (Xu *et al.*, 2020). It is widely used for pre-processing data and preparing it for training. Dlib provides highly accurate algorithms for face detection, image alignment and normalisation, which improves the quality of training and increases the accuracy of the fake images created. FaceSwap and DeepFaceLab are specialised tools and software for creating Deep Fake. They integrate the technologies described above and provide a user-friendly interface for working with them. These tools automate many aspects of Deep Fake creation, such as data collection, model training, fake image and video creation, and post-processing. FaceSwap and DeepFaceLab use GANs, auto-encoders, OpenCV, TensorFlow and PyTorch to ensure high realism and quality of the final product. Used together, these technologies and tools can be used to create Deep Fake with a high degree of realism, rendering them substantial in the hands of developers. However, they also raise important ethical and security issues, as they can be used to create misinformation and fraud.

Deep Fake detection and recognition is a complex task that requires the use of various technologies and methods. When creating Deep Fake, artefacts such as blurring, unnatural object boundaries, unrealistic lighting and shadows often occur. Deep Fake detection algorithms determine such anomalies in images and videos. Fake videos can have unnatural movements or behaviour of objects, such as inconsistent lip movements when words are spoken or facial expressions that do not match the words spoken. These features can be used to identify counterfeits. Deep Neural Networks (DNNs) are trained on large datasets, including real and fake videos and images. They study the characteristics that distinguish fakes from real content, such as skin textures, movement patterns and facial proportions. To protect the authenticity of content, digital signatures and hashing can be used to record and verify the originality and integrity of data. This can be used to track any changes or manipulations of media content. There are specialised software tools and platforms, such as DeepFaceLab, FaceForensics++, Reality Defender and others, that provide opportunities for analysing and detecting Deep Fake. They use a combination of the above methods to ensure accuracy and efficiency in detecting counterfeits. Together, these methods and technologies form a powerful toolkit to combat the problem of Deep Fake, providing protection against manipulation of media content and maintaining trust in digital data.

DISCUSSION

Cross-platform adaptation of algorithmic editing methods is a complex and multifaceted task that requires consideration of many aspects, from technical implementation to user experience. The main problem is the need to create algorithms that can run efficiently on different operating systems and hardware platforms without losing performance and data quality.

One of the key approaches to solving this problem is cross-compilation. This method can be used to generate executable code that can be executed on a variety of target platforms, providing versatility and independence from the source system. However, cross-compilation often faces incompatibility issues and requires significant resources to support different platforms. Similar conclusions were reached by M. Dempewolf (2020) in a study on the perception and experience of mobile developers when using cross-platform methods. As noted, the main reasons for choosing cross-platform solutions are cost reduction and accelerated development. The author emphasised the importance of ensuring compatibility and adequate testing when using cross-compilation, which is a key factor in successful cross-platform development.

Universal libraries and APIs are also significant in the cross-platform adaptation of algorithmic editing methods. These libraries and APIs provide a common programming interface that abstracts platform-specific details, making it much easier to develop and maintain algorithms. Despite some limitations, such as functionality and performance, they provide compatibility and integration with various platforms. The optimal use of universal libraries and APIs, combined with native platform-specific solutions, can provide the balance between versatility and performance needed for successful cross-platform adaptation. S. De (2021) also addressed the approach to designing a unified service API modelling, although for the semantic compatibility of inter-corporate automotive applications. The study results demonstrate that the use of a unified service API can significantly improve the interaction between different corporate systems, providing a higher degree of integration and compatibility.

The study identified several cross-platform mobile app development tools that have proven effective in adapting algorithmic editing methods. Flutter, developed by Google, supports iOS, Android and web applications, providing high performance using the Dart language and native user interfaces rendering. Flutter also provides a powerful set of development and debugging tools. Xamarin supports iOS, Android, and Windows, allowing developers to use C# and .NET to build cross-platform applications. The benefits of Xamarin include access to native APIs and integration with the .NET ecosystem, although performance may vary depending on the task. These tools were evaluated according to various criteria, including supported platforms, performance, versatility, available libraries and development tools. The choice of the right tool depends on the specific requirements of the project and the target platforms. Similar studies were conducted by M. Işıtan & M. Koklu (2020). The authors compared and evaluated various cross-platform mobile application development tools. This study demonstrated that cross-platform tools have unique strengths and weaknesses, and developers should carefully choose a tool depending on the requirements of their projects, accounting for performance, ease of use, support for native features, and community accessibility. Y. Kyelem *et al*. (2021) investigated a hybrid approach to developing a cross-platform mobile interface for Infrastructure as a Service (IaaS). As concluded, a hybrid approach to developing a cross-platform mobile interface for IaaS can significantly improve flexibility, reduce costs, and speed up the development process. However, for this approach to be successful, potential integration and interoperability challenges need to be considered.

This study determined that the application of AI in media is an important tool used for the production, processing, distribution and consumption of multimedia content. Key areas of AI use in media include task automation, improving media quality, adapting content to user preferences, creating new forms of content and optimising content management processes. AI can automate many routine and labour-intensive processes in the media industry. In the field of image processing, AI automatically improves image quality, removes noise and corrects lighting, which simplifies the process of preparing visual content. AI plays a key role in creating a personalised experience for users. AI-powered recommendation systems analyse user behaviour and preferences to offer relevant content, which increases audience engagement and improves user experience. K. Nader *et al*. (2024) drew similar conclusions but did so through a different approach. The scientists analysed how AI is represented in entertainment media and how these representations influence public perception and understanding of AI. J.V. Pavlik (2023) addressed the impact of using generative AI, such as ChatGPT, on the fields of journalism and media education. The results demonstrated the potential benefits and challenges of using generative models in journalism and media education. They can also help to develop guidelines for the effective and responsible use of such technologies, contributing to the development of more sustainable and ethical practices in these industries.

The study revealed how deep fakes are created and what technologies are used for this purpose. Deep fake is a technology based on the use of AI and ML that can be used to create fake videos or images by replacing the faces of some people with the faces of other people with a high degree of realism. The study determined that various technologies such as DNNs, Autoencoders, GANs and software platforms such as TensorFlow and PyTorch are used to create deep fakes. The study also highlighted that the ability to create such fake videos and images raises significant ethical and legal issues. Potential abuses include spreading disinformation, defamation and fraud. This underscores the need to develop new legislative measures and technologies to detect and prevent the use of deep fakes for malicious purposes, protect personal information and ensure security in the digital space. C.-C. Hsu *et al*. (2020) drew similar conclusions, however, they focused on detecting fake images and analysing threats related to deep fakes. The authors addressed the development and analysis of deep learning methods for detecting fake images created using GANs and other similar approaches.

The conducted study also identified which methods and neural network architectures are best suited to detecting different types of fake images. S. Negi *et al*. (2021) addressed and analysed the phenomenon of deep fakes – technologies that allow the creation of realistic fake images and videos using ML and computer vision. As such, the features and potential threats associated with the distribution of such content in the digital environment were identified.

In general, study results demonstrate how cross-platform algorithmic editing methods contribute to the creation of universal tools available for different operating systems and devices. This makes editing technologies more accessible to users with different preferences and needs. In addition, the study not only expands the technological horizons in the field of content editing but also helps to improve the user experience and develop new innovative solutions.

CONCLUSIONS

The study identified key technologies and technical features related to algorithmic video editing, the use of AI in media, and methods to improve the accuracy of working with large amounts of data. The issue of creating and detecting deep fake technologies was also considered. Cross-platform adaptation of algorithmic video editing methods allows algorithms to be effectively applied on different devices and platforms, ensuring the same high quality of processing and editing. AI in media is used to automate tasks, improve content quality, personalise user experience and optimise content management processes. This includes automatic image and video enhancement, user behaviour analysis to offer personalised content and other features. AI software is a key tool for a variety of media tasks, including content creation, editing, analysis and trend prediction.

The survey results highlight the need for further development and improvement of cross-platform adaptation technologies in the field of algorithmic editing. They also point to the current challenges faced by specialists in this field, including integration difficulties and the need for constant optimisation of algorithms. Deep Fake is a technology for generating fake images and videos using deep learning. It uses neural networks to create realistic fakes, which pose security and trust risks. Specialised software and technologies are being developed to detect and recognise Deep Fake, including methods of analysing deep textures and behavioural patterns, as well as the use of computer vision and ML technologies. Further research includes the development of universal libraries and frameworks that can simplify the integration of algorithmic editing across platforms, minimising the need for changes to the application source code. The study is limited to the selection of certain technologies and frameworks, which excludes other potentially effective solutions for cross-platform adaptation.

ACKNOWLEDGEMENTS

None.

CONFLICT OF INTEREST

None.

REFERENCES

[1] Ameen, S., & Mohammed, D. (2022). Developing cross-platform library using flutter. *European Journal of Engineering and Technology Research*, 7(2), 18-21. doi: 10.24018/ejeng.2022.7.2.2740.
[2] Bieda, I., & Panchenko, T. (2022). A systematic mapping study on artificial intelligence tools used in video editing. *International Journal of Computer Science and Network Security*, 22(3), 312-318. doi: 10.22937/IJCSNS.2022.22.3.40.
[3] Blanco, J.Z., & Lucrédio, D. (2021). A holistic approach for cross-platform software development. *Journal of Systems and Software*, 179, article number 110985. doi: 10.1016/j.jss.2021.110985.
[4] Casalicchio, E., & Iannucci, S. (2020). The state-of-the-art in container technologies: Application, orchestration and security. *Concurrency and Computation: Practice and Experience*, 32(17), article number e5668. doi: 10.1002/cpe.5668.
[5] De, S. (2021). *Design approach to unified service API modeling for semantic interoperability of cross-enterprise vehicle applications*. Plzeň: University of West Bohemia.
[6] de-Lima-Santos, M.-F., & Ceron, W. (2022). Artificial intelligence in news media: Current perceptions and future outlook. *Journalism and Media*, 3(1), 13-26. doi: 10.3390/journalmedia3010002.
[7] Dempewolf, M. (2020). *Perceptions of mobile developers adopting cross-platform methods: A generic qualitative inquiry*. (Doctoral dissertation, Capella University, Minneapolis, USA).
[8] Dey, N. (2021). *Cross-platform development with Qt 6 and Modern C++: Design and build applications with modern graphical user interfaces without worrying about platform dependency*. Birmingham: Packt Publishing.
[9] Eugeni, R., & Pisters, P. (2020). The artificial intelligence of a machine: Moving images in the age of algorithms. *European Journal of Media Studies*, 1, 91-100. doi: 10.25969/mediarep/14325.
[10] European Commission. (2021). *Ethics and data protection*. Retrieved from https://ec.europa.eu/info/funding-tenders/opportunities/docs/2021-2027/horizon/guidance/ethics-and-data-protection_he_en.pdf.
[11] Hsu, C.-C., Zhuang, Y.-X., & Lee, C.-Y. (2020). Deep fake image detection based on pairwise learning. *Applied Sciences*, 10(1), article number 370. doi: 10.3390/app10010370.
[12] Işıtan, M., & Koklu, M. (2020). Comparison and evaluation of cross platform mobile application development tools. *International Journal of Applied Mathematics, Electronics and Computers*, 8(4), 273-281. doi: 10.18100/ijamec.832673.
[13] Kishore, K., Khare, S., Uniyal, V., & Verma, S. (2022). Performance and stability comparison of react and flutter: Cross-platform application development. In *2022 international conference on cyber resilience (ICCR)* (pp. 1-4). Dubai: Institute of Electrical and Electronics Engineers. doi: 10.1109/ICCR56254.2022.9996039.
[14] Kwok, A.O.J., & Koh, S.G.M. (2021). Deepfake: A social construction of technology perspective. *Current Issues in Tourism*, 24(13), 1798-1802. doi: 10.1080/13683500.2020.1738357.
[15] Kyelem, Y., Kabore, K.K., & Bassole, D. (2021). Hybrid approach to cross-platform mobile interface development for IAAS. In S. Shakya, R. Bestak, R. Palanisamy & K.A. Kamel (Eds.), *Mobile computing and sustainable informatics* (pp. 225-238). Singapore: Springer. doi: 10.1007/978-981-16-1866-6_16.
[16] Li, C., Zhu, J., Bi, L., Zhang, W., & Liu, Y. (2022). A low-light image enhancement method with brightness balance and detail preservation. *PLoS ONE*, 17(5), article number e0262478. doi: 10.1371/journal.pone.0262478.
[17] Ma, Y., Yang, Z., Chiu, B., Zhang, Y., Jiang, J., Du, B., & Fan, H. (2022). Supporting cross-platform real-time collaborative programming: Architecture, techniques, and prototype system. In H. Gao & X. Wang (Eds.), *Collaborative computing: Networking, applications and worksharing* (pp. 124-143). Cham: Springer. doi: 10.1007/978-3-030-92638-0_8.
[18] Menegassi, A.A., & Endo, A.T. (2020). Automated tests for cross-platform mobile apps in multiple configurations. *IET Software*, 14(1), 27-38. doi: 10.1049/iet-sen.2018.5445.
[19] Nader, K., Toprac, P., Scott, S., & Baker, S. (2024). Public understanding of artificial intelligence through entertainment media. *AI & SOCIETY*, 39(2), 713-726. doi: 10.1007/s00146-022-01427-w.
[20] Negi, S., Jayachandran, M., & Upadhyay, S. (2021). Deep fake: An understanding of fake images and videos. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 7(3), 183-189. doi: 10.32628/CSEIT217334.
[21] Nguyen, H. (2022). *Programming language interoperability in cross-platform software development*. (Master's thesis, Aalto University, Espoo, Finland).

[22] Novac, O.-C., Chirodea, M.-C., Novac, C.-M., Bizon, N., Oproescu, M., Stan, O.P., & Gordan, C.E. (2022). Analysis of the application efficiency of TensorFlow and PyTorch in convolutional neural network. *Sensors*, 22(22), article number 8872. doi: 10.3390/s22228872.

[23] Pavlik, J.V. (2023). Collaborating with ChatGPT: Considering the implications of generative artificial intelligence for journalism and media education. *Journalism & Mass Communication Educator*, 78(1), 84-93. doi: 10.1177/10776958221149577.

[24] Revi, K.R., Vidya, K.R., & Wilscy, M. (2021). Detection of deepfake images created using generative adversarial networks: A review. In M. Palesi, L. Trajkovic, J. Jayakumari & J. Jose (Eds.), *2nd international conference on networks and advances in computational technologies* (pp. 25-35). Cham: Springer. doi: 10.1007/978-3-030-49500-8_3.

[25] Rieger, C., & Majchrzak, T.A. (2019). Towards the definitive evaluation framework for cross-platform app development approaches. *Journal of Systems and Software*, 153, 175-199. doi: 10.1016/j.jss.2019.04.001.

[26] Sajbidor, M., Vesely, P., & Krajewski, M. (2023). Creating cross-platform application in Java and C++. In N. Kryvinska, M. Greguš & S. Fedushko (Eds.), *Developments in information and knowledge management systems for business applications* (pp. 495-540). Cham: Springer. doi: 10.1007/978-3-031-27506-7_19.

[27] Toasa, R.M., Egas, P.F.B., Recalde, H., & Saltos, M.G. (2023). Mobile development with Xamarin: Brief literature, visualizations and important issues. In Á. Rocha, C. Ferrás & W. Ibarra (Eds.), *Information technology and systems* (pp. 299-307). Cham: Springer. doi: 10.1007/978-3-031-33261-6_26.

[28] Vassallo, K., Garg, L., Prakash, V., & Ramesh, K. (2019). Contemporary technologies and methods for cross-platform application development. *Journal of Computational and Theoretical Nanoscience*, 16(9), 3854-3859. doi: 10.1166/jctn.2019.8261.

[29] Xu, M., Chen, D., & Zhou, G. (2020). Real-time face recognition based on Dlib. In *Innovative computing* (pp. 1451-1459). Singapore: Springer. doi: 10.1007/978-981-15-5959-4_177.

[30] Zhang, Y. (2021). *Cross-platform methods in computer graphics that boost experimental film making*. (Master's thesis, Rochester Institute of Technology Rochester Institute of Technology, Rochester, USA).

Кросплатформна адаптація алгоритмічних методів редагування

Сян Лі
Аспірант
Чіангмайський університет
50200, дорога Хуай Каев, 239, м. Чіангмай, Таїланд
https://orcid.org/0009-0000-7073-3646
Тампрасірт Анукуль
Доктор філософії
Чіангмайський університет
50200, дорога Хуай Каев, 239, м. Чіангмай, Таїланд
https://orcid.org/0009-0005-5424-1170
Фанлі Ін
Доктор філософії
Чіангмайський університет
50200, дорога Хуай Каев, 239, м. Чіангмай, Таїланд
https://orcid.org/0000-0001-8390-3229

Анотація. Актуальність дослідження зумовлена стрімким розвитком технологій та зростаючою потребою в ефективній обробці даних на різних платформах. Метою дослідження була розробка методів, які б дозволили використовувати алгоритми редагування даних на різних операційних системах та апаратних платформах. Було розроблено методологію дослідження технологій кросплатформної адаптації, включаючи крос-компіляцію, віртуалізацію, використання універсальних бібліотек та інтерфейсу прикладного програмування, а також методи тестування та оптимізації продуктивності алгоритмів. У дослідженні були розглянуті різні підходи до реалізації крос-платформної сумісності, включаючи використання крос-компіляції, віртуалізації та контейнеризації. Основними технічними проблемами є управління ресурсами, оптимізація продуктивності та забезпечення сумісності з апаратним забезпеченням різних платформ. Принципи включають вибір найбільш підходящої технології для вирішення поставленого завдання, врахування вимог до продуктивності та безпеки, а також забезпечення ефективної інтеграції існуючих систем та інфраструктури. Робочі процеси зосереджені на створенні модульних і розширюваних рішень, які можуть легко адаптуватися до змін у технологічному середовищі та вимог користувачів. У контексті цього дослідження програмне забезпечення зі штучним інтелектом відіграє ключову роль у підвищенні ефективності та точності обробки даних на різних платформах. Результати показали, що програмне забезпечення зі штучним інтелектом може автоматизувати різні етапи процесу редагування відео та аудіо. Штучний інтелект використовується для аналізу великих обсягів даних контенту, таких як відеофайли, зображення та аудіозаписи. Дослідження показало, що штучний інтелект стає все більш актуальним у різних аспектах кіновиробництва. Штучний інтелект може аналізувати сценарії, прогнозувати їхній потенційний успіх і пропонувати покращення, використовуючи дані попередніх фільмів та їхній комерційний успіх

Ключові слова: оптимізація; програмне забезпечення; ефективність; сумісність; інтерфейс

BULLETIN of Cherkasy State Technological University

Journal homepage: https://bulletin-chstu.com.ua/en

Vol. 29 No. 2. 2024

UDC 004.7:658.512
DOI: 10.62660/bcstu/2.2024.57

Strategies for implementing or strengthening the DevOps approach in organizations: Analysis and examples

Bohdan Fedoryshyn*

Master
Lviv Polytechnic National University
79000, 12 Stepan Bandera Str., Lviv, Ukraine
https://orcid.org/0009-0005-3779-0186

Abstract. The aim of the study was to analyse the implementation of DevOps in organizations, in particular, to assess the impact of process automation, CI/CD, monitoring and orchestration of microservices on the efficiency of development and management. A methodology was created that allows organizations to effectively implement and enhance the DevOps approach, achieving high results in software development and management. The study looked at cultural change and training strategies, including building a culture of collaboration between teams and developing skills in modern DevOps tools. Process automation, including test automation and integration and deployment, plays an important role in improving code quality and reducing development time. Infrastructure as code allows managing and automating infrastructure configuration, enabling rapid scaling of environments. The work has shown that performance monitoring and feedback are critical to detecting problems early and continuously improving the product. Microservices orchestration, in particular, with Kubernetes, automates the deployment and scaling of containerized applications, which is critical for modern architectures with high performance and availability requirements. Case studies of companies such as Netflix, Spotify, and Airbnb demonstrate the successful application of DevOps practices and technologies to achieve high availability, scalability, and reliability of infrastructures. The study results confirm that the implementation of DevOps leads to a significant increase in development efficiency, software quality, and optimization of infrastructure management costs. In addition, the introduction of cultural changes and increased collaboration between development and operations teams contributes to greater flexibility and speed of response to changing market conditions

Keywords: microservices orchestration; tools; test automation; CI/CD processes; development efficiency

Article's History: Received: 10.12.2023; Revised: 05.04.2024; Accepted: 27.05.2024.

INTRODUCTION

In today's digital environment, organizations are increasingly adopting the DevOps approach to improve the efficiency of software development and operations management. DevOps, which combines development and operations practices, aims to shorten the development cycle, improve software quality and simplify the introduction of new features. This approach allows organizations to respond quickly to market changes and ensure high performance and reliability of their systems. DevOps provides integration and orchestration of microservices, process automation, and continuous monitoring. In particular, the automation of testing and build, integration and deployment (CI/CD) processes allows for the rapid and efficient introduction of new features and updates. Infrastructure as a Code (IaC) simplifies infrastructure management by enabling rapid scaling and repeatable environments. Continuous performance monitoring and well-established feedback processes help to identify and resolve issues early on, as well as continuously improve the product.

This study focuses on identifying the most effective strategies for implementing and strengthening the

Suggested Citation:
Fedoryshyn, B. (2024). Strategies for implementing or strengthening the DevOps approach in organizations: Analysis and examples. *Bulletin of Cherkasy State Technological University*, 29(2), 57-69. doi: 10.62660/bcstu/2.2024.57.

*Corresponding author

DevOps approach in organizations, taking into account numerous challenges such as cultural change, training and skills development, process automation, microservice orchestration, and monitoring and feedback. Implementing DevOps requires significant changes in the organization's culture, a shift to a more open and collaborative approach between development and operations teams, and the adoption of new tools and methodologies such as CI/CD, containerization, test automation, and IaC. Automating processes, such as building, testing, and deploying code, is a resource-intensive and complex process that requires significant investment in new technologies and tools. Managing microservices in complex architectures requires the use of orchestration systems such as Kubernetes, which also adds to the complexity. Monitoring performance and receiving feedback is critical to detecting and fixing problems at an early stage. Analysing examples of DevOps implementations at companies such as Netflix, Spotify, and Airbnb helps to highlight best practices and methodologies that other organizations can apply to achieve success.

The main challenge in this area is the difficulty of integrating DevOps into traditional organizations with established processes and structures, as well as the lack of knowledge and skills among staff. S. Schünemann (2023) investigated the use of automated test pipelines in continuous software development. As a result, he described methods and tools for the effective implementation of automated tests in the continuous development process. S. Kornitskyi (2024) studied the processes of implementing a service quality management system in accordance with ISO/IEC 20000 and described the benefits of standardization and its impact on the quality of services in technology companies. The impact of DevOps on organizational culture, metrics effectiveness, tool integration, staff training, innovation processes, security, and adaptation for different sizes of organizations should be further studied. Another existing challenge is the difficulty of integrating new methodologies into existing processes and structures of the organization, as well as the lack of knowledge and skills among staff to effectively apply DevOps practices.

N. Govil *et al*. (2020) describe methods and approaches for implementing DevOps in web development, including tools and practices that ensure the efficiency of web application development and maintenance. M. Muñoz & M.N. Rodríguez (2021) identified strategic approaches to DevOps implementation and provided examples of successful application of these strategies in organizations. Gaps that require further study include the impact of DevOps process automation on the quality of the final product and user experience. The problem in the field under study is the difficulty of transitioning to new methodologies and cultural changes, as well as the lack of readiness of organizations to adapt DevOps practices. S. Rafi *et al*. (2020) developed a model to assess the readiness of organizations to implement DevOps, which included an assessment of infrastructure, processes, and cultural aspects.

M.S. Khan *et al*. (2022) identified the main challenges organizations face when integrating a DevOps culture, including resistance to change and communication issues between teams. It is necessary to consider how to effectively manage cultural change and overcome resistance when implementing DevOps. A common problem is the difficulty of integrating new approaches such as DevSecOps and assessing the maturity of DevOps processes in organizations. M.A. Akbar *et al*. (2022) described a decision-making framework for the successful implementation of DevSecOps in software development organizations, which included methods for integrating security into DevOps processes. J. Radstaak (2019) developed a DevOps maturity model to assess the level of DevOps adoption in organizations, allowing organizations to measure and improve their DevOps processes. It is important to explore how to successfully integrate security into DevOps processes, including DevSecOps, as well as the management decisions required to do so.

The aim of the study was to analyse the implementation of the DevOps approach in companies, in particular, to study the impact of process automation, continuous integration and deployment (CI/CD), monitoring and orchestration of microservices on the efficiency of development and management. To achieve this goal, the following tasks were identified:

1. Analysing the cultural changes required for successful DevOps implementation, including the importance of building a culture of collaboration between development and operations teams.

2. Studying modern DevOps tools and practices, such as CI/CD, containerization and test automation, with a focus on training and development of employee skills.

MATERIALS AND METHODS

The study of the theoretical foundations of DevOps, its components, such as CI/CD, test automation, containerization, and infrastructure as code, allowed to form a knowledge base for further analysis. Successful case studies from real organizations were analysed to gain a practical understanding of DevOps implementation. Examples of companies such as Netflix, Spotify, and Airbnb that have implemented DevOps and achieved significant results in improving development processes and operations are considered. The necessary cultural changes for the successful implementation of DevOps were studied. The importance of creating a culture of collaboration between development and operations teams is assessed, which is critical to ensure effective information sharing and joint work on projects. An overview of modern DevOps tools and practices, such as CI/CD, containerization and test automation, is provided. The importance of employee training in these areas is emphasized.

The study included a detailed examination of the use of automated tests and continuous integration and deployment (CI/CD) systems, which are critical to ensuring code quality and automating development and deployment processes. The methodology involves developing test scenarios that cover various aspects of the application's functionality and performance, integrating tests with CI/CD processes to execute them every time a change is made to the code, and incorporating different types of testing, such as unit tests, integration tests, functional tests, and regression tests. IaC management tools such as Terraform, Ansible, and AWS CloudFormation are explored. This allows exploring how automating configurations and infrastructure management contributes to the rapid scalability and repeatability of environments. The overview includes an analysis of the functionality of each tool, as well as use cases for improving scalability and infrastructure management.

The methods of continuous performance monitoring and feedback are reviewed. This confirms the importance of these processes for identifying and fixing problems at an early stage, as well as for continuous product improvement. The study analyses in detail the role of microservices orchestration in modern DevOps practices, with a particular focus on managing deployed services using Kubernetes. This is an important aspect for ensuring the scalability and reliability of microservice architectures. The work also focuses on evaluating and comparing popular tools and platforms used to implement the DevOps approach, including Jenkins, Kubernetes, Docker, and IaC management systems. As part of the preliminary research, a comprehensive review of scientific publications, technical articles, reports, and documentation related to these tools was conducted (Smart, 2011; Morris, 2020). The review included a study of architectural solutions, functionalities, limitations, and features of Jenkins, Kubernetes, Docker, and IaC in the context of DevOps, which allowed to assess in detail their impact on software development and deployment processes. The results of DevOps implementation, including increased development speed, improved software quality, reduced infrastructure management costs, and the role of Kubernetes in ensuring high scalability and reliability, are evaluated.

RESULTS

Creating a culture of collaboration is one of the main strategies for successful DevOps implementation in organizations. It is critical for ensuring effective interaction between development and operations teams, reducing barriers between these groups and creating a unified working environment. The process of creating such a culture begins with the recognition of the need for constant information sharing and joint work on projects. In traditional approaches to software development, development and operations teams often work in isolation, which can lead to delays in implementing changes and unforeseen problems in the course of operation. DevOps seeks to break down the barriers between development and operations teams by integrating both teams at all stages of software development and deployment. To achieve this goal, DevOps implements several key initiatives:

1. Changing the way of thinking and approaches to work. DevOps focuses on changing the way we think and work, encouraging teams to work as a unit with a common goal of delivering new features and fixing bugs quickly and efficiently. This includes changes in communication, process organization and project management.

2. Regular meetings and information exchange. DevOps ensures that information is shared regularly between teams through joint meetings, huddles, and shared communication platforms. This allows all project participants to be aware of changes, problems, and needs that arise during development and operation.

3. Implementation of common tools. The use of common tools for project management, monitoring and reporting is central to DevOps. This helps to improve process transparency and provides a single source of truth for all project participants.

4. Cross-functional training. DevOps involves upskilling employees through training in new tools and methodologies used in both development and operations. This helps reduce dependence on specialized knowledge and promotes greater flexibility in work.

5. Management support. The leadership of organizations plays an important role in DevOps by actively supporting initiatives and creating the conditions for successful change. This includes financial and moral support for teams working on DevOps implementation.

Fostering a culture of collaboration is critical to the success of DevOps adoption, as it allows organizations to respond to change more effectively, introduce new features faster, and improve software quality (Luz *et al.*, 2018). When teams work together, their ability to innovate and solve problems increases significantly, leading to the achievement of common goals and increased competitiveness of the organization. Training and skills development are critical to the successful implementation of DevOps in organizations. Employees need to acquire in-depth knowledge and skills in modern DevOps tools and practices, such as CI/CD, containerization, and test automation, to effectively implement these practices and ensure their successful operation. The learning and skill development process includes an introduction to CI/CD tools such as Jenkins, GitLab CI/CD, CircleCI, or Travis CI, which automate the processes of building, testing, and deploying software. Mastery of these tools helps to reduce the time of development and release of new versions of products, as well as ensure their high quality. In addition, knowledge of containerization and container management principles is critical for the development and implementation of microservices. Employees need to be able to work with containerization tools such as Docker and understand how to manage containers using orchestration systems such as Kubernetes. Containerization simplifies deployment processes, provides isolation of environments, and makes it easier to scale applications.

Test automation is another crucial aspect that helps ensure software quality at all stages of development. Employees should be able to use automated testing tools such as Selenium, JUnit, or TestNG to create and execute test scripts. This helps to identify errors faster, reduce manual testing, and improve the overall quality of the product. In addition to technical skills, it is relevant to train employees in DevOps practices, which include building effective development and deployment processes, configuration management, and performance monitoring and analysis. This requires knowledge of DevOps best practices and standards, such as IaC and DevOps Culture. DevOps technologies and practices are constantly evolving, so it's important to ensure that employees are constantly trained and certified. This may include participation in trainings, webinars, and courses, as well as obtaining professional certifications that confirm knowledge and skills in the field of DevOps.

Implementing and maintaining training programmes for employees helps organizations ensure a high level of skill and readiness to work with modern DevOps tools and practices. This helps to improve the efficiency of development and deployment processes, increase software quality and reduce time to market. Active investment in employee skill development is critical to achieving success in DevOps implementation and provides a competitive advantage for the organization. Test automation is a critical component in DevOps implementation, as it allows effectively ensuring high-quality software at all stages of development. Implementing automated tests helps reduce bugs, shorten development time, and improve overall software stability and reliability.

The test automation process begins with the creation of test scripts that define which aspects of the software should be tested. These can be unit tests to check individual code components, integration tests to check the interaction between components, or system tests to check the operation of the entire application. After defining the test scenarios, the next step is to use automation tools. Tools such as Selenium for web applications, JUnit or TestNG for Java applications, or NUnit for NET applications help automate test execution and reduce manual work. The integration of automated tests with continuous integration (CI) and continuous deployment (CD) processes is an essential aspect. This allows running tests automatically every time it is necessary to make a change to the code, which helps to detect bugs at an early stage and ensures product stability. CI/CD tools such as Jenkins, GitLab CI/CD, or CircleCI are used to automate this process and ensure continuous testing (Singh, 2022).

After running automated tests, it is critical to collect and analyse the test results. This allows identifying problems, getting information about application performance, and defining areas for improvement. Metrics and reporting tools such as SonarQube or Allure help in visualizing test results and tracking trends. Regularly updating test scenarios is essential as software is constantly evolving and changing. This ensures that tests remain relevant and effective by adapting them to new features and requirements. Test automation also reduces the need for manual testing, which can be time-consuming and error-prone. Automated tests are faster and less likely to make human errors, which increases the accuracy and efficiency of testing. Implementing automated tests helps to identify defects faster, reduce testing time, and improve overall product quality, which is key to successful DevOps implementation and competitive advantage.

Implementation of continuous integration and deployment (CI/CD) systems is a key element of successful DevOps implementation, as these systems automate the processes of building, testing, and deploying code, which significantly increases development efficiency and ensures high-quality software (Singh & Mansotra, 2021). Continuous Integration (CI) involves automatically building and testing code whenever changes are made to the repository. This allows detecting defects in the early stages of development and reducing the likelihood of errors in the finished product. CI systems, such as Jenkins, GitLab CI/CD, CircleCI, or Travis CI, automate these processes by ensuring that tests and builds are performed regularly and automatically at each commit. This helps to identify and fix bugs faster, facilitates the integration of new features, and reduces development delays.

Continuous Deployment (CD) extends the CI concept by automating the process of deploying code to a test or production environment. This ensures fast and reliable deployment of new software releases, reducing the time from development to deployment. CD systems automatically check the code, deploy it to test environments, and, if everything goes well, to the production environment. This allows quickly releasing new versions of the product and maintaining a high level of stability and quality (Fig. 1).

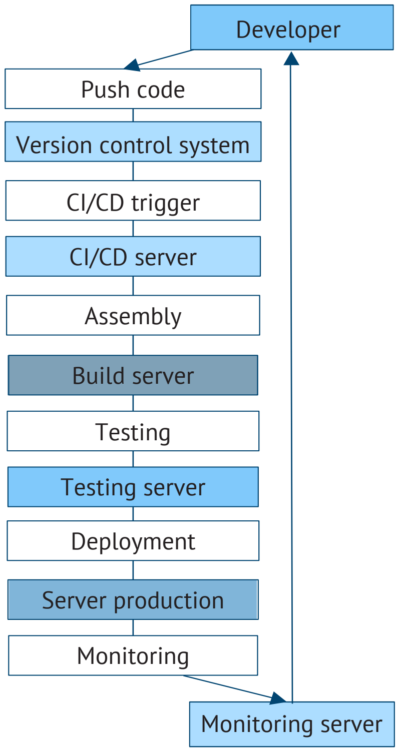


Figure 1. Diagram of interaction between CI/CD components
Source: compiled by the author based on O. Kravchuk (2023)

IaC is an important component in the implementation of CI/CD. IaC automates infrastructure management by using configuration files to describe and manage environment components such as servers, networks, and databases. This provides greater flexibility and scalability, and ensures consistent environments for development, testing, and performance. By using IaC tools such as Terraform, Ansible, or AWS CloudFormation, organizations can automatically build and manage infrastructure, which significantly reduces the likelihood of human error and increases the speed and accuracy of deployment. IaC also makes it easy to track changes to infrastructure, manage versions, and recover systems after a disaster. The integration of CI/CD and IaC provides a continuous development and deployment process that automates key stages of the software lifecycle, reduces time to market, improves product quality, and provides flexibility in scaling and infrastructure management. This allows organizations to respond quickly to changes, reduce risks and achieve high performance in a competitive environment.

Infrastructure as code is a relevant practice in modern DevOps strategies that allows managing and automating the configuration and management of infrastructure through code (Ljunggren, 2023). This significantly increases the efficiency and flexibility of infrastructure deployment processes and provides a number of benefits. Infrastructure as code uses configuration files or scripts written in specialized languages or formats, such as YAML Ain't Markup Language™ (YAML), JavaScript Object Notation (JSON), and HashiCorp Configuration Language (HCL), to describe and manage infrastructure resources. These files contain all the instructions needed to create, configure, and manage server instances, networks, storage, and other infrastructure components.

One of the main benefits of IaC is the ability to scale infrastructure quickly. With IaC, organizations can quickly and efficiently scale resources up or down to meet business needs without having to manually configure each component of the infrastructure. This is especially important in the face of dynamic and changing workloads, where the speed of scaling can affect the performance and availability of services. Infrastructure as code also ensures repeatability and reproducibility of environments. The code that describes the infrastructure can be saved in version control systems such as Git. This makes it easy to reproduce and reuse infrastructure configurations in different environments, such as development, testing, and production. All changes to the infrastructure are documented and verifiable, making it easier to manage configurations and reduce the likelihood of errors.

Automating infrastructure management is also a key benefit of IaC. Instead of manually configuring and administering resources, organizations can automate the creation, configuration and updating of infrastructure. This helps to reduce administration costs, reduce the likelihood of human error, and ensure consistency across all environments. IaC supports the use of infrastructure automation and management tools such as Terraform, Ansible, Puppet, or Chef. These tools allow describing one's infrastructure as code and automating its deployment and management, which simplifies processes and provides high efficiency and control over one's infrastructure. Performance monitoring is an essential component of successful DevOps implementation (Azad, 2022). It allows constantly monitoring the state of the system, identifying and fixing problems at an early stage. Ongoing monitoring ensures a high level of software availability and reliability, which is critical to providing a positive user experience and supporting business processes. The main purpose of performance monitoring is to collect and analyse metrics, logs, and bug tracking. Metrics allow tracking key system performance indicators, such as response time, CPU utilization, memory usage, network speed, and other important parameters. Collecting and analysing this data allows detecting anomalies, overloads, or other issues that may affect system performance in a timely manner.

Metrics collection tools provide automated collection and storage of system health data. They can provide real-time monitoring and the ability to view graphs and charts illustrating system performance. Tools such as Prometheus, Grafana, Datadog, or New Relic allow collecting data from a variety of sources, including services, applications, and infrastructure. Logs (log files) are another major part of monitoring, providing detailed information about events occurring in the system. Logs can contain records of errors, warnings, informational messages, and details about the execution of requests. Analysing logs helps to identify problems, track their causes, and understand how the system responds to different conditions. Tools for working with logs, such as ELK Stack (Elasticsearch, Logstash, Kibana) or Splunk, allow centralizing, analysing and visualizing logs to identify and resolve problems faster.

Error tracking involves monitoring and logging errors that occur in a system. Error tracking tools such as Sentry, Rollbar, or Raygun help to automatically detect and report errors, including their nature, frequency, and impact on the system. This allows developers to respond quickly to issues by providing detailed reports and context for resolution. By continuously monitoring performance, organizations can detect problems early, reduce time to resolution, and improve overall system stability and efficiency. Monitoring also provides crucial feedback on the state of the system, enabling informed decisions to be made about further development and optimization of the software and infrastructure.

Feedback is a critical element for continuous product improvement and ensuring that a product meets the needs of users. Establishing user feedback processes and automated systems allows organizations to make improvements based on real-world data and user experience. The first aspect is collecting feedback from users. It is relevant to establish effective channels for collecting feedback, such as online surveys, feedback forms, user forums, or functionality to send feedback directly from the application or website interface. Providing a convenient and easy way for users to provide feedback allows getting a variety of perspectives and identify potential problems or areas for improvement. The second aspect is the use of data analytics. Analysing feedback data helps to identify trends, common problems, and user preferences. Data analytics tools, such as Google Analytics, Mixpanel, or Amplitude, allow collecting and processing data on user interaction with the product (Mykhalchenko & Tytarenko, 2023). This includes information about the frequency of use, session duration, failure points, and other metrics that help to understand how the product is used and where there may be problems. The third aspect is integration with automated systems. Automated systems can provide an additional layer of feedback through performance, log and error monitoring. Performance monitoring tools such as Datadog or New Relic can provide information about anomalies or issues that users may not be able to detect directly. Integrating these systems can automatically generate reports and alerts about potential problems, making it easier to identify and fix errors. The fourth aspect is continuous product improvement. Feedback from users and automated systems should be used to regularly update and improve the product. This may include releasing new versions, fixing identified defects, optimizing functionality, and improving the interface. A continuous improvement cycle allows adapting the product to the changing needs and preferences of users, ensuring its competitiveness in the market. The last but not the least aspect is interaction with development teams. Establishing a feedback process also includes effective communication between development and support teams. Feedback should be systematically passed on to developers for analysis and implementation of changes. Regular meetings where feedback is discussed, and next steps are planned foster better interaction between teams and ensure focus on improving the product. Successfully establishing feedback processes allows organizations to adapt to user and market needs, ensuring continuous improvement in product quality and user satisfaction. It also helps build trust and loyalty among users, which is critical to the long-term success of a product.

Microservices orchestration is an important part of the modern DevOps approach, as it allows for efficient management of deployed services in a complex architecture (Waseem *et al.*, 2020). The growing popularity of microservice architecture, which breaks down large monolithic applications into small, independent services, has led to the need for tools that help coordinate and manage these services. Microservice orchestration automates the deployment, scaling, and management of containers, which increases the flexibility and efficiency of software development and operation. One of the key tools for orchestrating microservices is Kubernetes, which provides powerful capabilities for the automated management of containerized applications. Kubernetes allows automatically deploying, scaling, and managing containers, providing load balancing, disaster recovery, and simplified configuration management. Thanks to its powerful features, Kubernetes is becoming the de facto standard for microservice orchestration in many organizations. Microservice orchestration includes several key aspects. Firstly, it is the automation of deployment. With the help of orchestration tools, it is possible to automatically deploy new versions of services in different environments, from test to production, which ensures fast and secure implementation of changes. This reduces human error and speeds up the deployment process, which is critical in today's dynamic environments. The second crucial aspect is scaling. Microservice orchestrators allow automatically scaling services depending on the load. For example, Kubernetes can automatically add or remove container instances based on current resource requirements, which allows for efficiently using available resources and ensuring high system performance even during peak loads. Configuration management is another critical aspect of microservice orchestration.

Orchestration tools allow centrally managing the configuration of services, which simplifies the upgrade process and reduces the likelihood of configuration errors. It also allows quickly implementing changes and adapting services to new requirements.

In addition, microservice orchestration provides disaster recovery. Orchestration tools such as Kubernetes can automatically restart failed containers and move services to other nodes in the event of hardware failures, ensuring high reliability and availability of the system. Monitoring and logging are also important aspects of microservice orchestration. Orchestration tools are often integrated with monitoring and logging systems such as Prometheus and ELK Stack, which allows for centralized collection and analysis of data on the status and performance of services. This helps to identify problems in time and make informed decisions to resolve them. Kubernetes is a powerful container orchestration system designed to automate the deployment, scaling, and management of containerized applications (Zhou *et al.*, 2021). It provides centralized management of large container clusters, making it an indispensable tool for modern IT infrastructures. Kubernetes has a wide range of functionalities that provide high availability, automatic disaster recovery, load balancing, and automatic container scaling. High availability is achieved by automatically distributing the load between different nodes, which ensures the uninterrupted operation of applications even in the event of a failure of individual components. Automatic disaster recovery includes automatic restart of containers in the event of a malfunction, as well as the ability to move containers to other nodes in the event of hardware problems. Load balancing ensures an even distribution of requests between containers, which increases system performance and stability. Automatic scaling allows dynamically adding or removing containers depending on the current resource requirements, which ensures efficient use of resources and high system performance.

Kubernetes integrates with CI/CD processes to automate the deployment of new versions of applications, providing version and environment control. Integration with CI/CD tools such as Jenkins, GitLab CI/CD, CircleCI, or Travis CI allows automatically deploying code updates to different environments, which speeds up the development and deployment process. Kubernetes also supports the IaC concept, which allows describing infrastructure configurations as code and automatically applying them across different environments. This ensures high repeatability and consistency of configurations, reduces the likelihood of errors and simplifies infrastructure management. Kubernetes also supports high availability configurations, which allows applications to remain available even in the event of a node failure. In addition, Kubernetes fits well with the microservices architectural style. With Kubernetes, each microservice can be deployed as a separate container in a module, which enables independent scaling, management, and rapid development of microservice-based applications (Fig. 2).

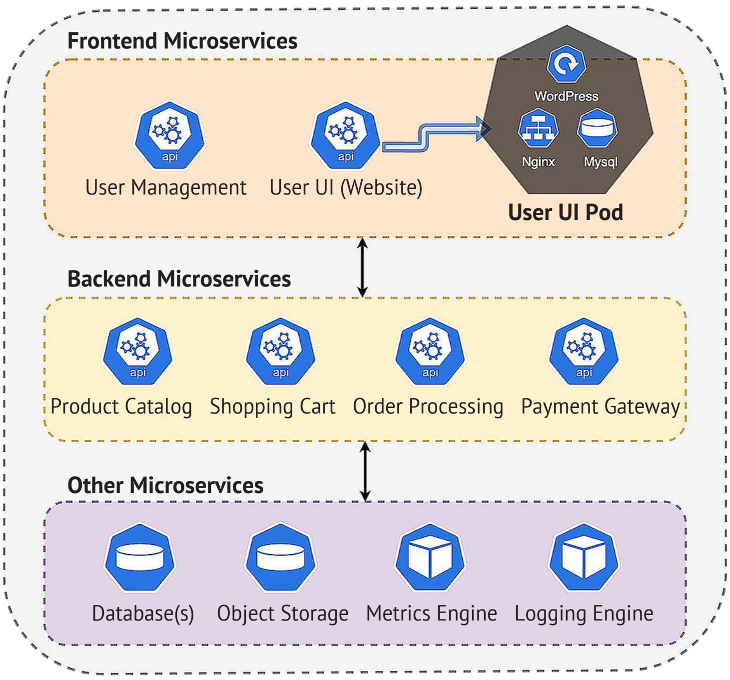


Figure 2. Microservice architecture in Kubernetes

Source: Yu. Vatsyk (2024)

Implementing DevOps is a complex process that requires changes in an organization's culture, the introduction of new tools and practices, and continuous monitoring and improvement of processes. However, the efforts invested in this transformation can significantly increase an organization's efficiency and competitiveness in the market. Netflix was one of the first large companies to implement Kubernetes to manage its microservices. Orchestrating microservices with Kubernetes allows Netflix to automate the deployment, scaling, and management of applications running in containers. This is critical for a company that serves more than 200 million users worldwide. Thanks to Kubernetes, Netflix can dynamically scale its services depending on the load, for example, when releasing new episodes of popular shows when the number of viewers increases dramatically. Kubernetes also allows for automated deployment of new versions of applications, minimizing the risks associated with human error and ensuring high service reliability.

Netflix has implemented continuous integration (CI) to ensure that changes to the code created by multiple development teams are integrated quickly and reliably. CI systems automatically test new changes, allowing potential problems to be identified early in the development process. This approach helps to minimize the time between writing code and deploying it to the production environment. Continuous deployment (CD) allows Netflix to automate the process of releasing new versions of applications, which significantly reduces the time to market for new features and updates. It also provides the ability to respond quickly to user feedback and quickly fix bugs or add new features. Netflix pays considerable attention to monitoring the performance of its services and collecting feedback from users. The company uses its own Atlas platform to collect performance metrics, which allows it to accurately assess the status of each service in real time. This includes monitoring indicators such as response time, number of errors, and other critical parameters that affect the user experience. Netflix has also developed Eureka, a service management system that allows managing the availability and scaling of microservices. Monitoring tools allow quickly identifying and fixing problems, ensuring uninterrupted service operation even during peak periods.

One of the key aspects of successful DevOps implementation at Netflix is cultural and organizational change. The company is creating cross-functional teams that bring together developers and operations specialists. This approach helps to improve communication and collaboration between different teams, which is critical for quick and effective implementation of changes. In addition, Netflix gives its teams a lot of autonomy in decision-making, which allows them to quickly adapt to changes and innovate without bureaucratic delays. This approach allows the company to remain a leader in the streaming services market, ensuring a high speed of development of new features and stability of work. In this way, Netflix demonstrates the effective use of DevOps tools and practices to manage complex infrastructure, which allows it to maintain a high quality of user experience and remain competitive in the market.

Spotify uses Kubernetes as its primary microservices orchestration tool, enabling it to efficiently manage the thousands of containers that power the many features on the platform. Spotify's microservice architecture separates the various functional components of the platform into separate services that can be independently deployed, scaled and maintained. This gives Spotify the ability to quickly adapt to changes in load, for example, during the release of new albums or significant events in the music world, when the number of active users can increase dramatically. Using Kubernetes also allows Spotify to automate the management of its microservices, ensuring that services are reliable and highly available even in the event of system failures.

Spotify actively integrates continuous integration (CI) and continuous deployment (CD) practices, which are key elements of DevOps at the company. CI processes automate the testing and integration of new code changes, which ensures that potential problems are quickly identified and resolved at the earliest stages of development. This helps to minimize the risks associated with releasing new versions and ensures platform stability. Continuous Deployment (CD) automates the process of releasing new features and updates, enabling Spotify to deliver new features to users quickly. For example, new features or updates can be released multiple times a day without interrupting the user experience. Performance monitoring and feedback gathering are an integral part of DevOps at Spotify. The company uses powerful monitoring tools such as Prometheus and Grafana to collect detailed metrics about system performance in real time. Prometheus collects and stores metrics that are used to identify potential issues and analyse performance, while Grafana provides visualization of this data, enabling teams to make quick decisions about optimizing and scaling resources. This is especially important for Spotify, which serves millions of users worldwide, and every delay or performance hit can have a significant impact on the user experience.

One of the key cultural changes that Spotify has implemented to support DevOps is the creation of cross-functional teams known as "Squads". These teams work autonomously and are responsible for specific aspects of the product. Each "Squad" is made up of developers, testers, DevOps engineers, and other specialists who work together to achieve common goals. This structure allows Spotify to quickly adapt to changes and respond quickly to market needs. In addition, Spotify has implemented a "Tribe" structure, which brings together several "Squads" working on broader areas of product development. This organizational structure promotes effective collaboration between different teams, ensuring consistency of action and high speed of project implementation. By implementing these practices and technologies, Spotify has been able to achieve a high level of performance and reliability of its platform, while maintaining flexibility in the development and implementation of new features. This has allowed the company to become one of the leaders in the audio streaming market and maintain a high quality of service for its users.

Airbnb actively uses Kubernetes to manage its microservices' infrastructure. Kubernetes allows Airbnb to automate the deployment, scaling, and management of containers, which ensures high availability and stability of services even during peak loads. For example, during high seasons, such as holidays or major events, when the number of users increases dramatically, Kubernetes automatically scales resources to handle the increased load. This allows Airbnb to keep the service running smoothly, minimizing downtime and disruptions. Using Kubernetes also simplifies infrastructure management, as it allows for rapid deployment and upgrades of individual components without impacting other parts of the system.

Airbnb actively integrates continuous integration (CI) and continuous deployment (CD) practices as an integral part of their DevOps approach. CI/CD automates the processes of building, testing, and deploying code, which can significantly reduce the time from development to deployment of new features or fixes. For example, new features or bug fixes can be released several times a day, allowing a company to respond quickly to changing user needs or vulnerability patches. This also reduces the risk of human error and improves the quality of the final product. To ensure high performance and stability of its services, Airbnb actively uses tools to monitor and manage feedback. Prometheus collects and stores metrics about system performance, allowing DevOps teams to identify potential problems at an early stage. Using Grafana to visualize this data helps to make quick and accurate decisions about optimizing and scaling resources. For example, if the system detects an increase in latency in responding to requests, teams can respond quickly by adjusting configurations or adding resources to address the issue. This allows Airbnb to maintain a high quality of user experience, which is critical for a company with millions of users worldwide.

The implementation of DevOps at Airbnb also includes important cultural and organizational changes. The company is creating cross-functional teams that bring together developers, DevOps engineers, testers, and other specialists responsible for different aspects of the product. These teams work autonomously, which allows them to quickly adapt to changes and implement innovations. In addition, this structure promotes closer cooperation between development and operations teams, which improves communication and coherence. For example, the team responsible for a particular microservice can quickly make changes or introduce new features without waiting for other teams to make decisions.

Airbnb actively uses the IaC approach to automate infrastructure management and configuration. Thanks to IaC, Airbnb can easily create and manage infrastructure resources such as servers, network settings, and databases using software code. This significantly reduces the risks associated with human error and ensures repeatable and standardized environments. In addition, automating these processes allows the company to quickly scale its resources in response to growing needs, which is especially relevant during peak times. By adopting these practices and technologies, Airbnb has been able to significantly improve the efficiency and reliability of its services, ensuring uninterrupted operation even during high loads. These innovations have allowed the company not only to maintain a high quality of user experience, but also to quickly adapt to changes in the market, remaining one of the leaders in the rental and hospitality industry.

Thus, Kubernetes is a key tool for orchestrating microservices in the context of DevOps. Its functionalities provide automation, high availability, scalability, and integration with CI/CD processes, which allows organizations to effectively manage their containerized applications and ensure high performance and reliability of their services. Analysing the above examples, several important conclusions can be drawn. Firstly, the implementation of Kubernetes for microservices orchestration is a key component of a successful DevOps implementation. It allows automating many aspects of infrastructure management, ensuring high availability, reliability, and scalability of services. Secondly, integration with CI/CD processes is critical to ensure the continuous release of new features and updates, which allows companies to respond quickly to changes in the market and user needs. Third, real-life examples of DevOps implementation at companies such as Netflix, Spotify, and Airbnb demonstrate that the successful use of this methodology can significantly improve development efficiency, and software quality, and optimize infrastructure management costs. They confirm that the correct implementation of DevOps tools and practices can lead to significant improvements in software development and management. These examples also highlight the importance of cultural change in organizations, including building a culture of collaboration between development and operations teams. Investing in employee training in new DevOps tools and practices is critical to success. Continuously monitoring system performance and receiving feedback allows quickly identifying and fixing problems, which contributes to continuous product improvement.

DISCUSSION

This study has examined in detail the implementation and strengthening of the DevOps approach in organizations, which is a complex and multifaceted process that requires a comprehensive approach and a deep understanding of both technical and cultural aspects. One of the most important components of a successful DevOps implementation is a cultural change in the organization. Fostering a culture of collaboration between development and operations teams is critical to ensuring effective information sharing and project collaboration. This includes creating an environment where each team member feels responsible for the end result, not just their own narrow area of work. This environment encourages more open discussion of problems and faster resolution. Employees should be trained in modern DevOps tools and practices, such as CI/CD, containerization, and test automation. For example, Continuous Integration and Continuous Deployment (CI/CD) are fundamental DevOps practices that ensure that new versions of software are constantly updated and released. This reduces the time to release new features and improvements, increasing the speed and flexibility of the organization to respond to market needs.

Similar conclusions were reached by J. Díaz *et al.* (2021), who explored the reasons why many companies are implementing a DevOps culture in their organizations. They noted that DevOps allows organizations to become more agile in their development processes. This means that companies can quickly respond to market changes and user requirements by adapting their products and services to new conditions. DevOps helps to accelerate development and deployment cycles, which allows new features and updates to be brought to market faster. Process automation is one of the key components of DevOps, as it allows achieving high efficiency and quality of software development. One of the main advantages of automation is the ability to detect errors at the early stages of development. This is achieved through the use of automated tests that can quickly identify defects in the code. This allows development teams to fix bugs quickly, reducing overall fix costs and minimizing the risk of serious problems later in the development process. Continuous integration and deployment (CI/CD) systems are an integral part of DevOps process automation. They automate the compilation, testing, and deployment of code, which significantly reduces the time between writing code and deploying it to a production environment. Thanks to CI/CD automation, teams can release new product versions more frequently, allowing them to respond more quickly to changing market and user requirements. The continuity of the development and deployment processes also ensures more stable and reliable software performance, as each change is put through standardized and automated testing procedures.

Studies in the same direction were conducted by I. Karamitsos *et al.* (2020), who investigated the application of DevOps practices in the context of continuous automation of machine learning processes. The authors found that the integration of DevOps practices into machine learning (ML) can significantly improve the efficiency of developing and deploying machine learning models. S.M. Mohammad (2018) studied the optimization of DevOps automation in the context of cloud applications, identifying several crucial aspects that affect the effectiveness of DevOps implementation in cloud environments. He focused on the use of automated tools for managing cloud resources, such as automatic scaling, configuration management, and application deployment.

Continuously monitoring system performance and receiving feedback is an integral part of DevOps and has a significant impact on software quality and stability. Using metrics, logs, and bug tracking tools allows quickly determining and fixing problems at an early stage, which helps maintain a high level of system performance. Monitoring provides continuous control over aspects such as response time, resource utilization, and overall system health, enabling responding quickly to potential issues. Analysing event logs and tracking errors using tools such as Sentry or Rollbar provides detailed information about the causes of faults and anomalies, which is important for their prompt resolution. Establishing processes for receiving feedback from users is also critical for continuous product improvement. Automated systems for collecting and analysing feedback help to identify systemic issues and trends, which ensures faster response to user needs and continuous product improvement, which is the basis of successful DevOps practice.

L. Giamattei *et al.* (2024) reached a similar conclusion in their paper on monitoring tools for DevOps and microservices. They emphasized that effective monitoring is critical to ensure the stability and performance of systems in a DevOps environment. The authors highlighted the importance of using tools that allow not only tracking system performance, but also analysing data to detect anomalies and potential problems in a timely manner.

Microservices orchestration is critical for successful DevOps implementation, and Kubernetes is one of the key tools in this process. Kubernetes automates the deployment, scaling, and management of containerized applications, ensuring high availability and reliability of the system. Thanks to automatic scaling, the system can adapt to changing workloads, ensuring efficient use of resources and maintaining performance. Load balancing between containers avoids overloading of individual components, which ensures stable operation of applications. Kubernetes also provides automatic disaster recovery, which is critical for maintaining business continuity. Thanks to these features, Kubernetes facilitates the efficient management of large container clusters, which ensures the scalability and reliability of microservice architectures.

Similar conclusions were reached by S. Baškarada *et al.* (2018), who analysed the practical possibilities and challenges of implementing this architecture in their work on microservices' architecture in the context of modern IT systems. They emphasized that although the microservice architecture provides flexibility and scalability, its implementation will require careful orchestration and management planning to ensure the efficient functioning of the system. C. Bühler (2021), studying microservices in the context of DevOps, pointed out the importance of integrating orchestration tools such as

Kubernetes to automate management tasks and ensure continuous application deployment. He noted that Kubernetes makes it easy to manage deployed microservices, which significantly increases the reliability and scalability of applications within the DevOps approach.

Real-life examples of DevOps implementation in companies such as Netflix and Amazon clearly demonstrate the effectiveness of this methodology. Netflix actively uses DevOps to achieve high availability and scalability of its services. The company uses Kubernetes to manage containers and automate the deployment of microservices. This allows it to provide continuous access to services even in the event of heavy loads, downloads, or failures. Amazon is also a prime example of successful DevOps implementation. The company uses test and deployment automation to increase the speed and quality of development. In addition, Amazon uses microservice orchestration to increase the reliability of its infrastructure, which allows it to manage huge amounts of data and complex system architectures without interruption. Both of these examples confirm that DevOps can significantly improve productivity and efficiency in large organizations. V. Sharma *et al.* (2024) focused on improving software automation through DevOps adoption. They noted that DevOps integration provides effective automation of processes, including testing, deployment, and monitoring, which is critical to reducing development time and improving product quality. Overall, the findings show that successfully implementing or strengthening a DevOps approach in organizations requires a comprehensive approach that encompasses both technical and cultural aspects.

CONCLUSIONS

The study found that successful implementation of DevOps requires profound cultural changes in an organization, including the formation of a culture of collaboration between development and operations teams. This ensures effective information sharing and joint work on projects, which contributes to overall productivity. Investing in employee training in new DevOps tools and practices, such as CI/CD, containerization, and test automation, is critical to success. They allow increasing the efficiency and speed of development, as well as adapting to technological changes.

The work showed that test automation and the implementation of continuous integration and deployment (CI/CD) systems are key to ensuring high code quality and reducing the time to release new product versions. This allows quickly identifying and eliminating errors, reducing the cost of fixing them and speeding up the development process. Continuous monitoring of system performance and receiving feedback from users contribute to the continuing improvement of the product. The use of tools for collecting metrics and tracking bugs is essential to maintain the high quality and reliability of the product. Orchestrating microservices using tools such as Kubernetes ensures scalability and reliability. Kubernetes automates the deployment, scaling, and management of containerized applications, which contributes to the high availability and reliability of microservice architectures. Practical examples from companies such as Netflix, Spotify, and Airbnb confirm that DevOps implementation significantly increases the efficiency of software development and management. These examples demonstrate that the successful use of DevOps tools and practices can significantly improve software quality, optimize infrastructure management costs, and ensure the rapid and continuous release of new features.

Prospects for further research include analysing the impact of new technologies, such as artificial intelligence and machine learning, on DevOps processes, including their use for automation, forecasting and improving code quality. The study of the integration of DevOps practices with cloud platforms and services, including the impact of cloud environments on the efficiency and scalability of DevOps processes. The study's limitations stem from the insufficient technical characteristics and capabilities of the DevOps tools and practices under study, which may affect the overall assessment of their effectiveness, as certain aspects of technologies or tools may not be fully represented in the analysis.

ACKNOWLEDGEMENTS

None.

CONFLICT OF INTEREST

None.

REFERENCES

[1] Akbar, M.A., Smolander, K., Mahmood, S., & Alsanad, A. (2022). Toward successful DevSecOps in software development organizations: A decision-making framework. *Information and Software Technology*, 147, article number 106894. doi: 10.1016/j.infsof.2022.106894.

[2] Azad, N. (2022). Understanding DevOps critical success factors and organizational practices. In *IWSiB '22: Proceedings of the 5th international workshop on software-intensive business: Towards sustainable software business* (pp. 83-90). New York: Association for Computing Machinery. doi: 10.1145/3524614.3528627.

[3] Baškarada, S., Nguyen, V., & Koronios, A. (2018). Architecting microservices: Practical opportunities and challenges. *Journal of Computer Information Systems*, 60(5), 428-436. doi: 10.1080/08874417.2018.1520056.

[4] Bühler, C. (2021). *Microservices in a DevOps context*. (Thesis, OST Eastern Switzerland University of Applied Sciences, Rapperswil-Jona, Switzerland).

[5] Díaz, J., López-Fernández, D., Pérez, J., & González-Prieto, Á. (2021). Why are many businesses instilling a DevOps culture into their organization? *Empirical Software Engineering*, 26(2), article number 25. doi: 10.1007/s10664-020-09919-3.
[6] Giamattei, L., *et al.* (2024). Monitoring tools for DevOps and microservices: A systematic grey literature review. *Journal of Systems and Software*, 208, article number 111906. doi: 10.1016/j.jss.2023.111906.
[7] Govil, N., Saurakhia, M., Agnihotri, P., Shukla, S., & Agarwal, S. (2020). Analyzing the behaviour of applying agile methodologies & DevOps culture in e-commerce web application. In *2020 4th international conference on trends in electronics and informatics (ICOEI) (48184)* (pp. 899-902). Tirunelveli: Institute of Electrical and Electronics Engineers. doi: 10.1109/ICOEI48184.2020.9142895.
[8] Karamitsos, I., Albarhami, S., & Apostolopoulos, C. (2020). Applying DevOps practices of continuous automation for machine learning. *Information*, 11(7), article number 363. doi: 10.3390/info11070363.
[9] Khan, M.S., Khan, A.W., Khan, F., Khan, M.A., & Whangbo, T.K. (2022). Critical challenges to adopt DevOps culture in software organizations: A systematic review. *IEEE Access*, 10, 14339-143349. doi: 10.1109/ACCESS.2022.3145970.
[10] Kornitskyi, S. (2024). *Implementation of a service quality management system in technology companies based on international standards ISO/IEC 20000*. (Master's thesis, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine).
[11] Kravchuk, O. (2023). CI/CD implementation model for optimizing IT project management. *Measuring and Computing Devices in Technological Processes*, 3, 73-82. doi: 10.31891/2219-9365-2023-75-8.
[12] Ljunggren, D. (2023). *DevOps: Assessing the factors influencing the adoption of infrastructure as code, and the selection of infrastructure as code tools: A case study with Atlas Copco*. (Master's thesis, KTH Royal Institute of Technology, Stockholm, Sweden).
[13] Luz, W.P., Pinto, G., & Bonifácio, R. (2018). Building a collaborative culture: A grounded theory of well succeeded devops adoption in practice. In *ESEM '18: Proceedings of the 12th ACM/IEEE international symposium on empirical software engineering and measurement* (article number 6). New York: Association for Computing Machinery. doi: 10.1145/3239235.3240299.
[14] Mohammad, S.M. (2018). Streamlining DevOps automation for Cloud applications. *International Journal of Creative Research Thoughts (IJCRT)*, 6(4), 955-959.
[15] Morris, K. (2020). *Infrastructure as code: Dynamic systems for the cloud age*. Sebastopol: O'Reilly Media, Inc.
[16] Muñoz, M., & Rodríguez, M.N. (2021). A guidance to implement or reinforce a DevOps approach in organizations: A case study. *Journal of Software Evolution and Process*, 36(3), article number e2342. doi: 10.1002/smr.2342.
[17] Mykhalchenko, H., & Tytarenko, M. (2023). Data analytics and personalized marketing strategies in e-commerce platforms. *Futurity Economics & Law*, 3(3), 115-139. doi: 10.57125/FEL.2023.09.25.07.
[18] Radstaak, J. (2019). *Developing a DevOps maturity model: A validated model to evaluate the maturity of DevOps in organizations*. (Master's essay, University of Twente, Enschede, the Netherlands).
[19] Rafi, S., Yu, W., Akbar, M.A., Mahmood, S., Alsanad, A., & Gumaei, A. (2020). Readiness model for DevOps implementation in software organizations. *Journal of Software Evolution and Process*, 33(4), article number e2323. doi: 10.1002/smr.2323.
[20] Schünemann, C. (2023). *Automating the build and test process of a regulated software project using continuous delivery pipelines*. (Bachelor's thesis, Technical University Ingolstadt of Applied Sciences, Ingolstadt, Germany).
[21] Sharma, V., Shrivastava, V., Pandey, A., & Gupta, P. (2024). A basic introduction to DevOps. *International Journal of Research Publication and Reviews*, 5(3), 725-731.
[22] Singh, A., & Mansotra, V. (2021). A comparison on continuous integration and continuous deployment (CI/CD) on cloud based on various deployment and testing strategies. *International Journal for Research in Applied Science and Engineering Technology*, 9(6), 4968-4977. doi: 10.22214/ijraset.2021.36038.
[23] Singh, V. (2022). *Developing a CI/CD pipeline with GitLab*. (Bachelor's thesis, Turku University of Applied Sciences, Turku, Finland).
[24] Smart, J.F. (2011). *Jenkins: The definitive guide*. Sebastopol: O'Reilly Media, Inc.
[25] Vatsyk, Yu. (2024). *Research and analysis of the relevance of Kubernetes in the modern IT market*. (Master's thesis, King Danylo University, Ivano-Frankivsk, Ukraine).
[26] Waseem, M., Liang, P., & Shahin, M. (2020). A systematic mapping study on microservices architecture in DevOps. *Journal of Systems and Software*, 170, article number 110798. doi: 10.1016/j.jss.2020.110798.
[27] Zhou, N., Georgiou, Y., Pospieszny, M., Zhong, L., Zhou, H., Niethammer, C., Pejak, B., Marko, O., & Hoppe, D. (2021). Container orchestration on HPC systems through Kubernetes. *Journal of Cloud Computing*, 10(1), article number 16. doi: 10.1186/s13677-021-00231-z.

Стратегії впровадження або посилення підходу DevOps в організаціях: аналіз та приклади

Богдан Федоришин

Магістр
Національний університет «Львівська політехніка»
79000, вул. Степана Бандери, 12, м. Львів, Україна
https://orcid.org/0009-0005-3779-0186

Анотація. Мета роботи полягала в аналізі впровадження DevOps в організаціях, зокрема в оцінці впливу автоматизації процесів, CI/CD, моніторингу та оркестрації мікросервісів на ефективність розробки та управління. Було створено методологію, яка дозволяє організаціям ефективно впроваджувати та посилювати підхід DevOps, досягаючи високих результатів у розвитку і управлінні програмним забезпеченням. У дослідженні було розглянуто стратегії культурної зміни та навчання, що включають формування культури співпраці між командами та розвиток навичок у сучасних інструментах DevOps. Автоматизація процесів, зокрема автоматизація тестування та інтеграція та розгортання, відіграє важливу роль у підвищенні якості коду і зменшенні часу розробки. Інфраструктура як код дозволяє управляти та автоматизувати конфігурацію інфраструктури, забезпечуючи швидке масштабування середовищ. Робота виявила, що моніторинг продуктивності та зворотний зв'язок є критичними для виявлення проблем на ранніх етапах і постійного вдосконалення продукту. Оркестрація мікросервісів, зокрема за допомогою Kubernetes, забезпечує автоматизацію розгортання і масштабування контейнеризованих додатків, що критично важливо для сучасних архітектур з високими вимогами до продуктивності та доступності. Приклади з практики таких компаній, як Netflix, Spotify і Airbnb, демонструють успішне застосування DevOps практик і технологій для досягнення високої доступності, масштабованості та надійності інфраструктур. Результати дослідження підтверджують, що впровадження DevOps веде до значного підвищення ефективності розробки, покращення якості програмного забезпечення та оптимізації витрат на управління інфраструктурою. Крім того, впровадження культурних змін і підвищення рівня співпраці між командами розробки та операцій сприяє більшій гнучкості та швидкості реагування на зміни ринкових умов

Ключові слова: оркестрація мікросервісів; інструменти; автоматизація тестування; CI/CD процеси; ефективність розробки

BULLETIN of Cherkasy State Technological University

Journal homepage: https://bulletin-chstu.com.ua/en

Vol. 29 No. 2. 2024

UDC 519.7
DOI: 10.62660/bcstu/2.2024.70

Fuzzy modelling of the environmental component of social security

Marianna Sharkadi

PhD in Economics, Associate Professor
Uzhhorod National University
88000, 14 Universytetska Str., Uzhhorod, Ukraine
https://orcid.org/0000-0002-1850-996X

Adam Dorovtsi*

Postgraduate Student, Lecturer
Uzhhorod National University
88000, 14 Universytetska Str., Uzhhorod, Ukraine
Ferenc Rakoczi II Transcarpathian Hungarian College of Higher Education
90202, 6 Kossuth Sq., Berehove, Ukraine
https://orcid.org/0000-0003-4038-4945

Abstract. Economic growth has several negative consequences for the environment. At first glance, the impact of the environment on the well-being of the population may seem insignificant. Still, several scientific studies demonstrate that the state of the environment has no less impact on the well-being of the population than economic or social factors. This study aimed to assess Ukraine's environmental well-being in comparison with the member countries of the Organisation for Economic Co-operation and Development. The assessment of environmental well-being is based on indicators from the Organisation for Economic Co-operation and Development Better Life Index and the Environmental Performance Index. Fuzzy set theory is used to research and evaluate environmental well-being. During the study, a fuzzy inference system was constructed, which was used to obtain an assessment of Ukraine's environmental well-being compared to the Organisation for Economic Co-operation and Development countries. Three indicators were used in the study: air pollution, sanitation and drinking water, and agriculture. Results indicate that compared to the Organisation for Economic Co-operation and Development countries, Ukraine is on the border between medium and high levels of environmental well-being. The research findings are supported by the Environmental Performance Index, where Ukraine ranked 41st out of 180 countries in 2024, climbing 11 positions compared to 2022. Although Ukraine has already demonstrated progress in the Environmental Performance Index in 2024 compared to the 2022 results, it can maintain such a trend only if it knows which specific environmental indicators need improvement. Therefore, the practical value of this research lies in demonstrating the environmental indicators that require improvement

Keywords: ecology; fuzzy sets; fuzzy inference system; environmental well-being; mathematical modelling

INTRODUCTION

Social security is a fundamental condition for the stable development of society, ensuring social justice and creating conditions for a decent life for every citizen. It requires a comprehensive approach that covers all major aspects of life. One of the main elements of social security is the well-being of the population. Like social security, it also consists of several main components, such as economic, social, and environmental. The environmental component includes aspects that directly affect people's health, life expectancy, labour productivity,

Article's History: Received: 30.01.2024; Revised: 25.04.2024; Accepted: 27.05.2024.

Suggested Citation:
Sharkadi, M., & Dorovtsi, A. (2024). Fuzzy modelling of the environmental component of social security. *Bulletin of Cherkasy State Technological University*, 29(2), 70-78. doi: 10.62660/bcstu/2.2024.70.

*Corresponding author

and psychological comfort. Taking into account environmental indicators when determining the level of population well-being allows for an assessment of not only the current state but also future prospects. This is an approach to sustainable development that is focused on preserving natural resources and ecosystems for future generations while ensuring a high standard of living for the current inhabitants of the planet.

S. Lyndiuk (2022) notes that the concept of "social security" is complex and encompasses various aspects such as economic, political, social, humanitarian, and value-based dimensions. The author highlights all the factors that influence human activity, including living conditions and quality of life. According to this, research aimed at determining the level of population well-being is not only of interest but is also one of the key elements of social security. However, defining the level of population well-being is a multifaceted task. Well-being can also be divided into three categories: economic, social, and environmental. There is also a significant amount of international scientific research on the impact of the environment on population well-being. In the study of C. Li (2022) it is emphasised that with increasing pressure on the global environmental landscape, natural capital has become scarce. The research by E. Kalaidjian *et al*. (2024) explores the active and passive impact of the environment on human well-being. The findings of J. Wang & D. Tang (2023) show that air pollution negatively affects the population's well-being. The authors propose an "environmental tax" to regulate air pollution while simultaneously enhancing well-being. The research of A. Vijaikis & M.S. Poškus (2024) shows that pro-environmental actions by citizens, i.e., individuals who care for the environment in their private lives, can have a positive impact on mental health.

The aforementioned studies are excellent examples of the extent to which environmental indicators influence population well-being. In previous research, the authors considered the construction of a fuzzy mathematical model to determine the level of economic (Sharkadi & Dorovtsi, 2024a) and social well-being (Sharkadi & Dorovtsi, 2024b) of the population. Fuzzy inference systems are also used by many scientists for similar research (Saini *et al.*, 2022). However, the mentioned study is not dedicated to determining the level of environmental well-being of a particular country but to building a fuzzy inference system to determine indoor air quality.

This article proposes the consideration of the third key element in determining population well-being: environmental well-being. The research aimed to develop a fuzzy mathematical model to determine the level of environmental well-being in Ukraine compared to the member countries of the Organisation for Economic Co-operation and Development (OECD). Within the framework of the study, it is proposed to solve the following tasks:

- classification of existing statistical data by country;
- definition of membership functions for the fuzzy inference system;
- construction of a knowledge base.

MATERIALS AND METHODS

The first step in conducting the research was to construct a hierarchical structure. The most typical and simple hierarchical structure is a two-level one, which has only one higher-level coordinating element and *n* subordinate lower-level elements (Fig. 1). Such a structure allows for the construction of any multi-level hierarchies from two-level subsystems, as if from individual modules.

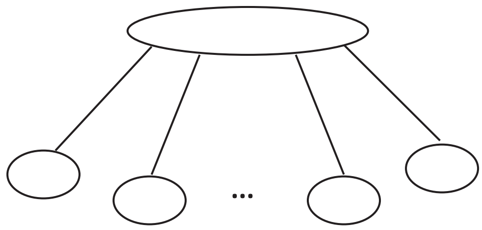


Figure 1. Two-level hierarchical structure
Source: developed by the authors

To achieve the stated goal, it is proposed to use fuzzy set theory. The use of fuzzy mathematics in determining the level of environmental well-being is justified for several key reasons. The level of environmental well-being depends on numerous factors that often have an uncertain or subjective nature. For example, the definition of "clean" or "polluted" air can be ambiguous and depend on the context or regulatory standards, which may also be fuzzy. Also, when determining the level of environmental well-being, it is necessary to consider various sources of information and expert assessments, which may be contradictory or incomplete. Fuzzy mathematics provides tools for integrating such data, allowing for informed conclusions even under conditions of incomplete information.

In previous studies, the authors examined issues of economic well-being and explored the social aspects of population well-being (Sharkadi & Dorovtsi, 2024a; Sharkadi & Dorovtsi, 2024b). This study is the third part of the aforementioned research, examining the third main component of population well-being – environmental well-being. To investigate the level of environmental well-being, three indicators were used: air pollution, sanitation and drinking water, and agricultural activities. The input data for air pollution was the annual average concentration of particulate matter with a diameter of up to 2.5 μm (PM2.5). It is important to note that according to the World Health Organization, the annual average concentration of PM2.5 in the air should not exceed 5 μg/m^3 (European Environment Agency, 2022). Such a level can be achieved by only a few countries. The indicators "sanitation and drinking water" and "agriculture" were composed using data from the Environmental Performance Index. In the course of the study, statistical data for all countries that are members of the OECD were collected and classified for all the aforementioned indicators.

To determine the level of environmental well-being, it is proposed to use a fuzzy inference system. There are several fuzzy inference algorithms, but in this study, the Mamdani fuzzy inference algorithm was used. Besides being the most common fuzzy inference algorithm, the Mamdani algorithm has several other advantages, such as being easy to understand, efficient for manual input, and the rule base is transparent and comprehensible. The Mamdani fuzzy inference system was built in the Fuzzy Logic Designer application of the Matlab software package. The Mamdani algorithm consists of the following steps: formation of a knowledge base; fuzzification of input variables; aggregation of antecedents in fuzzy production rules; activation of consequents of fuzzy production rules; accumulation of consequents of fuzzy production rules; defuzzification – i.e., converting to a crisp value.

RESULTS AND DISCUSSION

In this study, a fuzzy mathematical model was constructed to determine the level of environmental well-being in Ukraine compared to the member countries of the OECD. Determining population well-being is a complex task. The OECD Better Life Index measures the well-being of OECD member countries based on a range of indicators. In this study, the indicators were grouped and a hierarchical system was constructed, dividing the indicators that influence the level of well-being into three main parts. These three main groups include economic indicators, social indicators, and environmental indicators. The constructed hierarchical structure is shown in Figure 2.

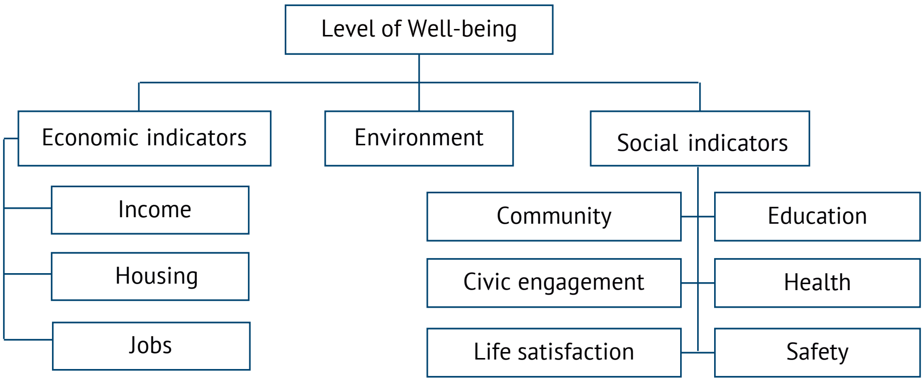


Figure 2. Hierarchical structure for determining the level of well-being
Source: developed by the authors

After gathering the information, the available data (the annual mean concentration of particulate matter with a diameter of up to 2.5 μm (PM2.5) in ambient air, "sanitation and drinking water" indicator of the Environmental Performance Index, "agriculture" indicator of the Environmental Performance Index) were divided into quartiles. Thus, the data was divided into three parts: low, medium, and high. After classifying the data by quartiles, a fuzzy inference system was constructed using the Matlab mathematical programming package. To construct the fuzzy inference system, membership functions were defined and used. Thus, for data belonging to the "low" category, a linearly z-shaped membership function was proposed, generated by the program according to the following formula:

$$f(x,a,b) = \begin{cases} 1, x \le a; \\ \frac{b-x}{b-a}, a < x < b; \\ 0, b \le x. \end{cases} \quad (1)$$

For data belonging to the "medium" category, a triangular membership function was proposed, described by the following formula:

$$f(x,a,b,c) = \begin{cases} 0, x \le a; \\ \frac{x-a}{b-a}, a \le x \le b; \\ \frac{c-x}{c-b}, b \le x \le c; \\ 0, c \le x. \end{cases} \quad (2)$$

For data belonging to the "high" category, a linearly s-shaped membership function was proposed, described by the following formula:

$$f(x,a,b) = \begin{cases} 1, x \le a; \\ \frac{x-a}{b-a}, a < x < b; \\ 0, b \le x. \end{cases} \quad (3)$$

The constructed fuzzy inference system is illustrated in Figure 3.

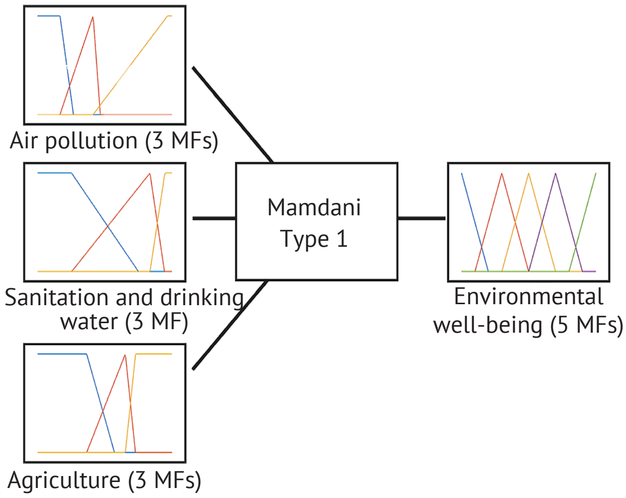


Figure 3. FIS structure for determining the level of environmental well-being
Source: developed by the authors

As shown in Figure 3, the Mamdani algorithm was used in the study. The fuzzy logical inference based on the Mamdani algorithm is formulated according to the following formula:

$$\bigcup_{p=1}^{k_j}\left(\bigcap_{i=1}^{n} x_i = a_{i,jp} \; with \; weight \; w_{jp}\right) \rightarrow$$
$$\rightarrow y = d_j, j = \overline{1,m}, \qquad (4)$$

where $a_{i,jp}$ – the fuzzy term used to evaluate the variable x_i in the row with the number $jp(p = \overline{1,k_j})$; w_{jp} – the weight coefficient of the rule with ordinal number jp within the range [0,1], which specifies the relative weight of the rule in the fuzzy logical inference; d_j – the fuzzy conclusion of the j-th rule (in the Mamdani-type algorithm, the conclusions of the rules d_j are defined by fuzzy terms); m – the number of terms used for the linguistic assessment of the output variable. Figure 4 shows a typical structure of a fuzzy inference model.

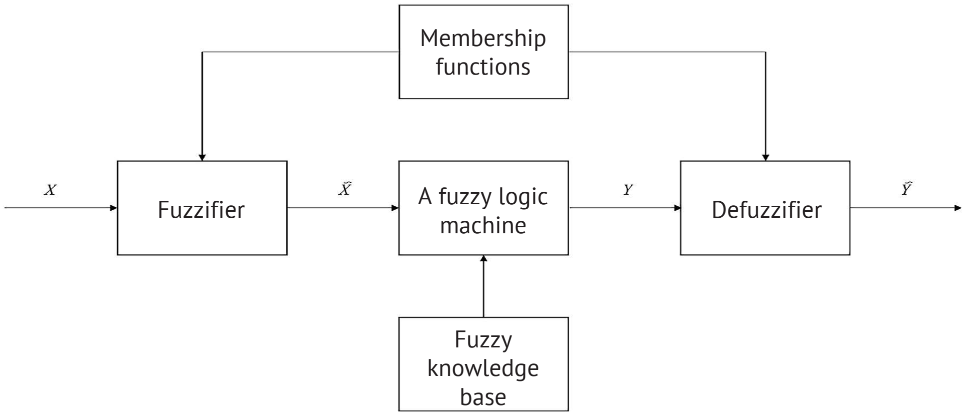


Figure 4. Typical structure of a fuzzy inference model
Source: developed by the authors based on the study A. Abramova (2022)

In the Mamdani algorithm, the construction of a knowledge base is a crucial part. Therefore, the next task was to create a knowledge base. The knowledge base in the Mamdani algorithm consists of logical rules of the "IF-THEN" type. The constructed knowledge base consists of 27 logical rules. The knowledge base constructed during the study is presented in Table 1. Once the knowledge base has been established, the output surface of the constructed model can be viewed in the Control Surface window of the Matlab program. Figures 5 and 6 display the output surface of the fuzzy system in relation to the input values.

Table 1. Knowledge base for the built fuzzy inference system

No. of the rule	Description of the logical rule	Weight	Name of the rule
1	If Air pollution is Low and Sanitation and drinking water is Low then Environmental well-being is Low	1	Rule 1
2	If Air pollution is Low and Sanitation and drinking water is Medium then Environmental well-being is High	1	Rule 2
3	If Air pollution is Low and Sanitation and drinking water is High then Environmental well-being is Very high	1	Rule 3
4	If Air pollution is Medium and Sanitation and drinking water is Low then Environmental well-being is Low	1	Rule 4

Continued Table 1.

No. of the rule	Description of the logical rule	Weight	Name of the rule
5	If Air pollution is Medium and Sanitation and drinking water is Medium then Environmental well-being is Medium	1	Rule 5
6	If Air pollution is Medium and Sanitation and drinking water is High then Environmental well-being is High	1	Rule 6
7	If Air pollution is High and Sanitation and drinking water is Low then Environmental well-being is Very low	1	Rule 7
8	If Air pollution is High and Sanitation and drinking water is Medium then Environmental well-being is Low	1	Rule 8
9	If Air pollution is High and Sanitation and drinking water is High then Environmental well-being is Medium	1	Rule 9
10	If Air pollution is Low and Agriculture is Low then Environmental well-being is Low	1	Rule 10
11	If Air pollution is Low and Agriculture is Medium then Environmental well-being is High	1	Rule 11
12	If Air pollution is Low and Agriculture is High then Environmental well-being is Very high	1	Rule 12
13	If Air pollution is Medium and Agriculture is Low then Environmental well-being is Low	1	Rule 13
14	If Air pollution is Medium and Agriculture is Medium then Environmental well-being is Medium	1	Rule 14
15	If Air pollution is Medium and Agriculture is High then Environmental well-being is High	1	Rule 15
16	If Air pollution is High and Agriculture is Low then Environmental well-being is Very low	1	Rule 16
17	If Air pollution is High and Agriculture is Medium then Environmental well-being is Low	1	Rule 17
18	If Air pollution is High and Agriculture is High then Environmental well-being is Medium	1	Rule 18
19	If Sanitation and drinking water is Low and Agriculture is Low then Environmental well-being is Very low	1	Rule 19
20	If Sanitation and drinking water is Low and Agriculture is Medium then Environmental well-being is Low	1	Rule 20
21	If Sanitation and drinking water is Low and Agriculture is High then Environmental well-being is Medium	1	Rule 21
22	If Sanitation and drinking water is Medium and Agriculture is Low then Environmental well-being is Low	1	Rule 22
23	If Sanitation and drinking water is Medium and Agriculture is Medium then Environmental well-being is Medium	1	Rule 23
24	If Sanitation and drinking water is Medium and Agriculture is High then Environmental well-being is High	1	Rule 24
25	If Sanitation and drinking water is High and Agriculture is Low then Environmental well-being is Low	1	Rule 25
26	If Sanitation and drinking water is High and Agriculture is Medium then Environmental well-being is High	1	Rule 26
27	If Sanitation and drinking water is High and Agriculture is High then Environmental well-being is Very high	1	Rule 27

Source: developed by the authors

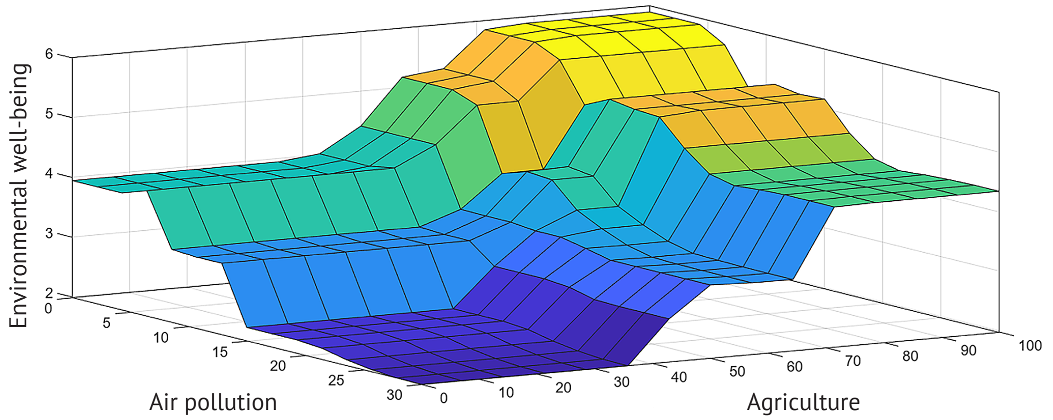


Figure 5. The output surface of the fuzzy system in relation to the input values "air pollution" and "agriculture"
Source: developed by the authors

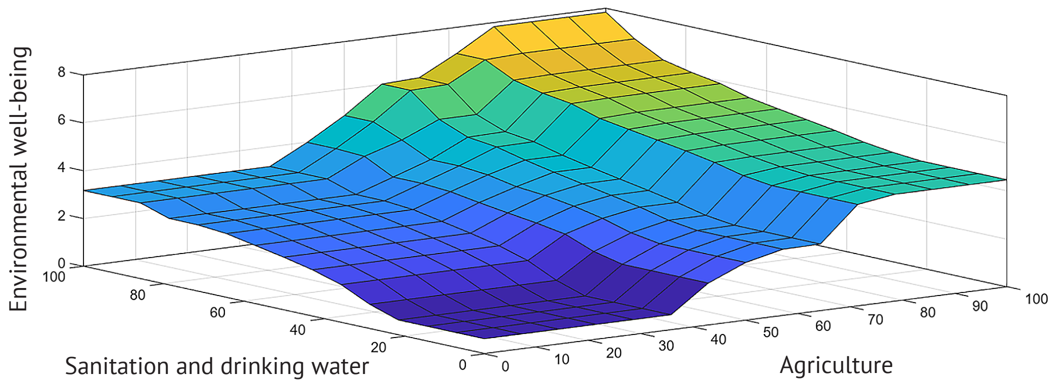


Figure 6. The output surface of the fuzzy system in relation to the input values "sanitation and drinking water" and "agriculture"
Source: developed by the authors

The final step in the research process is defuzzification. The Matlab software package, specifically the Fuzzy Logic Toolbox, offers several methods for defuzzification, including: 1) Centroid – the centre of gravity; 2) Bisector – median; 3) LOM (Largest Of Maximums) – largest of maximums; 4) SOM (Smallest Of Maximums) – smallest of maximums; 5) MOM (Mean Of Maximums) – mean of maximums. Among these, the most widely used method is the centroid or centre of gravity, which is described by the following formula:

$$y = \frac{\int_{min}^{max} x \cdot \mu(x)dx}{\int_{min}^{max} \mu(x)dx}, \tag{5}$$

where y – the defuzzification result; x – the variable corresponding to the output linguistic variable; $\mu(x)$ – the membership function of the fuzzy set corresponding to the output variable after the accumulation stage; *min* and *max* – left and right points of the support interval of the fuzzy set of the corresponding output variable (Butko *et al.*, 2022). The defuzzification result using the centre of gravity method is shown in Figure 7.

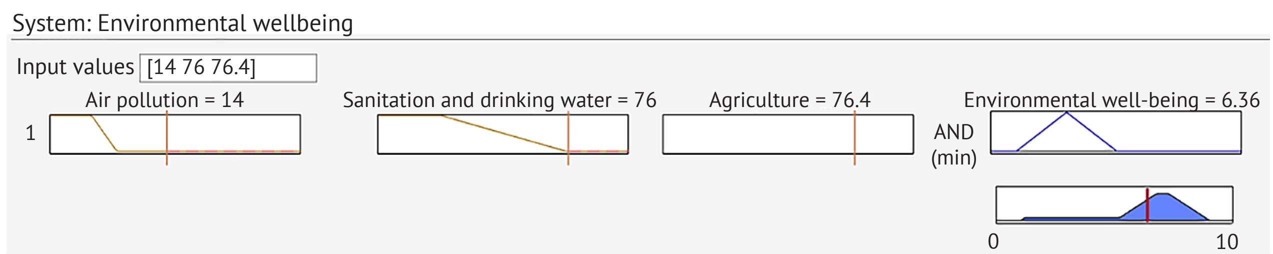


Figure 7. Clarification (defuzzification) of the built model using the centre of gravity method
Source: developed by the authors

As shown in Figure 7, the study not only constructed a fuzzy inference system capable of determining environmental well-being but also tested it on real statistical data for Ukraine. The statistical data used for testing the system includes the following:

- average annual concentration of particulate matter with a diameter of up to 2.5 μm (PM2.5) in the ambient air;
- the "sanitation and drinking water" indicator from the Environmental Performance Index;
- the "agriculture" indicator from the Environmental Performance Index.

Results show that, according to these three indicators, the environmental well-being in Ukraine is at the borderline between medium and high levels. This is confirmed by the Environmental Performance Index, where Ukraine ranked 41st out of 180 countries studied.

As of 2024, there are numerous international scientific studies examining the impact of environmental indicators on the subjective well-being of populations. Primarily, these studies focus on countries where the environmental impact (due to pollutant emissions) is at an extreme level. For instance, the influence of air quality on subjective well-being is examined in the research of L. Yuan *et al*. (2018). Similarly, the study by X. Zhang *et al*. (2017) investigates the effect of air quality on mental health and subjective well-being. These studies demonstrate that air pollution negatively affects the subjective well-being of the population. In this research, similar indicators are used; however, instead of analysing their impact on well-being, a fuzzy mathematical model has been developed to determine the level of environmental well-being.

Besides air quality, sanitation conditions and access to clean drinking water also have a significant impact on well-being. The report by C.N. Mock *et al*. (2017) demonstrates the influence of water supply infrastructure on people's mental and physical health. Furthermore, the research by A. Adukia (2017) shows that water supply infrastructure even affects school attendance, which is also a crucial indicator of social well-being. In the aforementioned studies, only the impact of air quality or water supply infrastructure on subjective well-being was examined. In this research, a broader range of environmental indicators influencing population well-being was analysed, and Ukraine's environmental indicators were compared to those of the OECD member countries.

Previous research has established a strong link between environmental indicators and population well-being, with health serving as a fundamental indicator of social well-being. Without a healthy population, it is difficult to speak of a high level of overall well-being. The studies of E. Ha (2020) and S. Khomenko *et al*. (2021) delve into the impact of air pollution on public health. These authors analysed the impact of various pollutants on public health, while the findings of the present study provide an assessment of environmental well-being. Moreover, a review of the literature reveals that air pollution has a detrimental effect not only on physical but also on mental health. Y. Song *et al*. (2020) and C. Sanduijav *et al*. (2021) explore the impact of polluted air on happiness, a crucial component of social well-being.

As previously mentioned, there is a substantial body of research dedicated to determining levels of environmental well-being. The article of X. Song *et al*. (2022) investigates the impact of the digital economy on environmental well-being. The study utilises panel data from Chinese provinces between 2011 and 2019. In their research, G. Ahumada & V. Iturra (2021) examine the effects of air pollution on the subjective well-being of the Chilean population. In contrast to these studies, this research classifies statistical data for the member countries of the OECD. Following data classification, a fuzzy logic inference system was constructed to determine the level of environmental well-being in comparison to OECD member countries.

The aforementioned studies demonstrate that environmental indicators have a significant impact on the population's well-being. However, research specifically focused on environmental well-being is scarce in the context of Ukraine. The fuzzy inference system developed in this study offers a straightforward and comprehensible method for assessing Ukraine's environmental well-being in comparison to member countries of the OECD.

CONCLUSIONS

This research underscores that human well-being is a far more complex issue than merely economic or even social indicators. By utilising the fuzzy logic system developed during the research, it is possible to easily compute even complex issues such as environmental well-being. The research successfully achieved its goal by creating a fuzzy logic system that assesses the environmental well-being of Ukraine in comparison to the member countries of the OECD. The findings highlight the importance of focusing on environmental protection in addition to economic aspirations. The proposed fuzzy logic system was tested using real statistical data for Ukraine. The results indicate that, according to three key indicators, Ukraine's environmental well-being lies between the medium and high levels. This is further validated by the Environmental Performance Index, where Ukraine ranked 41st out of 180 countries. Although Ukraine has attained a good level of environmental well-being, it is crucial not only to maintain but also to improve this level in the future, particularly concerning harmful air pollutants. The indicator used in this study (the average annual concentration of PM2.5 air pollution) exceeds the World Health Organization's recommended levels by almost threefold. As ecology is a broad concept, further research should include additional indicators that could affect the level of environmental well-being. The prospects for future studies lie in incorporating more environmental factors that influence population well-being.

ACKNOWLEDGEMENTS

This research was conducted within the framework of the state-funded research project at Uzhhorod National University, "Computer Intelligence Methods for Data Processing and Analysis" (state registration number 0121U109279).

CONFLICT OF INTEREST

None.

REFERENCES

[1] Abramova, A. (2022). Development of a Fuzzy Matlab inference algorithm for decision-making under uncertainty conditions for designed objects. In *Proceedings of 16th international conference measurement and control in complex system (MCCS-2022)*. Vinnytsia: Vinnytsia National Technical University. doi: 10.31649/mccs2022.17.

[2] Adukia, A. (2017). Sanitation and education. *American Economic Journal: Applied Economics*, 9(2), 23-59. doi: 10.1257/app.20150083.

[3] Ahumada, G., & Iturra, V. (2021). If the air was cleaner, would we be happier? An economic assessment of the effects of air pollution on individual subjective well-being in Chile. *Journal of Cleaner Production*, 289, article number 125152. doi: 10.1016/j.jclepro.2020.125152.

[4] Butko, T., Prokhorov, V., Prokhorchenko, H., & Shumyk, D. (2022). *Methodical instructions for practical classes and individual assignments in the discipline "Modern Information Systems in the Management of Railway Divisions"*. Kharkiv: Ukrainian State University of Railway Transport.

[5] European Environment Agency. (2022). *World Health Organization (WHO) air quality guidelines (AQGs) and estimated reference levels (RLs)*. Retrieved from https://www.eea.europa.eu/publications/status-of-air-quality-in-Europe-2022/europes-air-quality-status-2022/world-health-organization-who-air.

[6] Ha, E. (2020). Impact of air pollution hazards on human development. In R. Kishi & P. Grandjean (Eds.), *Health impacts of developmental exposure to environmental chemicals* (pp. 223-245). Singapore: Springer. doi: 10.1007/978-981-15-0520-1_10.

[7] Kalaidjian, E., Kurth, M., Kucharski, J., Galaitsi, S., & Yeates, E. (2024). Human well-being and natural infrastructure: assessing opportunities for equitable project planning and implementation. *Frontiers in Ecology and Evolution*, 12, article number 1271182. doi: 10.3389/fevo.2024.1271182.

[8] Khomenko, S., Cirach, M., Pereira-Barboza, E., Mueller, N., Barrera-Gómez, J., Rojas-Rueda, D., de Hoogh, K., Hoek, G., & Nieuwenhuijsen, M. (2021). Premature mortality due to air pollution in European cities: A health impact assessment. *The Lancet. Planetary Health*, 5(3), e121-e134. doi: 10.1016/S2542-5196(20)30272-2.

[9] Li, C. (2022). China's multi-dimensional ecological well-being performance evaluation: A new method based on coupling coordination model. *Ecological Indicators*, 143, article number 109321. doi: 10.1016/j.ecolind.2022.109321.

[10] Lyndiuk, S. (2022). Theoretical interpretations of the concept of "Social security" in the Ukrainian scientific discourse. *Scientific Herald: Public Administration*, 1(11), 92-113. doi: 10.33269/2618-0065-2022-1(11)-92-113.

[11] Mock, C.N., Nugent, R., Kobusingye, O., Smith, K.R., & Alwan, A. (2017). *Injury prevention and environmental health (English)*. Washington, D.C.: World Bank Group.

[12] Saini, J., Dutta, M., & Marques, G. (2022). ADFIST: Adaptive Dynamic Fuzzy Inference System Tree driven by optimized knowledge base for indoor air quality assessment. *Sensors*, 22(3), article number 1008. doi: 10.3390/s22031008.

[13] Sanduijav, C., Ferreira, S., Filipski, M., & Hashida, Y. (2021). Air pollution and happiness: Evidence from the coldest capital in the world. *Ecological Economics*, 187, article number 107085. doi: 10.1016/j.ecolecon.2021.107085.

[14] Sharkadi, M., & Dorovtsi, A. (2024a). Fuzzy models' use in sociological researches. *Scientific Bulletin of Uzhhorod University. Series of Mathematics and Informatics*, 44(1), 175-181. doi: 10.24144/2616-7700.2024.44(1).175-181.

[15] Sharkadi, M., & Dorovtsi, A. (2024b). Building a fuzzy model for determining the level of social well-being of the population. *Eastern-European Journal of Enterprise Technologies*, 4(130), 35-45. doi: 10.15587/1729-4061.2024.310142.

[16] Song, X., Tian, Z., Ding, C., Liu, C., Wang, W., Zhao, R., & Xing, Y. (2022). Digital economy, environmental regulation, and ecological well-being performance: A provincial panel data analysis from China. *International Journal of Environmental Research and Public Health*, 19(18), article number 11801. doi: 10.3390/ijerph191811801.

[17] Song, Y., Zhou, A., & Zhang, M. (2020). Exploring the effect of subjective air pollution on happiness in China. *Environmental Science and Pollution Research International*, 27(34), 43299-43311. doi: 10.1007/s11356-020-10255-8.

[18] Vijaikis, A., & Poškus, M.S. (2024). Does being an environmental citizen lead to greater well-being? The self-determination theory approach. *Sage Open*, 14(3). doi: 10.1177/21582440241267156.

[19] Wang, J., & Tang, D. (2023). Air pollution, environmental protection tax and well-being. *International Journal of Environmental Research and Public Health*, 20(3), article number 2599. doi: 10.3390/ijerph20032599.

[20] Yuan, L., Shin, K., & Managi, S. (2018). Subjective well-being and environmental quality: The impact of air pollution and green coverage in China. *Ecological Economics*, 153, 124-138. doi: 10.1016/j.ecolecon.2018.04.033.
[21] Zhang, X., Zhang, X., & Chen, X. (2017). Happiness in the air: How does a dirty sky affect mental health and subjective well-being? *Journal of Environmental Economics and Management*, 85, 81-94. doi: 10.1016/j.jeem.2017.04.001.

Нечітке моделювання екологічної складової соціальної безпеки

Маріанна Шаркаді
Кандидат економічних наук, доцент
Ужгородський національний університет
88000, вул. Університетська, 14, м. Ужгород, Україна
https://orcid.org/0000-0002-1850-996X
Адам Доровці
Аспірант, викладач
Ужгородський національний університет
88000, вул. Університетська, 14, м. Ужгород, Україна
Закарпатський угорський інститут ім. Ференца Ракоці II
90202, пл. Кошута, 6, м. Берегове, Україна
https://orcid.org/0000-0003-4038-4945

Анотація. Зростання економіки має низку негативних наслідків для навколишнього середовища. З першого погляду вплив навколишнього середовища на добробут населення здається незначним, але низка наукових робіт доводять, що стан навколишнього середовища має не менший вплив на добробут населення, ніж економічні чи соціальні чинники. Мета даного дослідження полягала у оцінці екологічного добробуту України у порівнянні з країнами-членам Організації економічного співробітництва та розвитку. Для оцінки екологічного добробуту використовуються індикатори Індексу кращого життя Організації економічного співробітництва та розвитку та Індексу екологічної ефективності. Для дослідження і оцінки екологічного добробуту використано математичний апарат теорії нечітких множин. У ході дослідження побудовано систему нечіткого логічного виводу, за допомогою якої отримано оцінку екологічного добробуту України порівняно з країнами-членам Організації економічного співробітництва та розвитку. При дослідженні використано три індикатори: забруднення повітря, санітарія та питна вода, сільське господарство. Результати дослідження свідчать про те, що порівняно з країнами-членами Організації економічного співробітництва та розвитку, Україна знаходиться на межі між середнім та високим рівнями екологічного добробуту. Результати дослідження підтверджуються Індексом екологічної ефективності, за яким Україна у 2024 р. посідає 41 місце зі 180 досліджуваних країн, піднявшись на 11 позицій порівняно з 2022 роком. Хоча Україна вже продемонструвала прогрес у Індексі екологічної ефективності у 2024 році порівняно з результатами за 2022 рік, вона може зберегти таку тенденцію лише за умови, що знатиме, які саме екологічні показники важливо покращити. Тож практична цінність даного дослідження полягає в демонстрації екологічних індикаторів, які потребують покращення

Ключові слова: екологія; нечіткі множини; система нечіткого виводу; екологічний добробут; математичне моделювання

BULLETIN of Cherkasy State Technological University

Journal homepage: https://bulletin-chstu.com.ua/en

Vol. 29 No. 2. 2024

UDC 621.3:004

DOI: 10.62660/bcstu/2.2024.79

Big Data technologies in the process of forecasting electricity generation from solar photovoltaic power plants

Oleksandr Stoliarov*

Postgraduate Student

National University of Food Technologies

01601, 68 Volodymyrska Str., Kyiv, Ukraine

https://orcid.org/0009-0001-7499-4672

Abstract. This research aimed to develop methods for using Big Data technologies to forecast electricity generation from solar photovoltaic power plants, which is crucial for optimising energy production and increasing the efficiency of solar resource utilisation. The study employed a method of analysing the economic feasibility of using energy storage systems and a comparative analysis of electricity buying and selling prices on the market. An experiment involving software tools and algorithms for processing, analysing, and modelling large volumes of data was also conducted. As a result of the research, methodologies were developed that encompass data collection and analysis, information visualisation, selection and training of forecasting models based on available data, as well as monitoring and testing their effectiveness. Graphical diagrams were constructed to illustrate the stages of data processing and analysis, the process of forecasting electricity generation for different time periods, and the process of training a model based on data, monitoring, and testing the model. Additionally, a graph was created to show the typicality and range of values, and a graph to display the change in electricity prices throughout the day. Furthermore, technological tools for using Big Data were described, the cost of electricity was calculated, and the economic attractiveness of using energy storage systems was assessed. As a result of the research, a potential profit indicator from price arbitrage was established, as well as economic parameters for the feasibility of using energy storage management based on an analysis of differences in electricity purchase and sale prices. The results obtained can be useful for energy companies and organisations involved in the production of electricity from solar photovoltaic power plants, allowing them to optimise energy production and increase the efficiency of solar resource utilisation

Keywords: information and analytical tools; energy market; big data processing; energy production optimisation; forecasting models

Article's History: Received: 05.01.2024; Revised: 15.04.2024; Accepted: 27.05.2024.

INTRODUCTION

Existing methods for predicting electricity generation have inherent limitations, as they do not always provide sufficient accuracy or the ability to anticipate changes in environmental conditions. Therefore, there is a need to develop new approaches based on modern Big Data technologies to improve forecasting efficiency and optimise electricity generation from solar photovoltaic power plants. The necessary theoretical knowledge includes concepts related to forecasting electricity generation from solar photovoltaic power plants and the use of Big Data technologies. This encompasses an understanding of the operating principles of solar photovoltaic power plants, their technical characteristics, and the influence of external factors on their performance. Additionally, it is important to understand Big Data technologies and their application in the energy sector, particularly methods for processing large volumes of data for forecasting and optimising electricity generation. This study examined how to implement Big Data technologies in forecasting electricity generation from

Suggested Citation:

Stoliarov, O. (2024). Big Data technologies in the process of forecasting electricity generation from solar photovoltaic power plants. *Bulletin of Cherkasy State Technological University*, 29(2), 79-92. doi: 10.62660/bcstu/2.2024.79.

*Corresponding author

solar photovoltaic power plants, how to use data from photovoltaic power plants to improve energy production efficiency, and which data analysis methods and algorithms can be applied to increase forecast accuracy.

To understand the current state of research in the field of Big Data technologies for electricity generation forecasting, it is necessary to refer to previous studies. In particular, authors U. Pysmenna *et al.* (2021) examined market mechanisms for stimulating the development of energy storage systems in Ukraine. They analysed the current state of such systems and the profitability of investment projects for their installation and operation to participate in various market segments. Researchers V. Stanytsina *et al.* (2023) investigated the possibilities of increasing the energy independence of territorial communities in Ukraine. They considered the criteria and technologies for distributed electricity generation and heat supply, as well as the use of renewable energy sources and local fuels to provide communities with electricity and heat. In their article, M. Kyzym *et al.* (2022) analysed the issues of energy efficiency and energy dependence in Ukraine, considering global trends. They developed a methodology for comparing the resource characteristics of power generation technologies to determine the most efficient Ukrainian solutions. Authors L.C. Kho *et al.* (2023) highlighted the important role of solar photovoltaic energy in distributed energy sources and explained that the amount of solar power generation is steadily increasing worldwide. They also noted that the high dependence of solar power generation on changes in ambient parameters and asset operating conditions is one of the challenges in solar photovoltaics.

Furthermore, S.S. Chandel *et al.* (2023) emphasised that accurate forecasting of electricity generation from industrial solar installations is crucial for ensuring grid stability. Their research presented an updated review of machine learning methods for photovoltaic energy forecasting. The research by K.J. Iheanetu (2022) demonstrated that while electricity production from renewable sources is rapidly increasing, its variable nature complicates forecasting. This poses problems for grid stability and reliability, alongside economic benefits. Additionally, R. Aman *et al.* (2023) highlighted that photovoltaic energy generation is dependent on variable meteorological conditions, and accurate photovoltaic energy estimation is essential for improving the performance of solar installations. A new deep learning model for photovoltaic energy forecasting based on various weather conditions was developed, employing a two-stage deep learning architecture combining long short-term memory (LSTM) and convolutional neural network (CNN). In the research of S. Tajjour & S.S. Chandel (2022), transfer learning was used to train a deep neural network based on solar irradiance data to forecast power output using a small dataset for a 400-kW solar plant in India. On the other hand, deep learning, which is widely used for photovoltaic generation forecasting, requires large amounts of data for proper training, which are initially unavailable for new power plants.

In their research, V.D. Gil-Vera & C. Quintero-López (2023) developed a model for forecasting solar energy production using machine-learning approaches and historical data. The results obtained indicate that the Random Forests method provides the most accurate forecasts, allowing grid operators to plan more effectively. Authors M.M. Hossain *et al.* (2023) highlighted the geographical and environmental characteristics of Bangladesh, where there is a need to find alternative energy sources. Solar power plants, particularly in the Char Fasson region, could be an effective solution as this region has a high potential for solar energy generation.

The aim of this research was to implement new methods for integrating Big Data technologies into the process of forecasting electricity generation from solar photovoltaic power plants. Overall, the study focused on addressing the problem of efficient use of solar resources and optimising energy production through the application of modern Big Data analysis technologies.

MATERIALS AND METHODS

This research addressed challenges related to the implementation of Big Data technologies in forecasting electricity generation from solar photovoltaic power plants, optimising energy production efficiency using data from photovoltaic power plants, and selecting the best data analysis methods and algorithms to improve forecast accuracy. To address these issues, a vast amount of data was collected using Market Operator (n.d.). The data was processed and analysed using tools such as Microsoft Word and Microsoft Excel, including statistical analysis and data visualisation. Trained models for forecasting electricity generation were tested on independent datasets to verify their effectiveness and accuracy.

After processing the data, forecasting models for electricity generation from solar photovoltaic power plants were developed and trained based on the collected data. Input data for training the models included various parameters affecting the electricity generation process. Specifically, data on solar activity levels, weather conditions (temperature, humidity, precipitation, etc.), geographic parameters (latitude, longitude, altitude), and the technical characteristics of the photovoltaic power plant itself. Model training was conducted on input data that had been carefully collected and processed. After training, the models were tested on independent datasets that were not used during training. This allowed for verification of the model's effectiveness and accuracy in forecasting real data that had not been used for their training.

An important aspect was determining the hourly buying and selling price in the "day-ahead" market (DAM) for the period from 14:00 to 15:00 (1):

$$P_{14\text{-}15} = P_{14.00} \cdot 0.5 + P_{15.00} \cdot 0.5, \quad (1)$$

where $P_{14\text{-}15}$ – the hourly DAM price for the operating period from 14:00 to 15:00 UAH/MWh; $P_{14.00}$ – the hourly DAM price at 14:00 UAH/MWh; $P_{15.00}$ – the hourly DAM price at 15:00 UAH/MWh.

Descriptive and inferential statistical methods were employed to analyse the data. These included analysis of variance, correlation analysis, and regression analysis. Each method was selected based on its suitability for the specific data type and research objectives. The data collection process was standardised and documented to ensure data quality and reliability. To assess the economic attractiveness of using energy storage systems, an analysis of the daily load market was conducted, and electricity costs were calculated based on the difference between electricity purchase and sale prices.

Two approaches were considered for calculations based on actual data from photogalvanic power plants (PGPPs). In the first approach, to assess the additional revenue from arbitrage and the payback period, actual photovoltaic power plant data was used, with the analysis based on hourly buy/sell prices of electricity on the DAM in 2023. To ensure objective assessment, calculations were performed for each month of the seasonal period (February, May, August, and November). Using a graphical method, the day with the minimum and maximum DAM prices for each corresponding month was determined, as well as the volume of electricity generation during the period of minimum prices. The second approach considered a simplified method for calculating additional arbitrage revenue, based on actual solar photovoltaic power plants data for 2023 and the difference between electricity purchase and sale prices determined by the 2023 price caps. Price caps, set by the National Commission, are the maximum and minimum prices for electricity on the Ukrainian electricity market. The application of this method allows for the estimation of the probable volume of electricity sales and its cost during periods of maximum and minimum prices, as well as determining the difference in sales value throughout the year. Formula (1) was used for the calculations.

The research also included an analysis and evaluation of the effectiveness of the developed methods. Additionally, software tools such as Apache Hadoop, Apache Spark, and Apache Kafka were described and analysed. All experiments were conducted using computers with high-performance computing capabilities, as well as specialised software for data processing and analysis.

RESULTS

Development of methods for utilising Big Data technologies to forecast electricity generation from solar photovoltaic power plants. The development of efficient methods for predicting electricity generation from solar photovoltaic power plants is becoming increasingly relevant in the context of the growing role of solar energy in modern energy systems. The use of Big Data technologies in this context opens up wide opportunities for improving forecasting through the analysis of large volumes of data and the application of specialised algorithms.

One method of utilising Big Data for forecasting electricity generation involves the collection and processing of large volumes of data. This procedure includes gathering a variety of information about electricity generation from solar photovoltaic power plants, such as data on solar irradiance, temperature, cloud cover, and other factors. Subsequently, data is processed to remove noise and anomalies, such as data loss or inaccuracies that may occur during collection. Additionally, the collected data can be divided into three main categories: meteorological data, solar photovoltaic power plant status data, and geospatial data. Meteorological data includes both historical and forecasted values of solar radiation, temperature, humidity, wind speed, and cloud cover. Data on the status of solar photovoltaic power plants includes the characteristics of panels and inverters, as well as operational data such as voltage, current, and power. Geospatial data encompasses digital elevation models for determining the tilt angle and azimuth of panels, alongside information on land use and shading. Data storage is also a critical consideration. Large volumes of information, such as data on solar activity, temperature, and other parameters, can be collected from photovoltaic power plants and environmental sensors. Storing this data in specific distributed data storage systems allows for efficient storage and processing of information for further analysis. The flowchart in Figure 1 illustrates the data processing and analysis process. The first block shows data collection and processing. The second block presents the analysis and visualisation of this data. The third block depicts the data storage process.

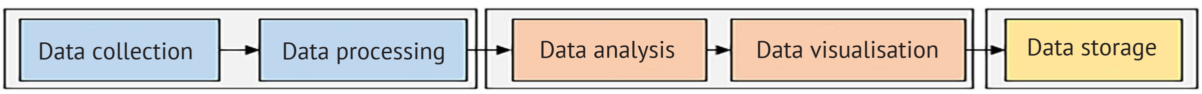


Figure 1. Data collection and processing

Source: compiled by the author

Another method of utilising Big Data involves data analysis and visualisation. After processing information about electricity generation from solar photovoltaic power plants, analysis is conducted to identify dependencies and correlations between various factors such as solar irradiance, temperature, time of day, etc. Various visualisation methods, such as graphs, charts, and heatmaps, are used to better understand the relationships between data. Data analysis includes data cleaning and processing, such as removing errors, missing values, and outliers, as well as normalising and standardising data. It also involves pattern discovery, i.e., using machine learning and data mining techniques to identify relationships between meteorological, operational, and geospatial data and electricity generation. Additionally, an important aspect is modelling, which involves developing models for forecasting electricity generation based on identified patterns. Overall, data analysis involves using machine learning and data mining techniques to identify relationships and patterns in the obtained data, forecasting electricity generation based on historical data and factors such as weather conditions, geographic location, etc. In turn, visualisation involves creating interactive monitoring and forecasting dashboards, visualising electricity generation forecasts, and comparing forecasts with actual values.

The next method involves the selection of a forecasting model (Fig. 2). This could be a statistical model, a machine learning model, or a neural network. Key factors in model selection include forecast accuracy, model complexity, and the ability to be extended to accommodate new data. Forecasting covers various time horizons, including short-term, medium-term, and long-term. Short-term forecasting involves generation in the next few hours or days, medium-term forecasting – in the next few weeks or months, and long-term forecasting – in the next few years, using methods such as energy models and system dynamics. The use of forecasting models involves developing models based on collected data and analysing them using machine learning methods such as linear regression, ensemble methods, and neural networks. These models are used to forecast electricity generation based on various factors. Initially, a forecasting model is selected, which can be statistical or machine learning-based. These models are then used to develop forecasts for short-term, medium-term, and long-term time horizons. The result is electricity generation forecasts that can be used for decision-making in the energy sector.

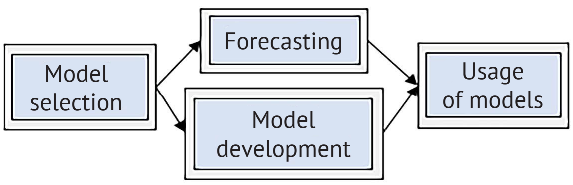


Figure 2. Choosing a forecasting model
Source: compiled by the author

Conversely, the method of training a model based on data involves conducting a detailed analysis and processing of the collected data before the actual training process begins (Fig. 3). After selecting the appropriate model, training starts with the aim of adapting the model to the input data and identifying patterns and relationships between them. However, attention should first be paid to data preparation, including cleaning it of anomalies, removing duplicates, and correcting missing or incorrect values. The training dataset can be divided into two parts: training and testing, for effective model validation. The model is then trained by adapting to the patterns and dependencies in the training data. This involves tuning the model's parameters to maximise forecast accuracy. The model can use various training methods such as gradient descent, backpropagation, etc., depending on the chosen algorithm. After model training is complete, it is evaluated using the test dataset to verify its accuracy and effectiveness. This process allows for an evaluation of how well the model can generalise to new, previously unseen data and draws conclusions about its predictive capabilities. The first block is responsible for the data preparation stage, the second – for model training, and the third – for model evaluation.

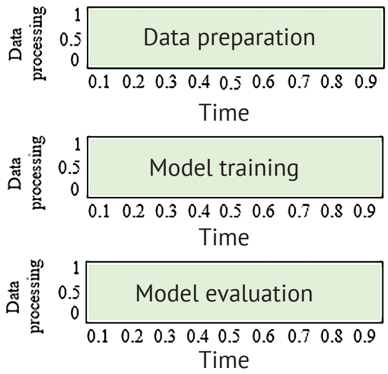


Figure 3. Model training based on data
Source: compiled by the author

The method of monitoring and testing the model involves implementing monitoring systems, testing, and evaluating the model. A crucial aspect of this method is the development of monitoring systems that can collect real-time data on solar activity, weather conditions, and the performance of photovoltaic power plants. The collected data is used for continuous improvement of forecasting models, ensuring their accuracy and adaptability to changing environmental conditions. Attention should also be paid to testing and evaluating the model on a dedicated test dataset. This allows for determining the accuracy of the model's forecasts and identifying its strengths and weaknesses. If the model proves satisfactory during testing, it can be considered suitable for use in forecasting electricity generation from solar photovoltaic power plants. The flowchart in Figure 4 illustrates the model monitoring and testing method. The first block shows the process of continuously monitoring the model's accuracy and adaptability by collecting feedback. The second block highlights the importance of developing monitoring systems to collect real-time data on solar activity and other factors. The third block shows the process of testing and evaluating the model to determine its accuracy, as well as its strengths and weaknesses. Thus, data analysis and visualisation, data-driven model training, and the model monitoring and testing method are key stages in this process. Overall, the use of Big Data methods for forecasting electricity generation from solar photovoltaic power plants allows for improved forecasting accuracy, ensuring that models are adaptable to changing environmental conditions and enabling the efficient use of solar energy in modern energy systems.

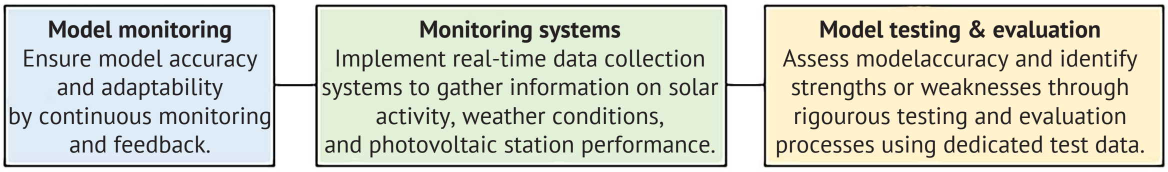


Figure 4. Model monitoring and testing

Source: compiled by the author

The use of Big Data and energy storage to optimise solar power generation and improve the efficiency of energy companies. The question of the effectiveness of the developed methods should also be considered in the context of Ukraine's Energy Strategy until 2035, as the significant impact of these methods on the power balance in the system is highlighted. The introduction of a share of renewable energy sources (RES), in particular photovoltaic power plants, at the level of 25% by 2035, poses the task for the energy system of ensuring the stability and reliability of power supply. The developed forecasting methods prove to be an effective tool for addressing these challenges, contributing to better energy resource management and ensuring the quality of power supply for consumers. In addition, the application of energy storage, particularly based on calculations using the universal energy storage (UES) model, aims to increase the efficiency of energy companies and improve the quality of services in the field of electricity generation. Such an approach to using the developed methods not only contributes to the optimisation of processes in the energy sector but also to achieving strategic goals in the development of renewable energy in Ukraine. The use of Big Data technologies and energy storage in the energy sector has significant potential for optimising solar power generation and increasing the efficiency of energy companies. One of the key aspects in this context is the concept of "arbitrage", which involves the commercially attractive accumulation of electricity during periods of low prices and its subsequent release into the grid during periods of high prices. This strategy allows for smoothing out nighttime troughs as well as morning and evening peaks of the daily load curve of the energy system. To substantiate the feasibility of using energy storage in conjunction with photovoltaic power plants in the conditions of the modern Ukrainian energy market, calculations were carried out based on actual data of the PGPPs operation for two scenarios. The first scenario was based on hourly buy-sell prices of electricity on the DAM in 2023, while the second scenario took into account the difference in buy-sell prices of electricity on the DAM, using price caps for 2023. Such an approach allows for providing convincing data on the prospects of using these methods in solving the current tasks of the energy sector (Fig. 5).

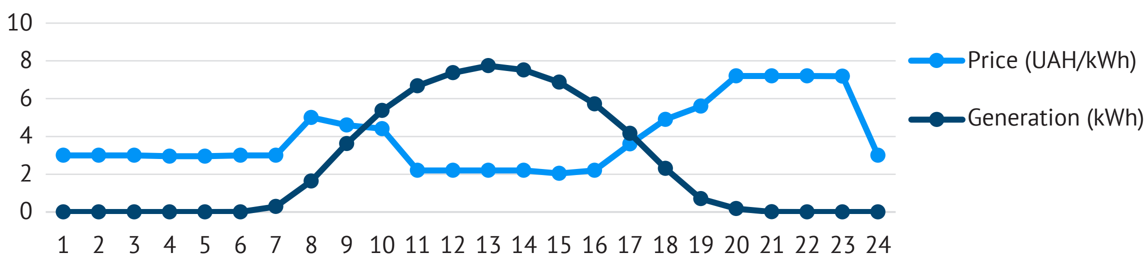


Figure 5. Graph of hourly prices for the buying and selling of electricity on the DAM and the generation of PGPPs for August 2023

Source: compiled by the author based on Market Operator (n.d.)

After this, calculations were conducted to estimate the expected volume of electricity sales and its value during periods of minimum and maximum prices (Table 1). The difference in the value of electricity sales was assessed for each month, as well as in total for the entire year based on the obtained results.

Table 1. Calculation of the difference in the value of electricity sales in 2023

	Sales volume (kWh)	Price (UAH/kWh)	Cost (UAH)	Difference (thousand UAH)	Difference (thousand UAH/ month)
February					
12:00-17:00	1027.33	3.790-3.840	3,905.56	0.039	1.1
19:00-20:00	1027.33	3.840	3,944.95		
May					
12:00-17:00	3099.38	2.650	8,213.36	4.184	129.71
19:00-20:00	3099.38	4.000	12,397.52		
August					
12:00-17:00	2910.94	2.200	6,287.01	14.55	451.05
19:00-21:00	2910.94	7.200	20,958.77		
November					
12:00-17:00	360.24	5.515-5.224	1,959.80	0.13	3.89
19:00-20:00	360.24	5.800	2,089.39		

Source: compiled by the author based on Market Operator (n.d.)

Final calculations were conducted to estimate the expected revenue from arbitrage, taking into account the self-consumption costs of solar photovoltaic power plants, as well as the costs of dispatching and transmitting electricity for 2023. However, these calculations did not consider any negative factors that could impact electricity generation. It is important to highlight the final calculations of the investment attractiveness of employing energy storage systems for collaboration with solar photovoltaic power plants to provide "arbitrage" generation services (Table 2). The assessment of investment attractiveness, considering the stated assumptions and parameters, indicates that the use of energy storage is commercially viable in this case. The projected annual return is approximately 40%, ensuring a return on investment within six years, which aligns with the projected lifespan of the battery energy storage systems. Considering the additional income, the payback period is reduced to 3.8 years. The second scenario employs a simplified method for calculating additional arbitrage revenue, relying on actual data from solar photovoltaic power plants in 2023 and the price difference between buying and selling electricity, determined by the price caps set for 2023. Based on the results obtained, the potential arbitrage revenue for 2023 was calculated, excluding self-consumption costs and other operational expenses (Table 3).

Table 2. Economic indicators of the investment attractiveness of using UES for combined operation with PGPPs based on 2023 data

No.	Indicator	Result
1	Volume of electricity sales (MWh)	1,156.8
2	Revenue from electricity sales per year (thousand UAH)	3,065.29
3	Operational costs per year (thousand UAH)	15.2
4	Costs for transmission system operator (TSO) and distribution system operator (DSO) services per year (thousand UAH)	628.63
5	Actual revenue for the year (thousand UAH)	2,421.46
6	Probable additional revenue for the year (thousand UAH)	1,757.25
7	Probable total revenue for the year (thousand UAH)	4,178.7
8	Average electricity generation from 12:00 to 17:00 (MW)	1,849.5
9	Required installed capacity of UES for RES (kW)	1.0
10	Cost of energy storage installation per kW (USD)	210
11	Exchange rate for the calculation period (UAH)	37.5
12	Estimated budget component of the UES project (CAPEX) (thousand UAH)	15,750
13	Estimated budget component of the UES project (OPEX) (thousand UAH)	123
14	Total costs for UES (thousand UAH)	15,873
15	Payback period of investments (years)	6.6

Continued Table 2.

No.	Indicator	Result
16	Payback period of investments considering probable additional revenue (years)	3.8
17	The projected operational lifespan of battery UES (years)	15

Source: compiled by the author based on Market Operator (n.d.)

Table 3. Potential arbitrage revenue for the year, excluding self-consumption costs, and costs for TSO and DSO services

Period	Potential income (thousand UAH)	Hours of minimum on the DAM	Price caps, min-max (UAH/MW)	Difference per 1 MW by price (UAH)	Difference by 1 MW in fact
December	1,696.20	13-15	2,646.25-4,000.0	max-1,353.75; min-0	54.72
January	1,887.04	12-13	2,646.25-4,000.0	max-1,353.75; min-0	60.87
February	5,889.25	13-14	2,646.25-4,000.0	max-1,353.75; min-0	210.33
March	11,515.95	14-15	2,646.25-4,000.0	max-1,353.75; min-0	371.48
April	27,642.80	14-15	2,646.25-4,000.0	max-1,353.75; min-0	921.43
May	41,644.45	13-14	2,646.25-4,000.0	max-1,353.75; min-0	1,343.37
June	38,286.05	13-14	2,646.25-4,000.0	max-1,353.75; min-0	1,276.20
July	152,020.92	13-14	10 (5,600.0)-7,200.0	max-7,190; min-1,600	4,903.90
August	156,585.05	13-14	10 (5,600.0)-7,200.0	max-7,190; min-1,600	5,051.13
September	154,151.91	13-14	10 (5,600.0)-7,200.0	max-7,190; min-1,600	5,138.40
October	151,152.02	13-14	10 (5,600.0)-7,200.0	max-7190; min-1,600	4,875.87
November	76,690.38	12-13	10 (5,600.0)-7,200.0	max-7,190; min-1,600	3,334.36
Total for the year	819,162				

Source: compiled by the author based on Market Operator (n.d.)

Final calculations were performed to assess the investment attractiveness of implementing a UES system based on its 2023 operational data, disregarding any negative factors that might affect electricity generation (Table 4).

Table 4. Economic parameters of the feasibility of using UES based on the difference in electricity purchase and sale prices on the DAM in 2023

Indicator	Result
Costs for transmission system operator (TSO) and distribution system operator (DSO) services per year (thousand UAH)	628.63
Volume of electricity sales (MWh)	1,156.8
Revenue from electricity sales per year (thousand UAH)	3,065.29
Operational costs per year (thousand UAH)	15.2
Actual revenue for the year (thousand UAH)	2,421.46
Probable additional revenue for the year (thousand UAH)	819,162
Probable total income for the year (thousand UAH)	3,240.62
Required installed capacity of UES for RES (MW)	1.0
Cost of energy storage installation per kW (USD)	210
Exchange rate for the calculation period (UAH)	37.5

Continued Table 3.

Indicator	Result
Estimated budget component of the UES project (CAPEX) (thousand UAH)	15,750
Estimated budget component of the UES project (OPEX) (thousand UAH)	123
Total costs for UES (thousand UAH)	15,873
Payback period based on actual income (years)	6.52
Payback period considering additional income (years)	4.9
The projected operational lifespan of battery UES (years)	15

Source: compiled by the author based on Market Operator (n.d.)

The assessment of investment attractiveness indicators under the considered conditions and parameters confirms that the use of a coordinated UES in this case is economically viable, as the expected additional annual income is approximately 25%. The payback period does not exceed the planned lifespan of battery energy storage systems and is 6.52 years; considering the expected additional income, this period is reduced to 4.9 years.

Thus, the calculated investment attractiveness indicators in both UES usage scenarios are almost identical, indicating its commercial attractiveness in the "arbitrage" generation sector. However, there are two key aspects that can affect the expected profit. Firstly, the highly unstable nature of generation from PGPPs depends on various factors such as weather conditions and equipment technical conditions. Secondly, the regulatory aspects of the Ukrainian electricity market. The implementation of ESS (energy storage systems) requires significant investments and the correct choice and consideration of the expected economic effect is a complex task. Given the various energy storage technologies and their economic maturity, many aspects of their application must be considered. The business model for using UES in the electricity market depends on the intended purpose of the electrical installation and the chosen market segment.

Technological tools and software in Big Data analysis. Within the realm of Big Data, numerous software tools are employed for the collection, processing, and analysis of vast datasets. Apache Hadoop stands out as one of the most popular open-source frameworks for the distributed processing of large data volumes (Fig. 6). It is grounded in the MapReduce model, which enables efficient data processing across a cluster of servers. Hadoop encompasses various components such as the Hadoop Distributed File System (HDFS) for storing data in blocks across multiple servers, MapReduce for data processing and parallel computing, and YARN for resource management and job scheduling within the cluster. A key advantage of Hadoop is its scalability, allowing it to handle massive datasets distributed across dozens or even hundreds of servers.

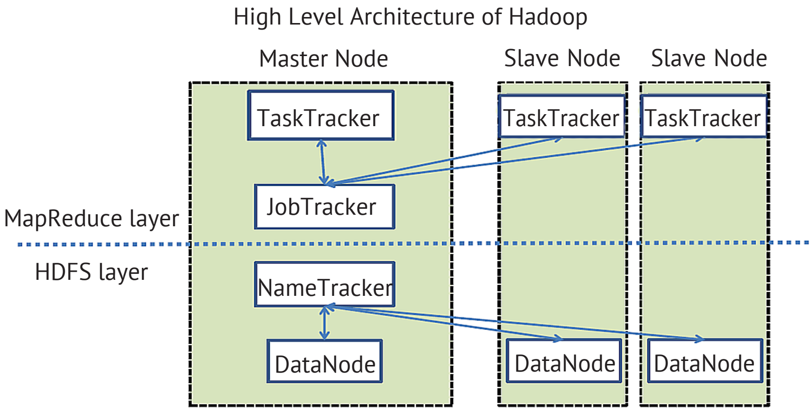


Figure 6. The main components of Apache Hadoop

Source: S. Bappalige (2014)

In turn, Apache Spark is a fast and powerful framework designed for the real-time processing of large datasets (Fig. 7). It provides an application programming interface (API) for distributed data processing, supporting programming languages such as Python, Java, and Scala. Spark leverages in-memory caching and optimised operations, enabling it to perform significantly faster than Hadoop MapReduce. Moreover, Spark incorporates modules for machine learning, deep learning, and graph computations, making it a versatile tool for data analysis. At the core of Spark lies the concept of resilient distributed datasets (RDDs), which serve as its fundamental data structure. Spark offers an API for manipulating RDDs, as well as modules for working with streaming data, machine learning, and structured query language (SQL) queries.

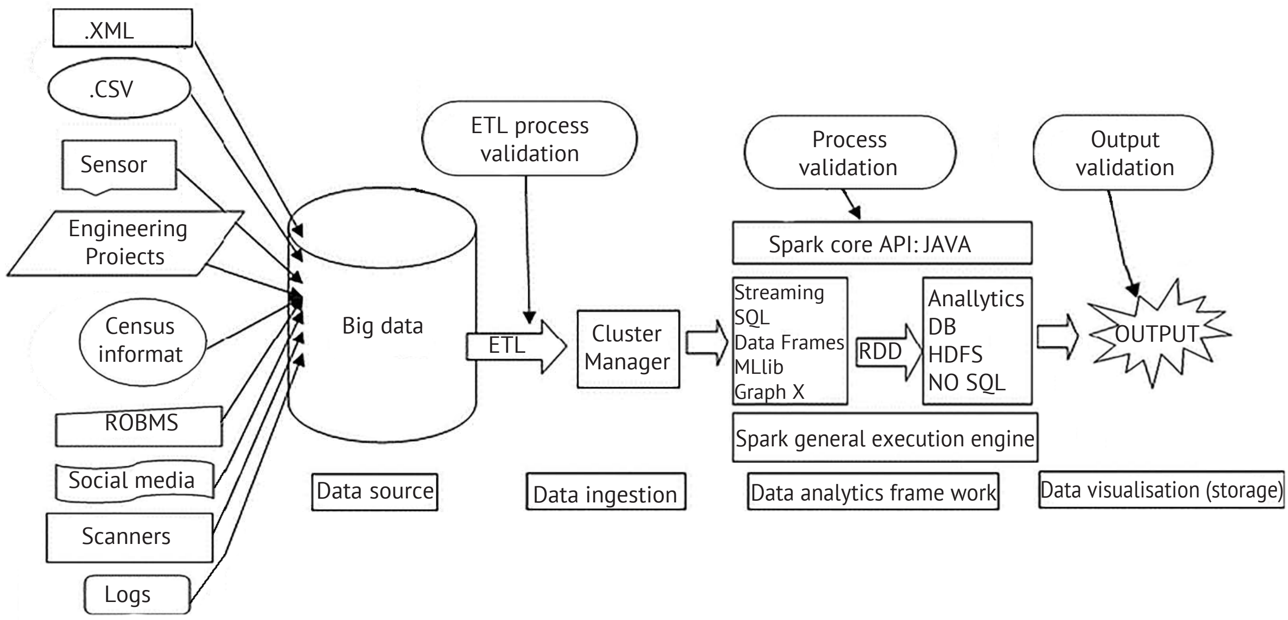


Figure 7. Big Data testing ecosystem with Apache Spark

Source: R.K. Pallamala & P. Rodrigues (2022)

Another notable technology is Apache Kafka, a distributed streaming platform designed for real-time data processing (Fig. 8). It facilitates efficient data exchange between various systems and applications, while also storing data for subsequent analysis. Kafka offers horizontal scalability and guarantees message delivery and order preservation. Apache Kafka is commonly used for real-time data streaming between different systems. It consists of three primary components: producers, which generate data streams; topics, which serve as categories or channels for data streams; and consumers, which process or store the data streams. Kafka ensures high availability and scalability, making it an excellent tool for stream processing.

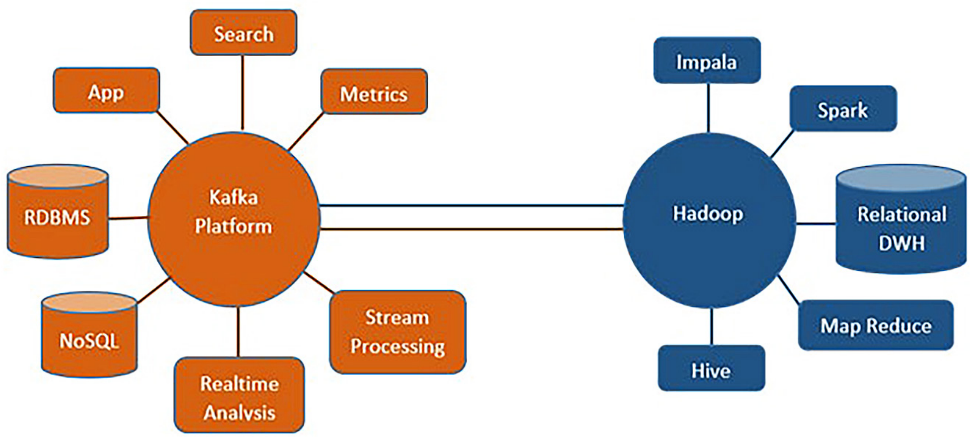


Figure 8. Comparison of Kafka and Hadoop platforms

Source: N. Saxena (2019)

Additionally, during the data processing phase, extract, transform, load (ETL) tools such as Apache NiFi or Talend can be employed. These tools enable the cleaning, transforming, and loading of data from various sources into appropriate systems for further analysis. The ETL process encompasses the stages of data extraction, transformation, and loading from one source to another, ensuring data quality and readiness for analytical use.

DISCUSSION

This study highlights the relevance of using Big Data models for forecasting electricity generation from solar photovoltaic power plants. The study by G. de Freitas Viscondi & S.N. Alves-Souza (2019) corroborates these findings, as it employed a methodology that confirmed the applicability of Big Data models for predicting electricity generation, particularly emphasising the accuracy of machine learning and neural network methods.

Furthermore, the conducted research on Big Data technologies acknowledges machine learning and neural networks as the most precise techniques for addressing this issue. However, this research also expands upon the understanding of this topic by utilising various methods and tools for data analysis.

This research expands upon the study of S.-M. Je *et al*. (2021), attempted to accurately forecast electricity demand through Big Data visualisation and proposed a flexible and balanced Big Data virtualisation for power generation based on a photovoltaic power plant is proposed. In contrast to the researchers' article, this study incorporates not only Big Data visualisation but also other tools for accurately predicting electricity production from solar photovoltaic power plants. This enables a more comprehensive examination of various aspects of this problem and facilitates the discovery of effective solutions to further advance renewable energy.

Similar to the research of M.A. Nazari *et al*. (2023), this research identifies solar energy as a key renewable source with significant potential in the modern energy landscape. However, unlike the researchers' study, this Big Data research focuses on developing and applying various methods and software tools for accurately forecasting electricity generation from solar photovoltaic power plants. This can facilitate the integration of solar energy into production processes.

To address the issue of solar energy intermittency within power grids, Y. Nie *et al*. (2022) identified that image-based forecasting using deep learning is a promising method for predicting short-term fluctuations. However, this study differs from the researchers' approach to solar energy forecasting. The current research focuses on leveraging Big Data to accurately predict electricity generation from solar photovoltaic systems, providing a deeper understanding and more precise forecasts for the integration of solar energy into the grid.

Given the pressing global energy and environmental challenges, the development of renewable energy, particularly photovoltaic electricity generation, is of paramount importance. The research by W. Liu (2023) highlights that the vast potential of solar energy paves the way for the advancement of photovoltaic technologies and industrial growth, with solar energy primarily utilised in photovoltaic power generation systems. The current study on Big Data technologies complements the researchers' article by examining specific methods and software tools for forecasting electricity generation from solar photovoltaic power plants. This study provides more accurate and efficient forecasts to facilitate the integration of solar energy into power grids, leveraging large datasets and diverse analytical approaches.

As the expansion of photovoltaic electricity generation necessitates accurate solar energy forecasting, this research investigates the potential of Big Data and explores specific approaches to predicting electricity production from solar photovoltaic power plants. In turn, the research conducted by Z. Zhang *et al*. (2023) offers an effective model for forecasting electricity generation using deep learning. Nevertheless, this study broadens the arsenal of methods for predicting energy performance and provides additional opportunities for precise and long-term forecasting of electricity generation from solar photovoltaic power plants.

D.K. Dhaked *et al*. (2023) focused on utilising LSTM and back propagation neural network (BPNN) models to forecast solar power generation. While the current study also explores a similar aspect, it concentrates on alternative forecasting methods, employing a variety of approaches and software tools for data analysis and modelling. This expands the repertoire of forecasting methods and offers additional capabilities for accurate and long-term prediction of solar photovoltaic power generation.

In comparing this study with the research conducted by K.E. Sarah *et al*. (2020), it is important to note that the present study addresses the issue of accurately forecasting electricity generation from solar photovoltaic power plants. In contrast, the other study provides a comprehensive overview of the progressive development of solar photovoltaic technologies from first-generation to contemporary configurations, as well as an analysis of solar power generation in various regions. Thus, the conducted research on Big Data technologies broadens the scope by presenting an analysis of diverse software tools and methods for accurate and long-term forecasting of electricity generation from solar photovoltaic power plants.

Accurate forecasting of next-day solar photovoltaic energy is essential for the successful integration of solar power into the electricity grid. J.F. Torres *et al*. (2019) compared the effectiveness of a deep learning-based algorithm with other advanced methods for predicting solar energy and analysed the suitability of all methods for processing large time series data. In turn, this Big Data research expands upon the researchers' study by exploring various methods for modelling and analysing data to forecast electricity generation from solar photovoltaic power plants. It provides additional capabilities for accurate forecasting using diverse approaches, which can contribute to improving the efficiency of energy integration into the electricity grid.

Given the highly variable nature of electricity generation from solar PGPPs due to various factors, accurate forecasting is crucial for reliable grid integration and ensuring energy supply in off-grid systems. M.B. Arias & S. Bae (2021) proposed a forecasting model to estimate solar PGPPs output based on historical data such as solar irradiance, solar energy, and meteorological data, which are processed using Big Data techniques. However, the current research, while also considering electricity generation, explores a broader range of factors. It may also contribute to improving forecasting accuracy and the reliability of integrating solar energy into the grid.

Commonalities between this research and the study of Y. Zhang & Y. Wang (2022) include the exploration of a hybrid CNN and LSTM approach for forecasting solar power generation. M. Sabri & M. El Hassouni (2023) also proposed using LSTM, specifically with an autoencoder, for photovoltaic power generation forecasting. Additionally, R. Vaish & U.D. Dwivedi (2022) compared the LSTM model but also mentioned the Seasonal Autoregressive Integrated Moving Average (SARIMA) model, which is not included in this study.

In contrast to the article of C. Soumya *et al.* (2021), which examines the establishment of decentralised autonomous solar systems in rural areas of countries with limited fossil resources, the present study on Big Data technologies does not focus on addressing the challenges of installing solar systems in specific countries. Meanwhile, J. Gaboitaolelwe *et al.* (2023) provided a comprehensive overview of the latest advancements in existing approaches to forecasting energy generation from solar photovoltaic systems based on machine learning, which are also applied in this study.

While this research specifically analysed solar energy forecasting, Rajnish *et al.* (2024) also considered wind energy, which plays a crucial role in ensuring reliable power supply and grid stability. Conversely, A. Nahar *et al.* (2024) employed a machine learning time series model to forecast real-time power generation in two solar power plants, whereas this study utilises Big Data technologies for forecasting electricity generation from solar photovoltaic power plants.

This study, unlike the article of N. Kumar & M.M. Tripathi (2022), focuses on the utilisation of Big Data technologies for forecasting electricity generation from solar photovoltaic power plants. While the other study analysed the impact of solar power generation on electricity price forecasting, this study, similar to Suwarno & D. Pinayungan (2024), investigates solar power generation as a source of renewable energy.

This research places particular emphasis on developing forecasting models that can overcome the limitations associated with the variability and unpredictability of energy production from solar photovoltaic power plants. It also highlights the importance of accurate power forecasting for ensuring reliable electricity supply and grid stability. Compared to other studies, this research stands out for its unique approach, high complexity, and focus on the real-time efficiency of solar photovoltaic power plant forecasting. This study aimed to improve forecasting tools and methods to ensure the stability and efficiency of using solar energy as a renewable energy source.

CONCLUSIONS

This research developed effective methods for utilising Big Data technologies to forecast electricity generation from solar photovoltaic power plants. It was successfully demonstrated that using real-time data analysis, coupled with advanced forecasting models, enables accurate predictions of electricity generation. To achieve this goal, various algorithms were employed to collect, process, and analyse large volumes of data. Graphical representations were used to visualise the data processing workflow, and forecasting models were monitored and tested. Additionally, the software tools used for Big Data analysis were described and evaluated. Particular attention was paid to the economic aspects of using energy storage systems.

Overall, this study confirmed the economic viability of employing various systems and identified the potential for arbitrage based on electricity price differentials. High accuracy and stability were achieved in forecasting electricity generation from solar photovoltaic power plants. Additionally, the economic benefits of using energy storage systems based on analysing price differences in the electricity market were identified. The results demonstrate the success of the developed methods in forecasting electricity generation from solar photovoltaic power plants. It was established that the analysed forecasting models effectively account for various factors influencing electricity production. Furthermore, the economic viability of using energy storage systems based on the analysis of differences in the buying and selling prices of electricity was confirmed. It was found that such systems can be a crucial component for optimising energy production and increasing the efficiency of solar resource utilisation.

The research was constrained by several factors that influenced its conduct. Notably, there were limitations in the availability of data on solar activity, meteorological parameters, and electricity generation from solar panels, which could impact the accuracy and reliability of the forecasting. The complexity of modelling solar systems and the influence of unpredictable factors should also be considered, as these can lead to a decrease in the accuracy of forecasts. These limitations should be taken into account as the characteristics of solar panels can change over time due to wear and the influence of external factors, which may affect the effectiveness of forecasting in the future.

It is recommended to continuously improve machine learning algorithms using large datasets on solar activity. It is also important to analyse the results to identify opportunities for enhancing the forecasting system. For further enhancement, the utilisation of more detailed data and the implementation of new energy management strategies should be considered, with the aim of optimising production and conserving resources.

ACKNOWLEDGEMENTS

None.

CONFLICT OF INTEREST

None.

REFERENCES

[1] Aman, R., Rizwan, M., & Kumar, A. (2023). Novel and efficient hybrid deep learning approach for solar photovoltaic power forecasting considering meteorological parameters. *Electrical Engineering*. doi: 10.21203/rs.3.rs-3260351/v1.

[2] Arias, M.B., & Bae, S. (2021). Solar photovoltaic power prediction using big data tools. *Sustainability*, 13(24), article number 13685. doi: 10.3390/su132413685.

[3] Bappalige, S. (2014). *An introduction to Apache Hadoop for big data*. Retrieved from https://opensource.com/life/14/8/intro-apache-hadoop-big-data.

[4] Chandel, S.S., Gupta, A., Chandel, R., & Tajjour, S. (2023). Review of deep learning techniques for power generation prediction of industrial solar photovoltaic plants. *Solar Compass*, 8, article number 100061. doi: 10.1016/j.solcom.2023.100061.

[5] de Freitas Viscondi, G., & Alves-Souza, S.N. (2019). A systematic literature review on big data for solar photovoltaic electricity generation forecasting. *Sustainable Energy Technologies and Assessments*, 31, 54-63. doi: 10.1016/j.seta.2018.11.008.

[6] Dhaked, D.K., Dadhich, S., & Birla, D. (2023). Power output forecasting of solar photovoltaic plant using LSTM. *Green Energy and Intelligent Transportation*, 2(5), article number 100113. doi: 10.1016/j.geits.2023.100113.

[7] Gaboitaolelwe, J., Zungeru, A.M., Yahya, A., Lebekwe, C.K., Vinod, D.N., & Salau, A.O. (2023). Machine learning based solar photovoltaic power forecasting: A review and comparison. *IEEE Access*, 11, 40820-40845. doi: 10.1109/ACCESS.2023.3270041.

[8] Gil-Vera, V.D., & Quintero-López, C. (2023). Predictive modeling of photovoltaic solar power generation. *WSEAS Transatcions on Power Systems*, 18, 71-81. doi: 10.37394/232016.2023.18.8.

[9] Hossain, M.M., Pranto, M.A., & Hasan, M.N. (2023). *Design of a 20 MW grid-connected solar photovoltaic power plant for a community in Char Fasson*. Dhaka: Ahsanullah University of Science and Technology. doi: 10.13140/RG.2.2.30335.38560.

[10] Iheanetu, K.J. (2022). Solar photovoltaic power forecasting: A review. *Sustainability*, 14(24), article number 17005. doi: 10.3390/su142417005.

[11] Je, S.-M., Ko, H., & Huh, J.-H. (2021). Accurate demand forecasting: A flexible and balanced electric power production big data virtualization based on photovoltaic power plant. *Energies*, 14(21), article number 6915. doi: 10.3390/en14216915.

[12] Kho, L.C., Ngu, S.S., Annie, J., Sahari, S.K., Kipli, K., & Rulaningtyas, R. (2023). A case study of using long short-term memory (LSTM) algorithm in solar photovoltaic power forecasting. *ASM Science Journal*, 18, 1-8. doi: 10.32802/asmscj.2023.1162.

[13] Kumar, N., & Tripathi, M.M. (2022). Investigation on effect of solar energy generation on electricity price forecasting. *Journal of Intelligent & Fuzzy Systems*, 42(2), 1185-1197. doi: 10.3233/JIFS-189781.

[14] Kyzym, M., Shpilevskyi, V., Zinchenko, V., & Shpilevskyi, O. (2022). Global challenges and prospects for the structural development of Ukraine's electric energy industry. *Business Inform*, 7, 86-98. doi: 10.32983/2222-4459-2022-7-86-98.

[15] Liu, W. (2023). Key technologies for photovoltaic power generation. *Highlights in Science Engineering and Technology*, 43, 74-83. doi: 10.54097/hset.v43i.7407.

[16] Nahar, A., Al Mamun Rudro, R., Al Sohan, F.A., Reya, R.I., & Uddin, M.H. (2024). Predicting photovoltaic power generation by machine learning using time series analysis. *Energy Proceedings*, 43(5), 1-4. doi: 10.46855/energy-proceedings-11024.

[17] Nazari, M.A., El Haj Assad, M., & Salem, M. (2023). Solar energy and power generation. In M. El Haj Assad & M.A. Nazari (Eds.), *Solar collectors and systems* (pp. 15-29). Hauppauge: Nova Science Publishers. doi: 10.52305/EAML0250.

[18] Nie, Y., Li, X., Scott, A., Sun, Y., Venugopal, V., & Brandt, A. (2022). SKIPP'D: A SKy images and photovoltaic power generation dataset for short-term solar forecasting. *Solar Energy*, 255, 171-179. doi: 10.1016/j.solener.2023.03.043.

[19] Official Website of "Market Operator". (n.d.). Retrieved from https://www.oree.com.ua/.

[20] Pallamala, R.K., & Rodrigues, P. (2022). An investigative testing of structured and unstructured data formats in big data application using Apache Spark. *Wireless Personal Communications*, 122(1), 603-620. doi: 10.1007/s11277-021-08915-0.

[21] Pysmenna, U., Sotnyk, I., Kubatko, O., Trypolska, G., & Kurbatova, T. (2021). Market mechanism formation for stimulating the development of energy accumulation systems in Ukraine. *Visnyk of Sumy State University*, 3(3), 31-39. doi: 10.21272/1817-9215.2021.3-4.

[22] Rajnish, Saroha, S., & Kumar Saini, M. (2024). Numerical modeling and performance assessment of machine learning-based solar photovoltaic energy forecasting system. In *Power engineering and intelligent systems* (pp. 391-404). Singapore: Springer. doi: 10.1007/978-981-99-7383-5_29.

[23] Sabri, M., & El Hassouni, M. (2023). Photovoltaic power forecasting with a long short-term memory autoencoder networks. *Soft Computing*, 27(15), 10533-105553. doi: 10.1007/s00500-023-08497-y.
[24] Sarah, K.E., Roland, U., & Okafor, E.N.C. (2020). A review of solar photovoltaic technologies. *International Journal of Engineering Research & Technology*, 9(7), 741-749. doi: 10.17577/IJERTV9IS070244.
[25] Saxena, N. (2019). *Apache Kafka value in Big Data*. Retrieved from https://medium.com/dataseries/big-data-solution-using-apache-kafka-9a9d27cbf51c.
[26] Soumya, C., Deepanraj, B., & Ranjitha, J. (2021). A review on solar photovoltaic systems and its application in electricity generation. *AIP Conference Proceedings*, 2396(1), article number 020011. doi: 10.1063/5.0066291.
[27] Stanytsina, V., Nechaieva, T., Trokhaniak, V., Horskyi, V., & Teslenko, O. (2023). Electricity and heat supply technologies for increasing the energy independence of certain territorial communities. *System Research in Energy*, 4(75), 32-44. doi: 10.15407/srenergy2023.04.032.
[28] Suwarno, & Pinayungan, D. (2024). Solar power forecasting model as a renewable generation source on virtual power plants. *Bulletin of Electrical Engineering and Informatics*, 13(2), 702-712. doi: 10.11591/eei.v13i2.5870.
[29] Tajjour, S., & Chandel, S.S. (2022). Power generation forecasting of a solar photovoltaic power plant by a novel transfer learning technique with small solar radiation and power generation training data sets. *SSRN*. doi: 10.2139/ssrn.4024225.
[30] Torres, J.F., Troncoso, A., Koprinska, I., Wang, Z., & Martínez-Álvarez, F. (2019). Deep learning for big data time series forecasting applied to solar power. In *Proceedings of the international joint conference SOCO'18-CISIS'18-ICEUTE'18* (pp. 123-133). Cham: Springer. doi: 10.1007/978-3-319-94120-2_12.
[31] Vaish, R., & Dwivedi, U.D. (2022). Role of machine learning in forecasting solar and wind power generation. In S.M. Tripathi & A.K. Singh (Eds.), *Energy conversion: Methods, technology and future directions* (pp. 211-241). Hauppauge: Nova Science Publishers. doi: 10.52305/VXCB5652.
[32] Zhang, Y., & Wang, Y. (2022). A hybrid neural network-based intelligent forecasting approach for capacity of photovoltaic electricity generation. *Journal of Circuits, Systems and Computers*, 32(10), article number 2350172. doi: 10.1142/S0218126623501724.
[33] Zhang, Z., Wang, J., Xia, Y., Wei, D., & Niu, Y. (2023). Solar-mixer: An efficient end-to-end model for long-sequence photovoltaic power generation time series forecasting. *IEEE Transactions on Sustainable Energy*, 14(4), 1979-1991. doi: 10.1109/TSTE.2023.3268100.

Технології Big Data в процесі прогнозування електрогенерації від сонячних фотоелектростанцій

Олександр Столяров

Аспірант
Національний університет харчових технологій
01601, вул. Володимирська, 68, м. Київ, Україна
https://orcid.org/0009-0001-7499-4672

Анотація. Мета дослідження полягала в розробці методів використання технологій Big Data для прогнозування електрогенерації від сонячних фотоелектростанцій, що важливо для оптимізації виробництва енергії та підвищення ефективності використання сонячних ресурсів. Було використано метод аналізу економічної вигідності застосування систем управління зберіганням електроенергії та порівняльний аналіз цін на купівлю і продаж електроенергії на ринку, а також експеримент, що включав використання програмних засобів та алгоритмів для обробки, аналізу та моделювання великих обсягів даних. У результаті дослідження було розроблено методики, які охоплюють збір та аналіз даних, візуалізацію інформації, вибір та навчання моделей прогнозування на основі наявних даних, а також проведення моніторингу та тестування їх ефективності. Було побудовано графічні схеми, що вказують на етапи обробки та аналізу даних, процес прогнозування електрогенерації на різні періоди часу, а також процес навчання моделі на основі даних, моніторинг та тестування моделі. Крім того, створено графік, що показує оцінки типовості та розмаху значень, і графік, що відображає зміну цін на електроенергію протягом доби. Додатково було описано технологічні засоби для використання Big Data, розраховано вартість електроенергії та проведено оцінку економічної привабливості застосування систем управління зберіганням електроенергії. У результаті дослідження було встановлено показник імовірного доходу від арбітражу за різницею цін, а також економічні параметри доцільності використання управління зберіганням електроенергії на основі аналізу відмінностей у вартості купівлі та продажу електроенергії. Отримані результати можуть бути корисними для енергетичних компаній та організацій, що працюють у сфері виробництва електроенергії з сонячних фотоелектростанцій, дозволяючи їм оптимізувати виробництво енергії та підвищити ефективність використання сонячних ресурсів

Ключові слова: інформаційно-аналітичні засоби; енергетичний ринок; обробка об'ємних матеріалів; оптимізація енерговиробництва; моделі передбачення

ВІСНИК
Черкаського державного технологічного університету

Науковий збірник

Том 29, № 2. 2024

Заснований у 1996 р. Виходить чотири рази на рік

Оригінал-макет видання виготовлено у редакційно-видавничому відділі
Черкаського державного технологічного університету

Відповідальний редактор:
Я. Бурлака

Редагування англомовних текстів:
С. Воровський, К. Касьянов

Комп'ютерна верстка:
К. Пилипенко

Підписано до друку 27 травня 2024 р.
Формат 60*84/8
Умов. друк. арк. 11
Наклад 50 прим.

Адреса видавництва:
Черкаський державний технологічний університет
18006, бульв. Шевченка, 460, м. Черкаси, Україна
Тел.: +380638013283
E-mail: info@bulletin-chstu.com.ua
https://bulletin-chstu.com.ua/uk

BULLETIN
of Cherkasy State Technological University

Scientific Collection

Volume 29, No. 2. 2024

Founded in 1996. Published four times per year

The original layout of the publication is made in the editorial and publishing department
of Cherkasy State Technological University

Managing editor:
Ya. Burlaka

Editing English-language texts:
S. Vorovsky, K. Kasianov

Desktop publishing:
K. Pylypenko

Signed for print of May 27, 2024.
Format 60*84/8
Conventional printed pages 11
Circulation 50 copies

Editors Office Address:
Cherkasy State Technological University
18006, 460 Shevchenko Blvd., Cherkasy, Ukraine
Tel.: +380638013283
E-mail: info@bulletin-chstu.com.ua
https://bulletin-chstu.com.ua/en